# RPAC-2658M User Manual

ICP DAS CO., LTD. would like to congratulate your purchase of our Win-GRAF PACs. The ease of the 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.

#### ICP DAS Win-GRAF Linux-based PAC includes:

RPAC: RPAC-2658M

ICP DAS Win-GRAF WinCE-based PAC includes:

ViewPAC:	VP- <b>x</b> 208-CE7	[x: 2, 3, 4, 5, 6]
	VP- <b>x</b> 238-CE7	[x: 1, 4, 6]
WinPAC:	WP-5238-CE7	
	WP-8 <b>x</b> 28-CE7	[x: 1, 4, 8]
	WP-9 <b>x</b> 28-CE7	[x: 2, 4, 8]
XPAC:	XP-8 <b>x</b> 38-CE6	[x: 0, 1, 3, 7]

### **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, neither for any infringements of patents nor other rights of third parties resulting from its use.

### **Trademarks & Copyright**

Names are used for identification purposes only and may be registered trademarks of their respective companies. Copyright © 2020 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 the website:

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

### **Technical Service**

- Win-GRAF Web site: https://www.icpdas.com/en/product/guide+Software+Development\_\_Tools+Win-GRAF
- Win-GRAF Download Center: https://www.icpdas.com/en/download/index.php?nation=US&kind1=&model=&kw=win-graf
- Win-GRAF Demo program: Refer to <u>Chapter 12</u> for more information about Win-GRAF demo programs.

If you have any problems, please feel free to contact us. Email: <a href="mailto:service@icpdas.com">service@icpdas.com</a>

Chapter 1		re Setting	
1.1	Installing the Win-GRAF Workbenc	٦	7
1.2	Run the Win-GRAF Workbench		8
	1.2.1 Win-GRAF Operating Enviro	nment	9
	1.2.2 Win-GRAF Library Manager		.0
1.3	Setting the Win-GRAF PAC's IP Add	ress1	.1
1.4	Description of the Hardware		.2
Chapter 2	A Simple Win-GRAF Program	1	3
2.1	Creating a New Win-GRAF Project.		.3
	2.1.1 Creating a Template Project	(Demo01)1	.3
	2.1.2 Important Project Settings .		.5
2.2	Introduction of the Project		.7
	2.2.1 Demo01 - LD Program		.7
	2.2.2 Demo01 - Variables		.9
2.3	Give it a Try		0
	2.3.1 Declaring Variables		0
	2.3.2 Creating an LD Program		2
	6	AC 2	
	2.3.5 Testing the Program		1
Chapter 3	Modbus Slave: Allow the SCADA	/HMI Software to Access Win-GRAF Variables .3	3
3.1	To Enable Win-GRAF PAC as a Mod	ous TCP Slave 3	3
3.2	To Enable the Win-GRAF PAC as a N	1odbus RTU Slave 3	9
Chapter 4	Using "I/O Boards" Function	4	2
4.1	Ping_ip: Check the Connection of a	n Ethernet/ Internet Device 4	.3
4.2	i_scale: Scale Conversion		5
4.3	RPAC_2000_LED: Control the L1 an	d L2 LED on the RPAC-2658M4	7
4.4	RPAC_PAC_state: Detect the LAN s	ate of the RPAC-2658M4	8
Chapter 5	Modbus Master: connecting to	Modbus Slave Devices4	.9
5.1	Enabling Win-GRAF PAC as a Modb	us RTU/ASCII Master 4	.9
	5.1.1 Read DI data	5	52
	5.1.2 Write DO Data	5	5
	5.1.3 Read AI Data	5	57

	5.1.4	Write AO Data (16-bit)	
	5.1.5	Write AO Data (32-bit)	
	5.1.6	To Disable/Enable the Modbus RTU/ASCII Master Port	
5.2	Enabli	ing the Win-GRAF PAC as a Modbus TCP/UDP Master	65
	5.2.1	Connecting ET-7000 Series I/O Module	69
	5.2.2	Connecting the ET-7060 (6 DI, 6 Relay Output)	
	5.2.3	Connecting the ET-7018Z (10 AI)	
	5.2.4	To Disable/Enable the Modbus TCP/UDP Master Port	75
5.3	Conne	ecting a Modbus TCP Slave Device with two IP addresses	76
5.4	Acces	sing multiple Modbus RTU Slave via the tGW-700 Gateway	80
	5.4.1	Configuring the tGW-700 (Modbus TCP to Modbus RTU/ASCII Gateway)	80
	5.4.2	Accessing the LC series Module via the tGW-700 Gateway	83
	5.4.3	Test the Project (demo_tgw725.zip)	87
Chapter 6	Retai	n Variable and Data Storage	88
6.1	Using	the RETAIN_xxx Function	89
	6.1.1	RETAIN_VAR: Retain a Variable	90
	6.1.2	RETAIN_ARY: Retain an Array Variable	91
	6.1.3	RETAIN_FLAG_SET/GET/CLR (Set/Get/Clear the Retain Flag)	92
6.2	Save F	Retain Variables and Data to a File	
6.3	Save t	the Retain Data to a FRAM	98
	6.3.1	EEP_READ (Read a Value from the FRAM)	99
	6.3.1 6.3.2	EEP_READ (Read a Value from the FRAM) EEP_WRITE (Write a Value to the FRAM)	
Chapter 7	6.3.2		99
Chapter 7 Chapter 8	6.3.2 Excha	EEP_WRITE (Write a Value to the FRAM)	99 <b>100</b>
•	6.3.2 Excha	EEP_WRITE (Write a Value to the FRAM)	99 <b>100</b> <b>104</b>
Chapter 8	6.3.2 Excha Conn Settin	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules	99 <b>100</b> <b>104</b> 105
<b>Chapter 8</b> 8.1	6.3.2 Excha Conn Settin	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules g the "DCON" I/O Board I/O Function Blocks	99 100 104 105 107
<b>Chapter 8</b> 8.1	6.3.2 Excha Conn Settin Using	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules g the "DCON" I/O Board I/O Function Blocks	99 100 105 107 108
<b>Chapter 8</b> 8.1	6.3.2 Excha Conn Settin Using 8.2.1	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules If the "DCON" I/O Board I/O Function Blocks "D_7065" Function Block	99 100 105 107 108 109
<b>Chapter 8</b> 8.1	6.3.2 Excha Conn Settin Using 8.2.1 8.2.2	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules Ig the "DCON" I/O Board I/O Function Blocks "D_7065" Function Block "D_7018Z" Function Block	99 100 105 107 108 109 111
<b>Chapter 8</b> 8.1	6.3.2 Excha Conn Settin Using 8.2.1 8.2.2 8.2.3	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules bg the "DCON" I/O Board I/O Function Blocks "D_7065" Function Block "D_7018Z" Function Block "D_7083" Function Block "D_87084_FREQ" Function Block "D_87084_CNT4" Function Block	99 100 105 107 108 108 109 111 112 113
<b>Chapter 8</b> 8.1	6.3.2 Excha Conn Settin Using 8.2.1 8.2.2 8.2.3 8.2.4	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules bg the "DCON" I/O Board I/O Function Blocks "D_7065" Function Block "D_7018Z" Function Block "D_7083" Function Block "D_7083" Function Block "D_87084_FREQ" Function Block "D_87084_CNT4" Function Block "D_87084_CNT8" Function Block	99 100 105 105 107 108 109 111 112 113 114
<b>Chapter 8</b> 8.1	6.3.2 Excha Conn Settin Using 8.2.1 8.2.2 8.2.3 8.2.4 8.2.5 8.2.6 8.2.7	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules ng the "DCON" I/O Board I/O Function Blocks "D_7065" Function Block "D_7018Z" Function Block "D_7083" Function Block "D_87084_FREQ" Function Block "D_87084_CNT4" Function Block "D_87084_CNT8" Function Block "D_100T485" Function Block	99 100 105 105 107 108 109 111 112 113 114 115
<b>Chapter 8</b> 8.1	6.3.2 Excha Conn Settin Using 8.2.1 8.2.2 8.2.3 8.2.4 8.2.5 8.2.6 8.2.7 8.2.8	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules ng the "DCON" I/O Board I/O Function Blocks	99 100 105 105 107 107 108 109 111 112 113 114 115 116
<b>Chapter 8</b> 8.1	6.3.2 Excha Conn Settin Using 8.2.1 8.2.2 8.2.3 8.2.4 8.2.5 8.2.6 8.2.7 8.2.8	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules ng the "DCON" I/O Board I/O Function Blocks "D_7065" Function Block "D_7018Z" Function Block "D_7083" Function Block "D_87084_FREQ" Function Block "D_87084_CNT4" Function Block "D_87084_CNT8" Function Block "D_100T485" Function Block	99 100 105 105 107 107 108 109 111 112 113 114 115 116
<b>Chapter 8</b> 8.1 8.2	6.3.2 Excha Conn Settin Using 8.2.1 8.2.2 8.2.3 8.2.4 8.2.5 8.2.6 8.2.7 8.2.8	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules ng the "DCON" I/O Board I/O Function Blocks	99 100 105 105 107 107 108 109 111 112 113 114 115 116 118
<b>Chapter 8</b> 8.1 8.2	6.3.2 Excha Conn Settin Using 8.2.1 8.2.2 8.2.3 8.2.4 8.2.5 8.2.6 8.2.7 8.2.8 Using 8.3.1 8.3.2	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules g the "DCON" I/O Board I/O Function Blocks "D_7065" Function Block "D_7018Z" Function Block "D_7083" Function Block "D_7083" Function Block	99 100 105 105 107 108 108 109 111 112 113 114 118 118 120
<b>Chapter 8</b> 8.1 8.2	6.3.2 Excha Conn Settin Using 8.2.1 8.2.2 8.2.3 8.2.4 8.2.5 8.2.6 8.2.7 8.2.6 8.2.7 8.2.8 Using 8.3.1 8.3.2 8.3.3	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules g the "DCON" I/O Board I/O Function Blocks	99 100 105 105 107 107 108 109 111 112 113 114 115 116 118 118 120 121
<b>Chapter 8</b> 8.1 8.2	6.3.2 Excha Conn Settin Using 8.2.1 8.2.2 8.2.3 8.2.4 8.2.5 8.2.6 8.2.7 8.2.8 Using 8.3.1 8.3.1 8.3.2 8.3.3 8.3.4	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules g the "DCON" I/O Board	
<b>Chapter 8</b> 8.1 8.2	6.3.2 Excha Conn Settin Using 8.2.1 8.2.2 8.2.3 8.2.4 8.2.5 8.2.6 8.2.7 8.2.6 8.2.7 8.2.8 Using 8.3.1 8.3.2 8.3.3	EEP_WRITE (Write a Value to the FRAM) ange Data between PACs (Data Binding) ecting DCON I/O Modules g the "DCON" I/O Board I/O Function Blocks	

Chapte	er 9	On-Line Change	124
9	.1	Limitations of "On-Line Change"	124
9	.2	Using "On-Line Change"	126
Chapte	er 10	Data/Type Conversion and Using the PAC Time	.129
1	0.1	AI Data Conversion (i_scale)	129
1	0.2	AO Data Conversion (i_scale)	130
1	0.3	Type Conversion Functions (ANY_TO_xxx)	132
1	0.4	BCD Conversion (BIN _TO_BCD, BCD_TO_BIN)	133
1	0.5	Pack/Unpack Integer or Boolean	134
1	0.6	Pack/Unpack BYTE, WORD, DWORD	136
1	0.7	Unpack Variable to Byte Array or Pack Byte Array into Variable	139
1	0.8	Get/Set the PAC Time	141
Chapte	er 11	Commonly Used Tools and Useful Tips	.142
1	1.1	Upgrade Win-GRAF Libraries	142
1	1.2	Upgrade Win-GRAF Driver	143
1	1.3	Spy List	145
1	1.4	Backup/Restore a Win-GRAF Project	147
1	1.5	Soft Reboot on the PAC	149
1	1.6	Using ST Syntax in an LD and FBD Program	150
1	1.7	Apply a Recipe on the PAC	151
1	1.8	Functions and Function Blocks Supported by Win-GRAF PACs	153
1	1.9	Upload the Win-GRAF Source Code	155
1	1.10	Set Up the PAC Password	157
1	1.11	Using Function or Function Block in the ST Program	159
1	1.12	Protect Your Win-GRAF Program to Avoid Unauthorized Use	160
Chapte	er 12	Description of Win-GRAF Demo Projects	164
1	2.1	The List of Demo Programs	165
1	2.2	Timer Operations	167
		12.2.1 Start, Stop and Reset the Timer	167
		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	
1	2.3	Operations of Serial Port Communication	
		12.3.1 Send a String via a COM Port	
		12.3.2 Request-Reply Communication via a COM Port	
		12.3.3 Wait for a Remote Device to Send Data to a COM Port	
		12.3.4 Reply Data to the Remote Device Periodically via a COM Port	Т/Ω

	12.4	Read/Write Data from/to a File on the PAC	179
		12.4.1 Write Data to a File on the PAC	180
		12.4.2 Read Data from a File on the PAC	182
		12.4.3 Data Logging	185
Chap	oter 13	Using a C Program to Access Win-GRAF Variables	190
	13.1	Publishing the Win-GRAF Variable for C Program	190
	13.2	Downloading and Compiling the C Program	192
	13.3	Accessing Win-GRAF Variables	195
	13.4	Descriptions of Functions in the "Quicker.a"	196
		13.4.1 R/W a Boolean Variable	196
		13.4.2 R/W an Integer Variable	197
		13.4.3 R/W a REAL Variable	199
		13.4.4 R/W a String Variable	200
Chap	oter 14	Redundancy	201
	14.1	What Kinds of Data will Automatically Send to the Passive-PAC?	203
	14.2	Redundant System (Rotary Switch: 7 & 9)	204
		14.2.1 The Architecture of Win-GRAF Near-Field Redundant System	204
		14.2.2 Set up the Redundant System	207
		14.2.3 Test the Redundant Project (demo_rdn_2)	210
	14.3	Description of Win-GRAF Demo Projects	212
		14.3.1 [Important] "I/O Board" Settings (i_redundancy and i_redundancy_rs485)	212
		14.3.2 Declaring Variables (demo_RDN_2)	216
		14.3.3 Introduction of the "demo_RDN_2" Project	217
	14.4	The Description for Other Redundant Projects	219
		14.4.1 Test the Redundant Project (demo_rdn_3, demo_rdn_1)	219
		14.4.2 Introduction of the "demo_RDN_4" Project	223
		14.4.3 Test the Redundant Project (demo_rdn_4)	228
Chap	oter 15	Schedule Control	235
	15.1	Install the Schedule-Control Utility and Restore the Win-GRAF Project	235
	15.2	Introduction of the "demo_schedule" Project	236
		15.2.1 "I/O Boards" Setting (Schedule)	237
		15.2.2 Public the Variable Data (Data Binding)	238
	15.3	Description of the Schedule-Control Utility Example	239
	15.4	Testing the Schedule-Control on the PAC	241
	15.5	How do I Use the Schedule-Control Utility	243
		15.5.1 Address for Control Variables (BOOL, DINT, and REAL)	243
		15.5.2 Configure the Target	244
		15.5.3 Configure the Season	246

	15.5.4 Configure the Normal Day / Holiday / Special Day	248
	15.5.5 Configure the Schedule Time	250
	15.5.6 Save and Send the Configuration File	251
	15.5.7 Time Synchronization	252
Chapter 16	Win-GRAF SMS Function	253
16.1	"GSM_Open", "Send_SMS" and "Read_SMS" Functions	254
16.2	Introduction of the "Demo_SMS" Project	257
16.3	Test for SMS Messaging	260
Appendix A	A Data types and Ranges	264
Appendix B	B Troubleshooting while On-Line the PAC	265
Appendix C	C Allows the Win-GRAF to Connect via PAC's Serial Port	267
Appendix D	Pin Assignment of PAC's Serial Ports	271

# Chapter 1 Software Installation & Hardware Setting

## 1.1 Installing the Win-GRAF Workbench

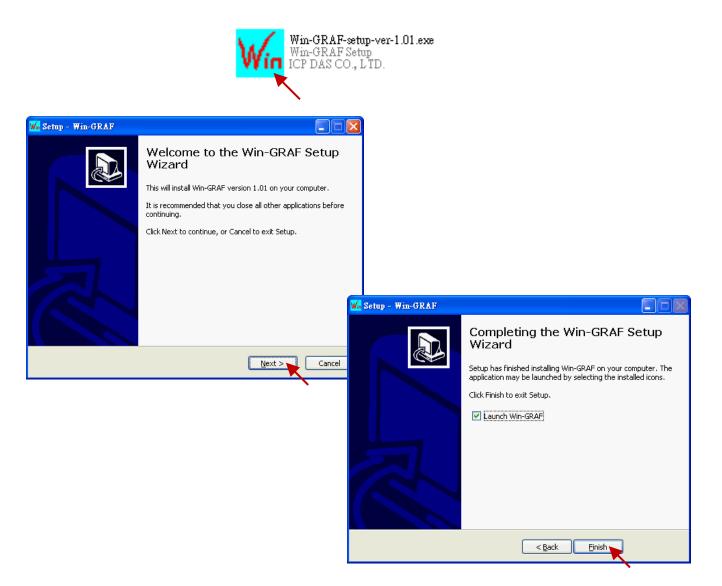
Before installing the Win-GRAF Workbench, check the installation environment on your PC.

#### System requirements:

- O.S.: Windows 7, Windows 8, Windows 10 (32-bit or 64-bit)
- Microsoft .Net Framework 3.5 (Download it on the Microsoft web site: <u>https://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:**

- Download the Win-GRAF workbench installation from the ICPDAS website. (https://www.icpdas.com/en/download/show.php?num=711, Win-GRAF-setup-ver-1.xx.zip)
- 2. Double-click the "Win-GRAF-setup-ver-x.xx.exe" setup execution file and follow the execution steps.



# 1.2 Run the Win-GRAF Workbench

Before running the Win-GRAF Workbench, make sure the USB license Key (Win-GRAF Dongle) is plugged into your PC, otherwise, the workbench will run in demo mode.

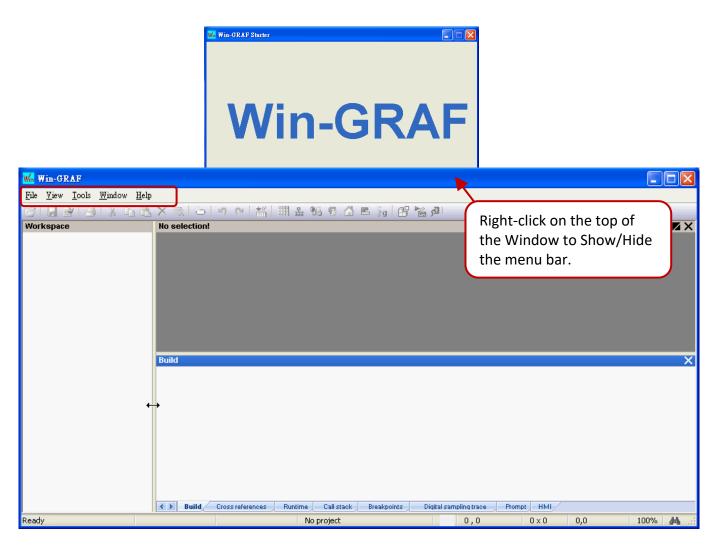
Limitations in demo mode:

- 1. Applications are limited to 40 I/Os.
- 2. The code generated by the compiler stops after 15 minutes.
- 3. The simulation stops after 15 minutes.

Start the software by clicking the "Win-GRAF" in the Start menu.



**Note:** If the USB license Key is plugged after starting the software, restart the Win-GRAF Workbench to run the fully licensed version.



## 1.2.1 Win-GRAF Operating Environment

Win-GRAF - demo\_dl\_100t485 File Edit View Insert Project Tools Window Help 5 9 6 🚽 🖹 🖂 🛛 | 🏭 🏭 🏪 😘 🕵 🟠 🛋 🗞 🔡 🎽 🖉 В Α Workspace i demo\_dl\_100t485 Name Туре 🗄 🔤 Exception programs This Demo works with Winpac COM2, baudrate 9600 + DL-100T485, ID : 1 🖃 📄 Ma Inst\_DL\_100T485 DL\_100T4 📄 Programs Inst DL 100T485 COM 한 Main 🗉 🚮 Global variables R1 EN DL 10 📲 Watch (for debugging) COM BOOL 🔊 Soft Scope RH1 REAL 2. RH1 C RH 📰 Initial values Temp\_1 REAL 鴉 Binding Configuration RETAIN variables Temp\_1 Temp C §g Global defines %IX10 - DCON 🚮 Variables 📥 Types 🗀 (All Projects) **>>** 🚞 Ethernet Powerlink -01 🚞 Files ICP DAS - XP-WP-VP ю \_3G\_CONNECT (\*Co... ID- \_3G\_DISCONNECT (... D □ \_3G\_OPTION (\*Set a ... 읭 ☐ \_3G\_READ\_CMD (\*R... COM CLEAR (\*Clear t... COM CLOSE (\*Close ... V Build Ε Build Cross references Runtime Call stack mpt HMI Ready OffLine 192, 168, 79, 12:502 0,0 121 x 18 0,0 142% 44

The main user interface (UI) of the Win-GRAF workbench is shown below.

#### A. The Workspace:

The Workbench enables you to edit several projects in the same workspace. Each project and relevant information are stored as a folder on the disk. The project list is stored in a file suffixed by ".W5L" that only stores the list of project folders and few configuration data.

#### B. The Editor Area:

Double-click the item listed in the Workspace to display the editing page. E.g., program and variables.

- C. The Variable Editor enables the declaration of variables and instances quickly during development.
- D. The **Blocks** pane provides some available functions and blocks you can use in programs. The **Spy list** pane enables the quick dynamic view of variables during debugging.
- E. The **Output** window reports messages and provides most of the diagnostic tools

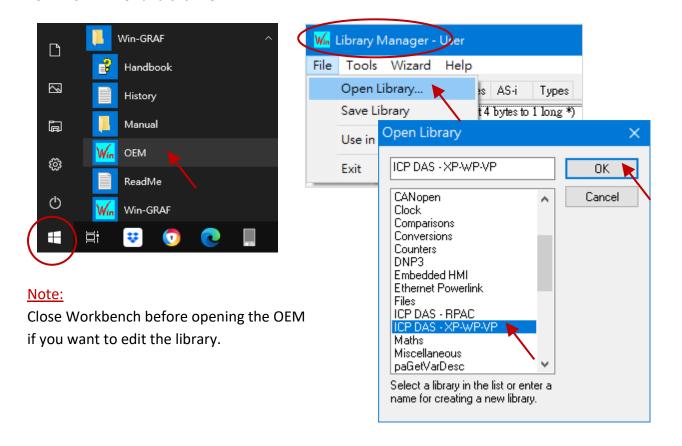
#### Tips:

- 1. Press the "F1" key where you want to get help with the function.
- 2. To resize a window, click and drag the side or corner of the window to change its size.
- 3. Click the menu command View Output or Infos Tab1 or Infos Tab2" if the pane is closed carelessly.
- 4. Click the menu command **Help Language** to change the language. The setting will be applied after automatically restarting the Win-GRAF workbench.

## 1.2.2 Win-GRAF Library Manager

To help with Functions and Function Blocks, open the Win-GRAF Library Manager.

- 1. Click the **Start** menu, and click **OEM** in the **Win-GRAF** folder.
- Click File and Open Library in the Library Manager window then select "ICP DAS XP-WP-VP" or "ICP DAS – RPAC" and click "OK".



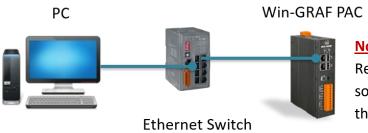
3. Click any entry under the "Function and FBs" tab and click the "Description" to view the description of this Function or Function Block.

🖌 Library Manager - ICP DAS - XP-WP-VP	—		×
File Tools Wizard Help			
Function and FBs /Os Profiles AS-i Types			
FSend_File_State (* Get the state after Send_File_To' is called . *) FSend_File_To (* Send a file to a remote PC which is running WG-File-Server *)	^	Nev	۷
B Time_Get (* Function Block: Get date and time of the PAC *)	<b>~</b>	Rena	me
Parameters Description		Dele	ete
Time_Get : Get date and time of the PAC.	^	Sto	re
*** Function Block		Reset Cl	hanges
*** Input parameters ( No input parameter )		M.	J
*** Output parameters			
Year : DINT : 2013 ~ 2999	~		
<	>		

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

Follow the steps below to configure the IP, Mask, and Gateway address of the RPAC-2658M.

#### Hardware Wiring Diagram



#### Notice:

Recommended to connect to the PC/ software (e.g., Win-GRAF) via **LAN1** with the same network segment.

- 1. Download and install the "eSearch Utility" software on the ICP DAS website.
- 2. Click the **Search Server** button to search devices on the network. Three IP settings will be displayed for one RPAC.

🥩 eSearch Utility	[v1.2.6, Dec.09, 2020 ]	The de	efault addresses		×
File Server Too	ls				
Name	Alias	IP Address	Sub-net Mask	Gateway	^
RPAC-2658M	RPAC-2658M_L1	192.168.255.1	255.255.0.0	192.168.0.254	
RPAC-2658M	RPAC-2658M_L2	172.16.255.1	255.255.0.0	172.16.0.254	
RPAC-2658M	RPAC-2658M_L3	10.0.255.1	255.255.0.0	10.0.0.254	$\checkmark$
<	1. Configuration (		eb	Exit	, ,
Status					

3. Click the **Configuration** button to set the desired field and click the **OK** button.

Configure Server (UI	DP)						×
Server Name :	RPAC-2658M						
DHCP:	0: OFF 🗸	·	Sub-net Mask :	255.255.0.0	Alia	s:	RPAC-2658M_L1
IP Address :	192.168.79.11		Gateway :	192.168.1.1	MAG	C:	00:0d:e0:c0:00:28
Warning!! Contact your Ne	twork Administrator to	o get	correct configura	tion before any changing	ļ		OK Cancel

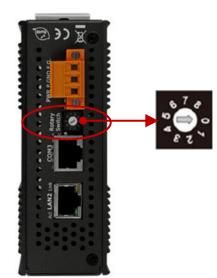
4. Search the device again to refresh the status. Note that LAN1, LAN2, and LAN3 must be set as different network segments.

File Server Tools					
Name	Alias	IP Address	Sub-net Mask	Gateway	
RPAC-2658M	RPAC-2658M_L1	192.168.79.11	255.255.0.0	192.168.1.1	
RPAC-2658M	RPAC-2658M_L2	172.16.255.1	255.255.0.0	172.16.0.254	
RPAC-2658M	RPAC-2658M_L3	10.0.255.1	255.255.0.0	10.0.0.254	$\sim$
<					>
Search Serv	ver Configuration (UD	P) We	:b	Exit	

	RUN L1	O PWR L3	
¥[U]*	LED Indicator	Color	Description
	RUN	Green	Power on and OS is running (Flashing state)
tx0 BR0 CND	PWR	Red	Power is ON
C GND	L1	Green	User programable LED & hardware error indicator
P +	L2	Orange	User programable LED
F.G 10	L3	Red	Redundancy status indicator (Active PAC)

# <u>LED 指示燈</u>

# 旋轉開關 (Rotary Switch)



<b>Rortary Switch</b>	Description
0	Normal Mode
1	Don't start user application
2	LAN1: DHCP Mode
3	LAN1: Static IP Mode (IP: 192.168.255.1, Mask: 255.255.0.0)
4	Firmware Update Mode

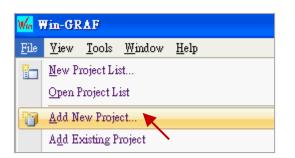
# Chapter 2 A Simple Win-GRAF Program

## 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)

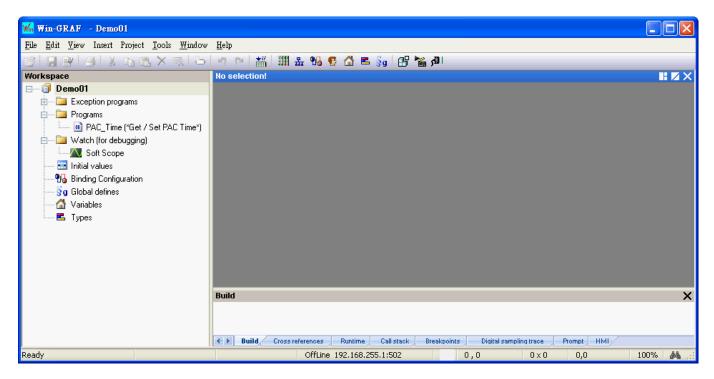
1. Run the Win-GRAF Workbench (refer to <u>Section 1.2</u>), and click the menu command "File - Add New Project...".



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 an "ICPDAS\_template" option provided by Win-GRAF Workbench, just click "Next" to continue.

🖹 Project wizar	đ	$\mathbf{X}$
Project From template XML Import Library Automation sc		
Creates a new pro	pject using a template	
New project		Recommend to use the default folder.
Destination folder	: C:\Win-GRAF\Projects	Browse
Name:	Demo01	
Comment:	Test Demo	
		Next Cancel Help
	Template: ICPDAS_	template
		Previous Next

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



**Note:** If you change to click "Project" in step2 to create a project, select the "Release" in the "Compiling options" setting and click "OK". The rest of the settings items can be set in the subsequent chapters.

Language:	LD: Ladder Diagram		~
Compiling opti	ons		
	O Debug		
(	© Release		
· · · · ·			
Communicatio	n options		
Settings:	192.168.71.19:502	"PAC IP:502",	
Protocol:	T5 Runtime	Refer to Section 2.3.5	~
11010001.			
Other			
	l values with the Recipe edito		

## 2.1.2 Important Project Settings

Two important settings must be done after creating the project.

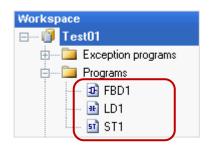
#### 1. Program Execution Sequence



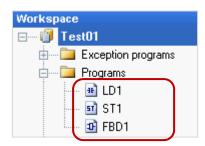
#### A. Check Program Execution Sequence

Right-click the project name (e.g., "Demo01"), select the "Alphanumeric Sorting" option to display the program name in alphanumeric order; unselect the option to display the name in execution order.

(In alphanumeric order if selected)



(In execution order if unselected)



#### **B. Modify Program Execution Sequence**

Right-click the project name (e.g., "Demo01") and click "Cycle" 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	<b>~</b>	1	0
sī pStartup	Image: A start of the start	1	0
pShutDown	<b>~</b>	1	0

### 2. Enable the "Complex variables in a separate segment" option

If the complex variables such as Array, Structures, or FB Instances will be used in the project, click the menu command "Project - Settings..." to open the Project settings window. Click the "General" option and set the "Complex variables in a separate segment" to "Yes" and then click "OK".

Proje	ct	<u>T</u> ools	<u>W</u> indow	<u>H</u> elp				
₿	Bui	ild All F	rojects		F7			
	<u>C</u> le:	an All I	Projects					
** 	Dov	wnload	All Project	s				
12333	Sett	tings						
	_							
		Рю	ject settin	igs				
		⊂:\∀	√in-GRAF\F	Projects\De	:moO1			
			neral 🦕		Name		Value	
			ntime 📉 📉 mpiler		🔊 Commu	nication parameters	192.168.2	55.1:502
		Det	jugging		🐒 Cycle ti		0	
		Ad <sup>i</sup> (All	vanced )			eneration	Release	<u> </u>
		ų	,			x variables in a separate segment	Yes	
					Con Line	e Change	Disabled	/02/25 12:17
					1 Librarie		Edit	0272512:17
						* ternal objects	Edit	
					Enables com time consum	plex data such as arrays of structures. Th ing.	is option is	OK Cancel

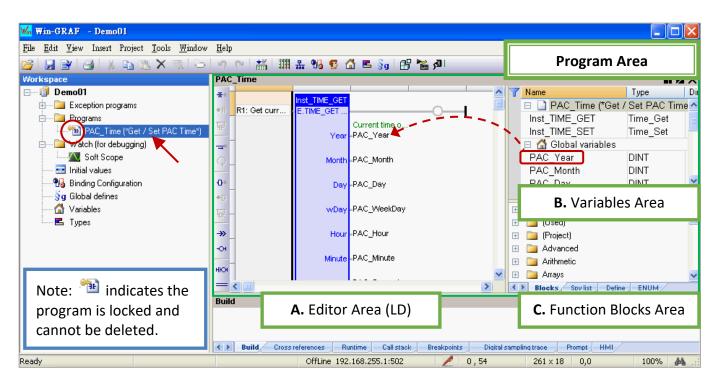
#### Note:

- 1. For more details, click the menu command **Help Topic** and search the keyword "Complex variables in a separate segment".
- 2. Make sure the Code Generation is 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:



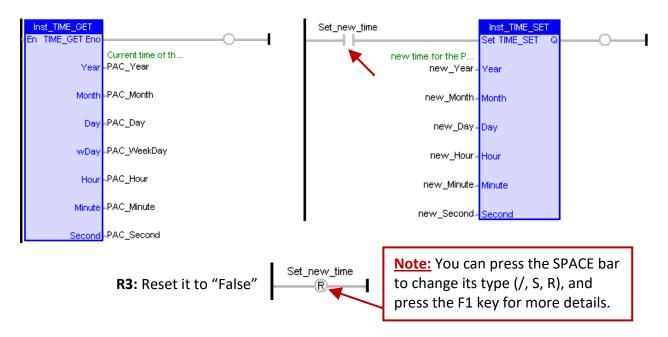
**Tips:** Click on the Editor Area and press the "+" or "-" key to zoom in or zoom out the content.

#### A. Editor Area:

This area allows you to edit the program, click the object button on the left side for programming, and assign variables by drag and drop them from the variable area.

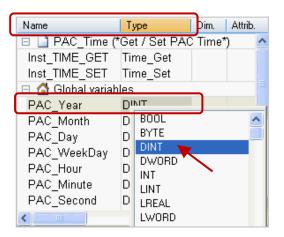
#### **R1:** Get current time of the PAC

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

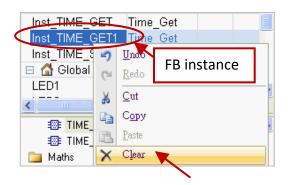


#### **B. Variables Area:**

This area shows FB instances and variables that are used in this program. Double-click any "Name" or "Type" entry to modify its name or data type, then click "Enter" to complete the setting.

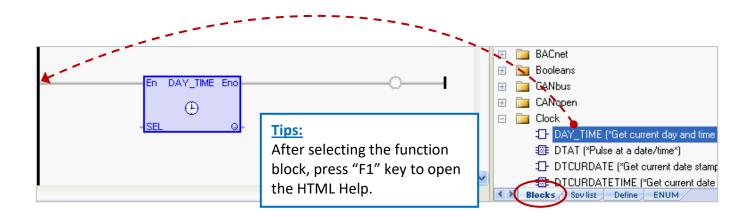


Note: For using function block (FB) in the program correctly, the instance name "Inst\_xxx.." will be automatically added when a function block is added. For safety reasons, this FB instance name will not be automatically deleted even if the function block has been removed in the program. If it is necessary, right-click the FB instance name and select "Clear" to manually remove it.



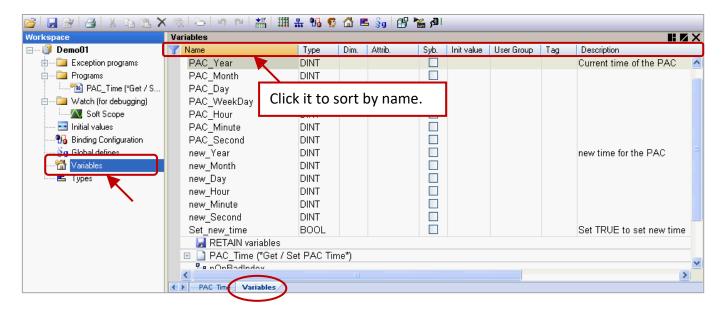
#### C. Function Blocks Area:

In the "Blocks" tab, drag and drop the specific function blocks to the editor area.



## 2.2.2 Demo01 - Variables

In the workspace, 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, click any field name (e.g., "Name") to sort data. Also, right-click anywhere and select the "Cancel Sorting" option to back to the original order.



Field description: (Press the "F1" key to look up the details)

Double-click any entry in a column 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., "_").
Туре:	Data type. (Refer to <u>Appendix A</u> for the value range)
Dim.:	To specify the range of an array. (E.g., enter "10", which means the use of the Counter [ <b>0</b> ] to [ <b>9</b> ]).
Attrib.:	Double-click this field item to set it to "Read Only" which means users can only read this variable but cannot modify it.
Syb.:	The function will be available soon. The selected variable name will be embedded in the application that can be used to access parameter data by call the name in 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

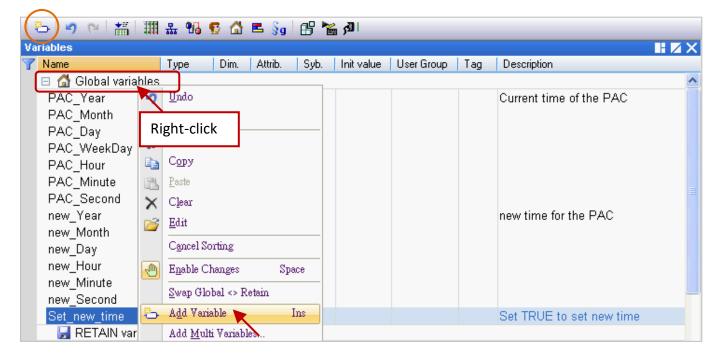
We have described the LD program (<u>Section 2.2.1</u>) and variables (<u>Section 2.2.2</u>) in the "Demo01" project. This chapter describes how to declare variables and add an LD program with the blinking function. **Note: "ULINT" and "LWORD" are not supported for Win-GRAF PAC.** 

## 2.3.1 Declaring Variables

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

1. In the "Variables" window, right-click anywhere in the "Global variables" group and select "Add

Variable" to add a variable. (Also click the icon or press the "Ins" key).



2. Double-click the "NewVar" to change the name to "LED1" and 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	ROOL	4		
LED1				

Note: 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 #1:**

To add variables in sequential order, set the name as "LED" in step2 and then press "Ctrl+C" and "Ctrl+V" many times to create "LED1" and "LED2"..., then delete the "LED" variable.

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

1. To add multiple variables in sequential order, for example, "**Bool\_01**" to "**Boo\_16**" Boolean variables. First right-click on the "Global variables" group and select the "Add Multi Variables".

Workspace	Variables	
Image: Construction of the second	Name       Type       Indo         Global variables       Redo         RETAIN variables       Redo         Main       Right-click         B pOnBadIndex       Paste         P ShutDown       Edit         D pStartup       Edit         Cancel Sorting       Enable Changes       Spe         Swap Global <> Retain.	int value

2. Set the Name as "Bool\_%%", the Type as "BOOL", and the **From** and **To** fields to 1 and 16. Then, click the Create all button to add variables.

Creation	of N variables						
Enter info	ormations						
Name	Bool_%%				Create <u>a</u> ll		
Туре	BOOL		*		<u>C</u> ancel		
Group	Global variables			~	Help		
	Read only	Dim. 0		[]			
From	1	To 16					
	From variable Bool_	Variables Name Name Boo Boo Boo Boo Boo Boo Boo Bo	Type           Global variables           L01         BOOL           L02         BOOL           L03         BOOL           L04         BOOL           L05         BOOL           L06         BOOL           L07         BOOL           L08         BOOL           L09         BOOL           L09         BOOL           L09         BOOL           L10         BOOL		Attrib. Syb.	Init value User	
		Boo Boo <	I_14 BOOL I_15 BOOL I_16 BOOL				>

## 2.3.2 Creating an LD Program

In this section, we will create an LD program with a blinking function in the "Demo01" project. To begin, follow the steps below:

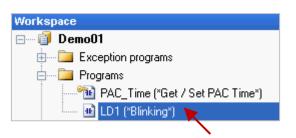
1. In the workspace, right-click the "Programs" folder and select "Insert New Program...".



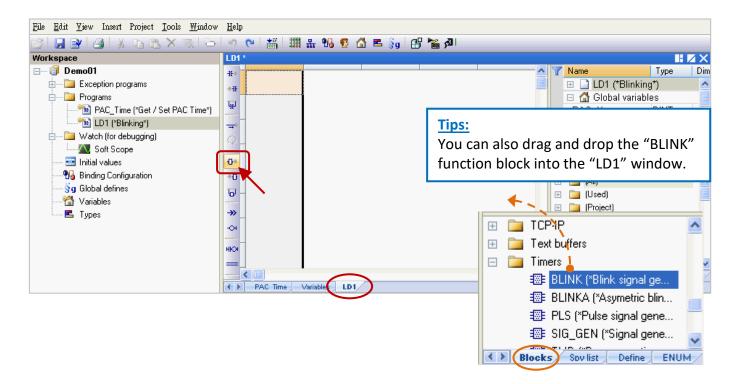
2. Fill in a program name in the "Name" field and enter a simple note in the "Description" field, and then select the "LD – Ladder Diagram" as the programming language and click the "OK" button.

New program			×						
Properties Advance	d Description								
Program	Program								
Name:	Name: LD1								
Description:	Description: Blinking								
Programming lan	Programming language								
SFC - Sequenti FBD - Function	SFC - Sequential Function Chart - Grid editor SFC - Sequential Function Chart - Free form editor FBD - Function Block Diagram								
ST - Structured IL - Instruction	LD - Ladder Diagram ST - Structured Text IL - Instruction List PACKML - PACKML State Machine								
Execution style -			í						
💽 Main progra	m								
🔾 Sub-program	n								
🔵 UDFB (Uær	Defined Function Block)								
Child SFC p	○ Child SFC program								
Child of:	Child of:								
	確定	取消 說明							

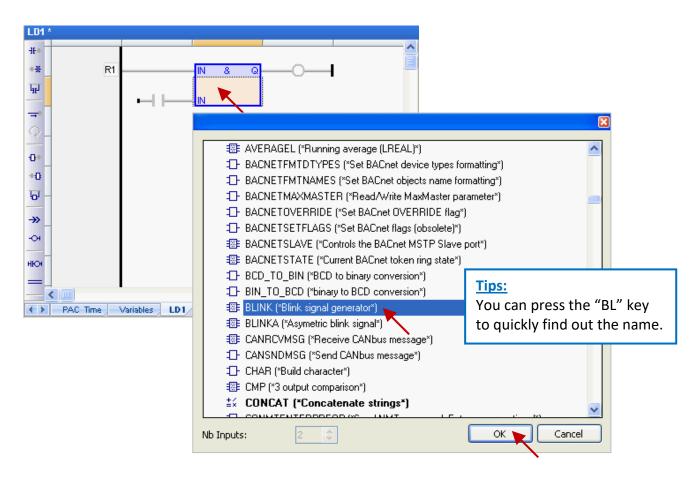
3. Double-click the "LD1" program to open the editor window.



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



5. Double-click this function block and select the "BLINK", then click the "OK" button.

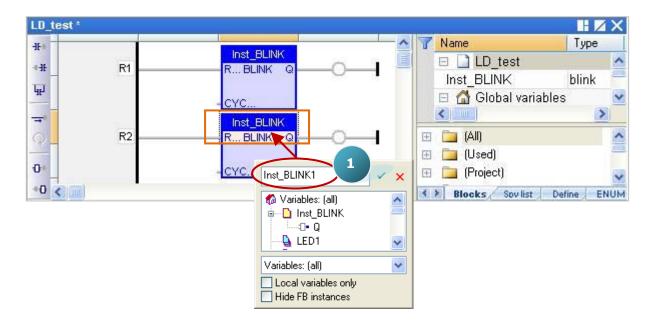




## **Important Notice:**

Users may add a function block by copy & paste an existing one during programming. But, this may cause an exception due to the same function instance name. Therefore, **users must create a new name for the copied function block.** 

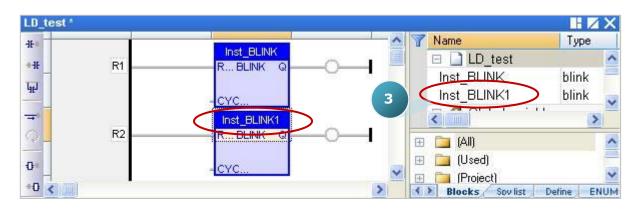
1. Double-click the function block and enter a new name (e.g., "Inst\_BLINK1"), then click the button to complete the setting.



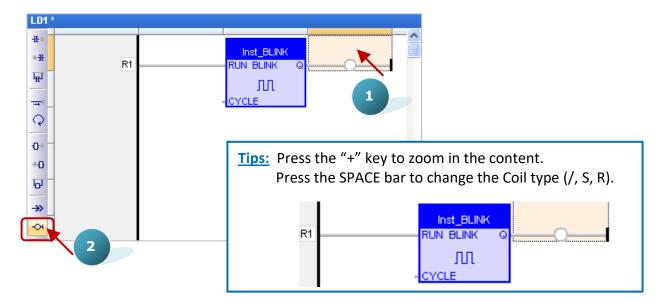
2. In the "Inst\_BLINK1" window, click "Yes" to create this function instance name.

	l does not exist. Do you want to:	
	a the variable a new variable	
×		
Type:	blink	×
Where:	LD1	~

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



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



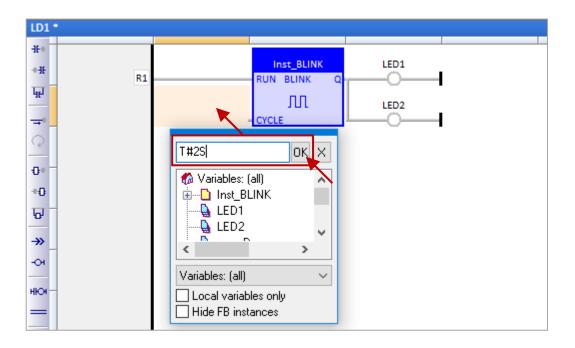
7. Double-click the first "Coil" and double-click "LED1" to assign the variable.

LD1 *	
-16-=	
++H R1	
<b>GET</b>	
	- <u>CYCLE</u>
Q_	
	LED2
+0	new_Hour 🤍
<u>ज</u> ्	<
	Variables: (all) 🗸 🗸
→> _	Local variables only
· <del>·</del>	Hide FB instances

Also, mouse drag-and-drop the "LED2" variable from the variables area.

LD1 *					
-IF+			1	Name	Туре
+- <del>1</del> E	R1	R., BLINK Q		new_Day	DINT
ц.				new_Hour	DINT
46		- CYCLE		new_Minute	DINT
		*		new_Second	DINT
Q		Ň		Set_new_time	BOOL
<u> </u>		N		LED1	BOOL
0+				LED2	BOOL
+0				🚽 RETAIN variables	

8. Double-click on the left of the "CYCLE" and enter "T#2S" (to blink every two seconds) and then click OK to finish the setting.



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

Win-GRAF - Demo01	-	- 🗆 X
File Edit View Insert Project Tools	Window Help	
🖆 🔜 🕑 🎒 🕹 🖬 🖱 🗙 🤿 🎘	>  이 연   詣   翔 品 % 😨 🖆 트 🖗   🕄 🕍 🔊	$\sim$
Workspace	LD1 *	🔣 🖌 🖌
Demo01     Exception programs     Programs     Programs     PAC_Time (*Get / Set     Soft Scope     Initial values     Global defines     Global defines     Variables     Types     (All Projects)	H   H   H   H   H   H   RUN BLINK   ILD1   H	
	Build Cross references Runtime Call stack Breakpoints Digital sampling trace Prompt HMI	
Ready	OffLine 192.168.255.1:502 🖉 0 , 108 123 x 18 0,0	120 🔬:

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

## 2.3.3 Compiling the Program

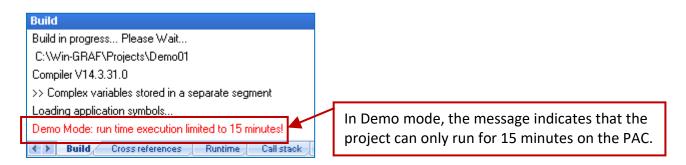
In the previous section, we have added and saved the LD program. For the Win-GRAF project can function properly on the PAC, we need to compile the programs. To begin, follow the steps below:

1. Click the menu command "Project - Build All Projects" to compile all programs.

₩a Win-GRAF - Demo	01	
<u>F</u> ile <u>E</u> dit <u>V</u> iew Insert	Proje	ect <u>T</u> ools <u>W</u> indow <u>H</u> elp
🚰   🛃 🖹 🖂   🐰	₿	Build All Projects F7
Workspace		<u>C</u> lean All Projects
Demo01     Exception pro	*** 1999	<u>D</u> ownload All Projects <u>S</u> ettings

If "No error detected" appears in the message area which means the compilation was successful.
 <u>Note:</u> If the program is modified and saved again, run the "Clean All Projects" command before compiler the program.

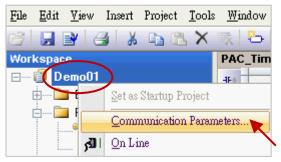
Win-GRAF - Demo01						
<u>File E</u> dit <u>V</u> iew Insert Project <u>T</u> ools	s <u>W</u> indow <u>H</u> elp					
🕝 🔛 🖻 🖂 🗼 🖻 🖄 🗡		í III 🏭 🐝 🤨	) 🕼 🖪 💡 🛛 🖓			
Workspace	PAC_Time					∎ Z X
⊡ 🗊 Demo01	-IE+			<u> </u>	7 Name	Туре
🖶 🔤 Exception programs	-H:	Inst_TIME_GET			🗆 🗋 PAC_Time (*)	Get / Set F 📥
📄 🔤 Programs	R1: Get current t	En TIME_GETEno	0		Inst_TIME_GET	Time 📃
PAC_Time (*Get / Set	95				Inst_TIME_SET	Time
📖 🏦 LD1 (*Blinking*)	<b></b> *	Var	Current time of th _PAC_Year		😑 🚮 Global variabl	
🖆 🔤 Watch (for debugging)	Q <sup>-</sup>	rear	-rAC_rear		PAC_Year	DINT
Soft Scope					PAC_Month	DINT 🧹
Initial values	0	Month	-PAC_Month			
	+()				🕀 🧰 (All)	~
🔆 🖁 g Global defines	T	Dou	-PAC Day			The second secon
Variables		Day	-1 -10_003		E D (Project)	_
IIII E Types					🗄 🧰 Advanced	
	-04	wDay	-PAC_WeekDay		🗄 🧰 Arithmetic	
	ню				🗄 🧰 Arrays	
		Hour	-PAC_Hour	~	🗄 🛅 AS-interface	~
		n nour		>	Blocks Sov list	Define ENUM
	Build				,	×
	Relocating code					~
	< Code CRC=d1b4ae39 -	File CRC=e6e6ea7f	Size=2984 > Check to	o see if t	there is any	
	No error detected				,	_
			error me	coodec.		~
	Build Cross refu	rences Runtime	Call stack Breakpoints D	iqital sampling t	race Prompt HMI	
Ready		OffLine	192.168.255.1:502	0,0	1 × 1 0,0	



## 2.3.4 Download the Program to PAC

Before downloading the program, set the Ethernet communication parameters properly. Also, refer to <u>Appendix C</u> uses the serial link.

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

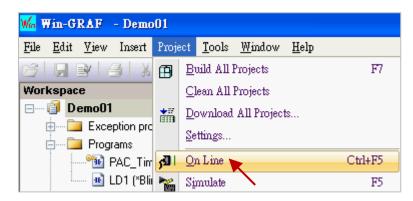


2. Enter the IP address with the TCP port number (e.g., the factory setting "192.168.255.1:502") and click "OK". Also, click the button to add/modify the IP address.

Communication Settings					
T5 Runtime 192.168.255.1:502 192.168.255.1:502	Communication Set	OK Cancel Browse	×		
		192.168.255.1 502	OK Cancel <u>H</u> elp		
	○ Serial link PC port: Baudrate: Parity:	COM15 💉 19200 💉 None 💉			
Tips: Select a IP address and delete unwanted item.					
192.168.78.8:502          192.168.255.1:502          192.168.71.19:502          192.168.78.8:502          192.168.78.8:502					

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

4. Click the menu command "**Project - On Line**" (or 🟓) to connect with the PAC.



5. If the displayed project name (TEST) is different from the current one (Demo01), click the "Stop application" button to stop running the old project. Then, click the "Download" button to download the "Demo01" project.

Win-GRAF - Demo01			1.
<u>File E</u> dit <u>V</u> iew Insert Project <u>T</u> ools	<u>W</u> indow <u>H</u> elp		
🔗   🕢 🖻   🎒   🖇 🖬 🛍 🗡 🗏	🗧 👪 🏪 (🗢 🗠 🔚 🗱 🖌	🖇 🖾 🗷 💡 🕒 📸 🚮 App: TEST	20
No application	Download Continue? 是(Y) 否(N)	Stop application Continue ? 是①	<b>王</b> 王王王王王王王王王王王王王王王王王王王王王王王王王王王王王王王王王王王

6. If "RUN" appears which means the "Demo 01" project is running on the PAC. <u>Note:</u> Refer to <u>Appendix B</u> for troubleshooting if an error message is displayed.

2   B B   3   X 🖬 🖻 X	刻 谷 9 8 番	i 🎟 🏦 😘 (	🗗 🔂 🖪 💡 🗗	📲 🚰 RUN		- 🕅 🛗 🛱 🗉 🖉	ù 🧔
Workspace	PAC_Time]						∎ Z X
⊡ Demo01 [RUN]	-It-e				<u>^</u> 7	Name	Value
Exception programs	-1:	Inst_TIME_GET				🗉 🗋 PAC_Time (	*Get / Set F 📥
🛓 🔤 Programs	R1: Get current t			—Ö—		Inst_TIME_GET	
PAC_Time (*Get / Set	·단					Inst_TIME_SET	
🔤 LD1 (*Blinking*)			Current time of th			🗉 🚮 Global variał	oles
🖨 📴 Watch (for debugging)	0-	Yea	_PAC_Year = 2014			PAC_Year	2014
Soft Scope						PAC_Month	4
🛅 Initial values	-D++	Month	-PAC_Month = 4			PAC_Day	7
🐜 📆 Binding Configuration	+0		_			PAC_WeekDay	1
🚽 🖇 🚽 Global defines	U					PAC_Hour	13
🔤 🎦 Variables		Dav	_PAC_Day = 7			PAC_Minute	55
E Types	→>>			$\sim$		PAC Second	23 🗸
	-O1	wDa	_PAC_WeekDay = 1	Diamlay the		to no time o	>
	HKO1			Display the	e sys	tem time.	~
	нкол		•		+	🛅 (Used)	
		Hour	_PAC_Hour = 13		Œ	(Project)	_
					Œ	🛅 Advanced	
		Minute	-PAC_Minute = 55		Ð	🚞 Arithmetic	
			-		Ð	🚞 Arrays	
					÷	🚞 AS-interface	
	_	Second	PAC_Second = 23	/	✓ ÷	🚞 BACnet	~
				>	4	Blocks Sov list	Define ENUM
	PAC Time Varia	ables					

RPAC-2658M User Manual, V1.0, Mar. 2021 by ICP DAS 29

#### Cycle time

When the Workbench is connected with the PAC, move the cursor over the "RUN" position on the toolbar to view the cycle time of the program on the PAC. You can also view the cycle time in the bottom-right corner of the message area.

🚾 Win-GRAF - Demo01 \_ 🗆 🗙 <u>File Edit View</u> Insert Project <u>T</u>ools <u>W</u>indow <u>H</u>elp 🚮 RUN 🛒 🗁 🕐 🖓 🏭 🛲 😘 💁 📥 😼 🖪 😭 👫 🛗 🗉 🔏 🧔 📙 🕑 🦂 👗 🖻 🖹 🗙 PAC\_Time] Workspace × Cycle time (ms): □---- ▶ Demo01 [RUN] Тур T Nam ast = 2 inst\_TIME. Allowed = 0🗄 ---- 🚞 Exception programs A( 🔨 R1: Get c... TIME\_GET +H Maximum = 36 🗄 ---- 🚞 Programs Ins Overflow = 0Current tim... PAC\_Year ... Ins Yea Local application: V21 - 09/04/2014 - 08:57:05 CRC=16#cffd8172 ---\*🎦 LD1 (\*Blinking\*) Ξ 🖞 -----PAC\_Mont... Mo. 🚊 🗝 📴 Watch (for debugging) PA 1 Target application: V21 - 09/04/2014 - 08:57:05 CRC=16#cffd8172 RAM = 75216 bytes 🔣 Soft Scope PA 1 Day PAC\_Day = 1 0+ 📰 Initial values PA [~ 📲 Binding Configuration +0 Elapsed: 15m13s < =PAC\_Week... ωDav §g Global defines + 🚞 (All) ^ 📸 Variables Hour PAC\_Hour ... 🗄 🚞 (Used) 🖪 Types 🗄 🚞 (Project) -PAC\_Minut... Mi. 🗄 🚞 Advanced 🗄 🚞 Arithmetic PAC Seco... Se. + 🚞 Arrays Inst\_TIME.. AS.interface < > < Blocks Sov list Define ENUM PAC Time Variables LD1 Demo01 - 192.168.71.19:502 # Event description RUN Time Cycle time (ms): Last = 2 Allowed = 0 Maximum = 36 Overflow = 0 Local application: < > Build Cross references Runtime all stack 🖉 🚽 Breakpoints Digital sampling trace Prompt HMI

Also, check to see if there is an error message in the Runtime window while the program is running.

## 2.3.5 Testing the Program

In the previous section, the "Demo01" project is successfully downloaded. The following will describe how to test the program.

#### The "PAC\_Time" Program:

1. Double-click the variable name (e.g., "new\_Year") in the "TIME\_**SET**" function block (or the Variables area) to change the PAC's system time.

	∎ [PAC_Time]			
-11-0	Set TRUE to set n		Name	Value
+-IE	Set_new_time = F		PAC_Month	4 🔨
	R2: Set TRUE to	Set TIME_SET Q	PAC_Day	7
됴			PAC_WeekDay	1
		new time for the	PAC_Hour	14
Q		new_Year = 0_Year	PAC_Minute	16
		new_Year	PAC_Second	41
0.			new_Year	0
+0		2014	new_Month	0 🗧
ਯ			new_Day	0
	-	<u>Force</u>	new_Hour	0
≫			new_Minute	0 🗸
-O1			<	>
ню			🚞 (All)	^
нюя		Unlock	📜 (Used)	
-	-	15 87 0	(Project)	_
-3i			🛅 Advanced	
-+ ·		31 24 23 16	🚞 Arithmetic	
	Set TRUE to set n		🚞 Arrays	
	Set_new_time = F		🚞 AS-interface	
-	R3R	✓ ±	🛅 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.

[PAC_Time]								$\overline{\Delta}$	K
-16-00	Set TRUE to set n			^	T	Name	Value		
	Set_new_time = F		Inst_TIME_SET			PAC_WeekDay	1	2	^
R2: Set TRUE to			Set TIME_SET	a		PAC_Hour	14		
FF -						PAC_Minute	30		
		new time for the				PAC_Second	37	_	
Q-		new_Year = 2015-	Year			new_Year	2015		
						new_Month	1		
0#		new_Month = 1 -	Month			new_Day	1		
+0						new_Hour	12		
T						new_Minute	30		
		new_Day = 1 =	Day			new_Second	35		
						Set_new_time	FALSE		~
-O1		new_Hour = 12-	Hour			<		>	
HICH			S	et_new	_tim	e	x		^
1107									
		new_Minute = 30-	Minute			TRUE 🍗	(1)		-
그러									
		new_Second = 35-	Second			FALSE	(0)		
:	Set TRUE to set n								
	Set_new_time = F					Lock			
R3	R							•	~
	•					Unlock		ENU	м
PAC Time Varial	bles								

RPAC-2658M User Manual, V1.0, Mar. 2021 by ICP DAS 31

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.

	e [PAC_Time]								<b>∡</b> X
-11-00					^	7	Name	Value	
IE		Inst_TIME_GET	>				new_Year	2015	^
फ़ा -	R1: Get current t	En TIME_GETEno		O			new_Month	1	
95							new_Day	1	
		Vaar	Current time of th -PAC_Year = 2015				new_Hour	12	
Q	-	rear	-FAC_16al - 2013				new_Minute	30	
0							new_Second	35	
-()=#		Month	=PAC_Month = 1			9	Set_new_time		_>
+0				<b>•</b>			LED1	FALSE	
귱		Day	-PAC_Day = 1				LED2	FALSE	
	-	Day	=FAC_Day = 1						
≫				The new sy	yste	em	time. 1.0 -	i_8055_DI	~
-O1		wDay	=PAC_WeekDay = 4						>
ню						+	🚞 (All)		^
_		Hour	-PAC_Hour = 12			+	🚞 (Used)		
	-	r iour	1110_1000 12			+	🧰 (Project)		
						+	Advanced		
-	_	Minute	-PAC_Minute = 30			+	Arithmetic		
						+	Di Arrays		
		Second	PAC_Second = 35			+	AS-interface		
	<	Gecond			⊻	÷	Di BACnet		×
* >		ables				4	Blocks Sov li	st Define E	ENUM
		/							

# The "LD1" program:

4. When the "Demo01" project is running, you can see both the LED1 and LED2 are blinking every two seconds (i.e., "T#2S") that cannot be changed. If you want to change the time, click the button again to disconnect and assign a "TIME" variable.

₩ Win-GRAF - Demo01	– 🗆 X
File Edit View Insert Project Too	ols Window Help
63 🖌 🖻 (3) 🕹 👗 🛍 🗶 来	🌤   🤊 (~   🏭   🎟 🏪 😘 😨 🟠 🛋 🗞   😷 🕍 📶 RUN 👘 00000 🎀 🏭 🛱 💷 🐒 🦉
Workspace	💶 💷 🛛 [LD1]
🖃 🕑 Demo01 [RUN]	He Name Value
<ul> <li>Exception programs</li> <li>Programs</li> <li>PAC_Time (*Get /</li> <li>LD1 (*Blinking*)</li> <li>Watch (for debugging)</li> <li>Soft Scope</li> <li>Initial values</li> <li>Binding Configuration</li> <li>Global defines</li> </ul>	+1E     Inst_BLINK     LED1=TRUE       RUN BLINK     Image: Constraint of the second seco
🖓 🖓 Variables	Build
□ Types □ (All Projects)	Build, Cross references Runtime Call stack Breakpoints Digital sampling trace Prompt HMI
Ready	RUN (192.168.79.99:502) 🔂 1,1 1 x 1 1,1

<u>Note:</u> Click the "On Line" button instead of the "Stop Application" button or the project on the PAC will stop running.

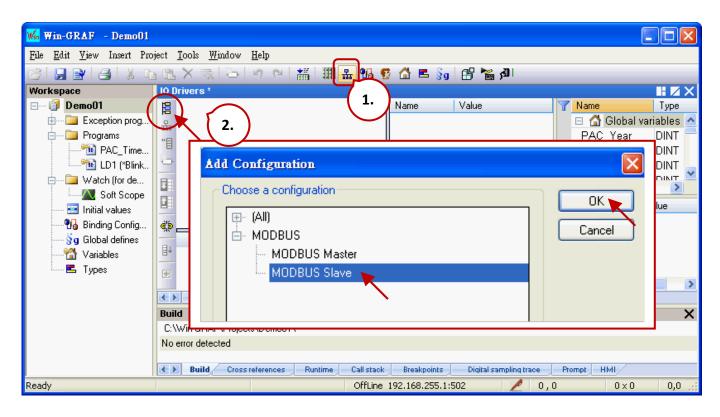
# Chapter 3 Modbus Slave: Allow the SCADA/HMI Software to Access Win-GRAF Variables

The chapter describes how to allow the SCADA/HMI software (e.g., "<u>InduSoft</u>") to access Win-GRAF variables data via Modbus **TCP or** Modbus **RTU** protocol. To begin, follow the steps below to set Win-GRAF PAC as Modbus Slave and specify variables.

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

### Setup the PAC to act as a Modbus Slave

- 1. Click the "Open Fieldbus Configuration" button on the toolbar 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.



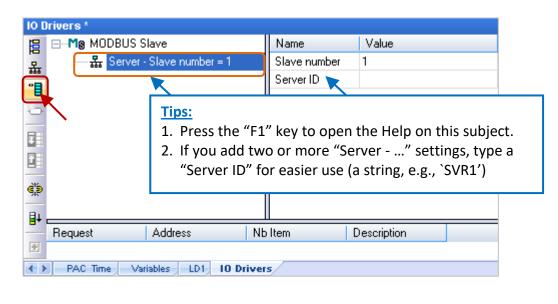
### Set the Slave ID (or Server ID)

 Click the "Insert Master/Port" button to set the "Slave number" (e.g., "1") and click "OK".

IO Drivers *		
MODBUS Slave	Name	Value
		_
MODBUS Slave Protocol	D	3
Slave number: 1	OK 🔪 Cancel	
<u>ل</u>	11	
<u></u>		
(†)		
PAC Time Variables LD1 10 Drivers		

#### Create data Block

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



#### 5. Read AI variables:

Enter a note in the "Description" field and click the "Input Registers" option. It is recommended to set "Base address" to "1" and set "Nb items" to a value greater than "200".

MODBUS Slave Request	×
Request	ОК
Description: Read_Value	Cancel
Data read by the master – Enter a simple note.	
O Input Bits	
⊙ Input <u>R</u> egisters	
Data read or forced by the master	
○ Coil Bits	
O Holding Registers	
Data block	
Base <u>a</u> ddress: 1	
<u>N</u> b items: 2000	

The "Nb items" refers to how much variables data (bits or words) can be accessed in a "data block". If the address of the data request from the Modbus Master (i.e., SCADA/HMI) is greater than this value (e.g., "2000"), the Modbus Slave (i.e., Win-GRAF PAC) will not respond.

#### Specify variables for Modbus Master to access data

6. Drag and drop the variables (e.g., "PAC\_xxx", data type: "DINT") to the address mapping area.

101	IO Drivers *								
目	MODBUS S	lave		Name	Value	7	Name	Туре	
恭	ि <mark>क्ष Server</mark> -	Slave number :	= 1	Request	Input Registers		🗉 🚮 Global vari	ables	^
**	⊞*∎ Inpu	t Registers [1]	2000] - Read_Va	Address	1		PAC_Year	DINT	
5				Nb Item	2000		PAC_Month	DINT	
-				Description	Read Value		PAC_Day	DINT	
							PAC_Hour	DINT	
	Symbol	Offset	Mask	Storage			PAC_Minute	DINT	
214	DAC M. II	0	FFFF	Default	^		PAC_Second	DINT	
ġ,þ	PAC_Day	0	FFFF	Default			PAC_WeekDay	DINT	
≣∔	PAC_Hour	0	FFFF	Default			new Year	DINT	× 1
-	PAC_Minute	0	FFFF	Default			-		_
÷	PAC_Second	0	FFFF	Default		Na	ame Valu	e	
	PAC WeekDay	0	FFFF	Default		- 1			
	<				>				

#### Specify the offset to map with Modbus address

- Double-click the "Offset" field and fill in a value, then press the "Enter" key to finish the setting. Note:
  - (1) "Offset" is a 0-based address. So, the Modbus address (1-based) is equal to the Offset plus 1.
  - (2) When using a 32-bit (or above) variable (e.g., "DINT"), it requires two consecutive addresses. 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 💌	FFFF	Default
PAC_Second	0	FFFF	Default
PAC_WeekDay	0 Enter	FFFF	Default

### Tips:

Click column header "Offset" to select entire columns and click the "Iterate Property" button on the left side to open the settings window.

1	Symbol	Offset	Mask	Storage
	PAC_Year	0	FFFF	Default
ġ,þ	PAC_Month	0	FFFF	Default
∎∔	PAC_Day	0	FFFF	Default
	PAC_Hour	0	FFFF	Default
Ð	PAC_Minute	0	FFFF	Default
	RAC_Second	0	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	\$
Results	7				
0					
2 4 6 8					
10					
12					
			ОК		Cancel

#### Set Storage for 32-bit variables

8. Click the column header "Storage" to select entire columns and press the "Enter" key to display a drop-down menu. Then, double-click "DWORD (Low – High)" to set all entries.

Symbol	Offset	Mask	Storage 🥿 🔺 Range (Low)
PAC_Year	0	FFFF	Default
PAC_Month	2	FFFF	Default
PAC_Day	4	FFFF	Default
PAC_Hour	6	FFFF	Default
PAC_Minute	8	FFFF	Default
PAC_Second	10	FFFF	Default
PAC_WeekDay	12	FFFF	Default DWORD (High - Low) DWORD (Low - High) STRING(6)
PAC Time	ariables 📃 L	D1 10 Drivers	STRING(8) STRING(10)

To expand this "Data Block" and you can see that the data will be stored by using two addresses.

10 0	)rivers *					
E	📋 📲 Inp	ut Registers [12000	] - Read_Value 🛛 🔼	Name	Value	
뮮		+01: PA0_Year		Request	Input Reg	isters
*	È 🖸	+23: PAC_Month		Address	1	
	🗖	+45: PAC_Day		Nb Item	2000	
8	🗖	+67: PAC_Hour	=	Description	Read_Val	ue
	🖸	+89: PAC_Minute				
	🖸	+1011: FAC_Seco	nd 🧮			
	····· 🗖	+1213: FAC_Weeł	(Day 🔽			
¢,5	Symbol	Offset	Mask	Storage		Range (Lo
	Symbol PAC_Year	Offset O	Mask FFFF	Storage	w - High)	Range (Lo
< € 10 10 10 10 10 10 10 10 10 10 10 10 10	-		1	-		Range (Lo
	PAC_Year	0	FFFF	DWORD (La	w - High)	Range (Lo
₽	PAC_Year PAC_Month	0 2	FFFF FFFF	DWORD (La DWORD (La	ow - High) ow - High)	Range (Lo
₽	PAC_Year PAC_Month PAC_Day	0 2 4	FFFF FFFF FFFF	DWORD (La DWORD (La DWORD (La	ow - High) ow - High) ow - High)	Range (Lo
₽+	PAC_Year PAC_Month PAC_Day PAC_Hour	0 2 4 6	FFFF FFFF FFFF FFFF	DWORD (La DWORD (La DWORD (La DWORD (La	ow - High) ow - High) ow - High) ow - High)	Range (Lo
<b>∎</b> +	PAC_Year PAC_Month PAC_Day PAC_Hour PAC_Minute	0 2 4 6 8	FFFF FFFF FFFF FFFF	DWORD (La DWORD (La DWORD (La DWORD (La DWORD (La	ow - High) ow - High) ow - High) ow - High) ow - High)	Range (Lo

### Create the second data block

- 9. To add a "Data Block" for the Modbus Master to read Boolean value. The configure way is similar to steps 4 to 8:
  - 1) Click the "Insert Slave/Data Block" button on the left side to open the settings window.
  - 2) In the "MODBUS Slave Request" window, enter a note and select the "Input Bits" option, then set "Base address" to "1" and set "Nb items" to "2000".

IO Driv	rers *
¦冒 □ 品	M® MODBUS Slave
*	⊞*■ Input Registers [12000] - Read_Value
	MODBUS Slave Request
¥≣ 1≣	Request     OK       Description:     Read_Boolean       Cancel
	Data read by the master Input Bits Enter a note
	Data read or forced by the master
	O ⊆oil Bits O <u>H</u> olding Registers
	Base <u>a</u> ddress: 1
	<u>N</u> b items: 2000

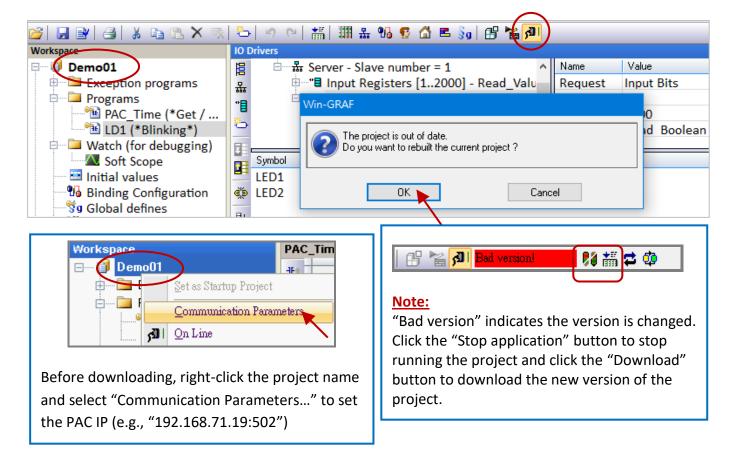
3) Drag and drop Boolean variables (i.e., "LED1" and "LED2"; data type: BOOL) to the address mapping area and set the "Offset" to "0" and to "1".

IO D	Drivers *									×
E	B MODBUS S	blave			Name	Value	Y	Name	Туре	
뮮	🖻 🚠 Server -	Slave number =	1		Request	Coil Bits		new_Second	DINT	^
*			000] - Read_Valu	Je	Address	1		Set_new_time	BOOL	
	⊡— <b>"</b> ∎ Coil	Bits [12000] -	Read_Boolean		Nb Item	2000		LED1	BOOL	
8					Description	Read Boolean		LED2	BOOL	
	······ • +	-1: LED2						<	>	Ť
	Symbol	Offset 🛆	Mask	Stor	age		Na	ame Vali	ue	
¢,5	LED1	0	FFFF	Def	ault					
	LED2	1	FFFF	Def	ault					
∎∔										

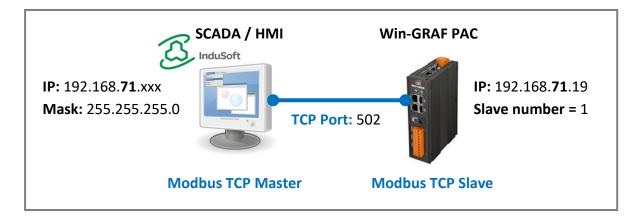
You have completed the settings for the Modbus Slave. Finally, follow the instructions below to download the program to the Win-GRAF PAC.

### **Download this project**

10. Click the "On Line" button ( ) to establish a connection and download this project to the Win-GRAF PAC. If the project has ever been compiled or changed, the dialog box below will be displayed. Click OK to re-built and download the project.



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

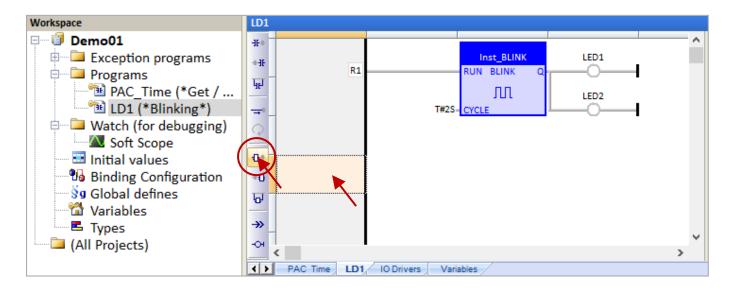


# 3.2 To Enable the Win-GRAF PAC as a Modbus RTU Slave

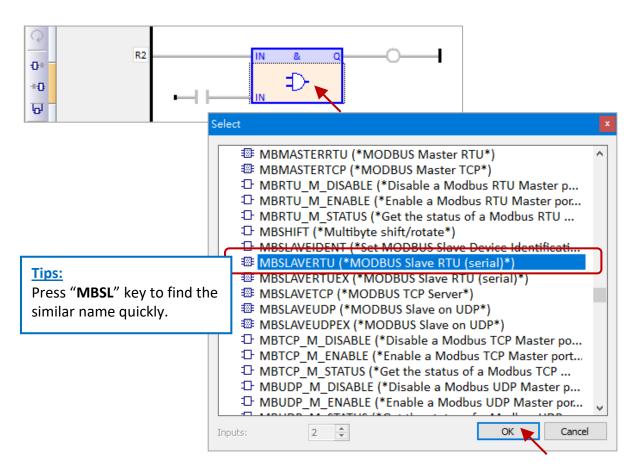
Before you get started, make sure that Win-GRAF PAC is set as a Modbus Slave and variables are set to public, refer to Section 3.1. To access data via Modbus RTU, the "MBSLAVERTU" (or "MBSLAVERTUEX") function block must be added to the program. To begin, follow the steps below:

### Add the "MBSLAVERTU" function block

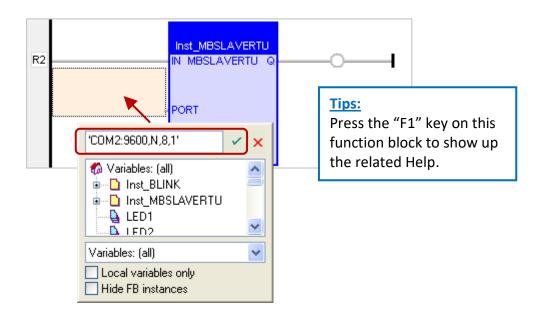
1. In the "LD1" window, click on the position where you want to add this function block and click the "Insert FB.." button on the left side of the window.



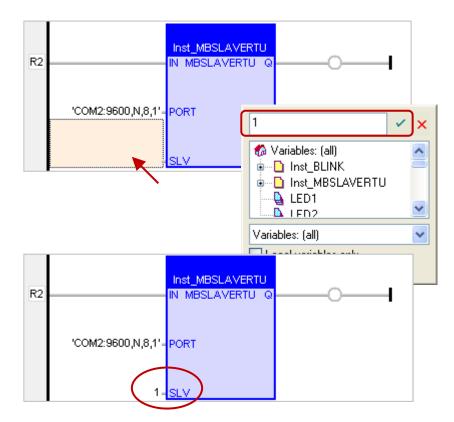
2. Double-click the function block and select the name "<u>MBSLAVERTU</u>", then click "OK".



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

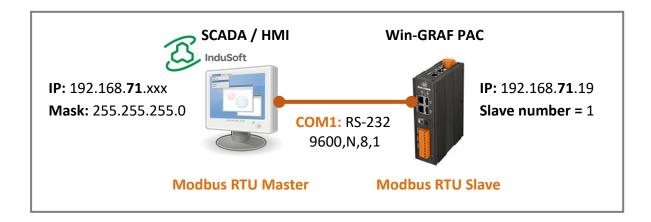


4. Double-click the left side of the "SLV" and enter "1" (the value set in <u>Section 3.1</u> - Step 3), then click
 ✓ to finish the setting.



Now, you have completed the setting of the "MBSLAVERTU" function block, and then download the project to the Win-GRAF PAC.

**Note:** Users can enable multiple Modbus RTU Slave ports on a PAC (recommend not over 16 Ports). The way is to add multiple "MBSLAVERTU" function blocks and set the different "Port" numbers.

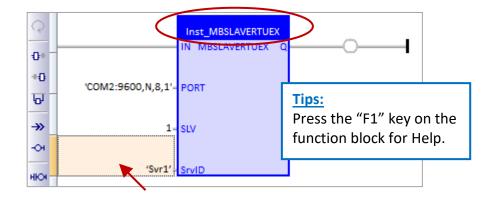


### Add the "MBSLAVERTUEX" function block

The "MBSLAVERTU**EX**" function block can be used if there are several Modbus Slave configurations (recommend to set one) in the "IO Drivers" window.

101	Orivers *		
	I M& MODBUS Slave	Name	Value
몳	🚊 🚠 Server - Slave number = 1 🍗	Slave number	1
*	Input Registers [12000] -Read_Value	Server ID	Svr1
	🗄 📲 Input Bits [12000] - Read_Boolean		
	효···· 쁆 Server - Slave number = 2		

Follow steps 1 to 4 above to add the "MBSLAVERTU**EX**" function block in the "LD1" window. Compare with "MBSLAVERTU", it adds a "SrvID" setting. Double-click on the left side of the "SrvID" and enter the Server ID (e.g., 'Svr1' with String format) that is set from the MODBUS Slave configuration.



### Note:

- With MBSlaveRTU**ex**, you can specify the ID of a server from the MODBUS Slave configuration. If you specify an empty string, the first server is used.
- MBSlaveRTU always works with the first server in the configuration.

The "MBSLAVERTUEX" function is completed by now, download the project to the Win-GRAF PAC.

# Chapter 4 Using "I/O Boards" Function

# **Note:** RPAC-2658M does not support Local I/O. For connecting with remote I/O modules, refer to Chapter 8 – Connecting DCON I/O modules.

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



2. Double-click any number to choose the needed function.

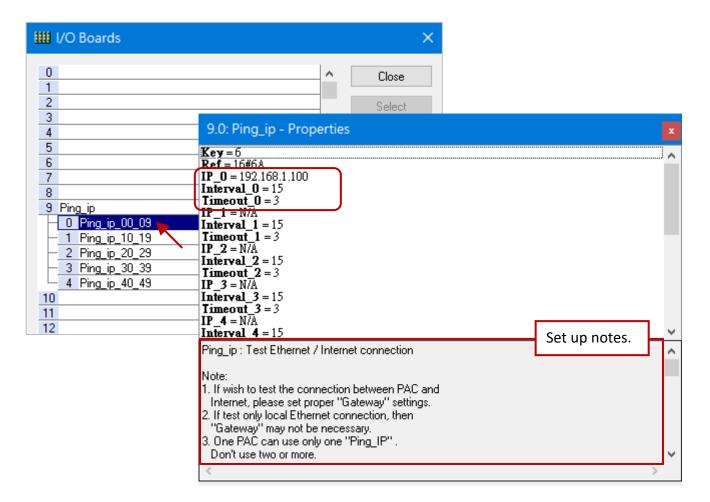
III I/O Boards			×		
0 1 2	<u>^</u>	Close			
3 4 5		Delet	е		
6 7		Renan	ne		
8		Propert	ies		
10		Virtual/F	Real		
11 12 9: 13 Library: Dev	e:				x
14         [al]         i.8           15         ICP DAS · XP·WP·VF         i.8           16         Standard         i.1           17         I.1         I.1           18         I.1         I.1           19         20         Final	084_cnt_08ch (* 8- 084_freq (* 8-ch 32- H17W (* 8-channel H ist (* test if i-8k and i- lundancy (* Enable r lundancy_rs485 (* 3 le (* Setup the scaling ip (* Test Ethernet / dule (* Enable schedu H16BI H16BO	bit frequency (ART Current 37k boards ex edundancy in Detect the RS- (function No. (nternet conne- de-control *)	t input Mo dist or not the PAC -485 port 01 to 29 action *)	: *) : state of the passive PA	^ 
Ping_ip : Test Ethernet / In Note: 1. If wish to test the connec Internet, please set prope 2. If test only local Ethernet "Gateway" may not be ne 3. One PAC can use only o Don't use two or more. 4. When Ping success_retu	ion between PAC and "Gateway" settings. connection, then cessary. e "Ping_IP" .				^ ~

Note: "i\_8xxx" or "i\_87xxx" is only available for the PAC that supports the local I/O module.

# 4.1 Ping\_ip: Check the Connection of an Ethernet/ Internet Device

The "Ping\_ip" function can be used to know if the Ethernet/ Internet connection of the remote device is working. Up to 50 IP settings can be used. Refer to <u>Chapter4</u> to add this I/O board.

1. Double-click the "0: Ping\_ip\_00\_09" to open the "Properties" window.



#### Note:

- 1. To detect the connection between PAC and the Internet, set the "Gateway" properly.
- 2. When detecting the Ethernet LAN connection, the "Gateway" setting is necessary.
- 3. Only one "Ping\_IP" can be used for each PAC.
- 4. When the connection (Ping) is successful, a boolean value "TRUE" will be returned.
- 5. When the connection (Ping) fails, it will try again. If it still fails, a boolean value "FALSE" will be returned.

#### 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 is 15 seconds by default. The valid value used in the PAC is from 6 to 86,400 seconds.

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

The unit is second. The timeout setting of the "ping" command is 3 seconds by default. The valid value used in the PAC is from 2 to 30.

- Note: The valid Interval\_x value taken in the PAC must be at least triple the Timeout\_x value. For example, if "Timeout\_0" is set as 10 however "Interval\_0" is set as 20, then the "Interval\_0" value used in the PAC will be 30.
- 2. Double-click the item to fill in the value, then press the "Enter" key to complete the setting.

9.0: Ping_ip - Properties	×
Key = 6 Ref = 16#6A IP_0 = 192.168.1.100 Interval_0 = 30 <u>Timeout_0 = 10</u> IP_1 = 192.168.78.88	^
Interval_1 = 30 Timeout_1 = 10 IP_2 = N/A Interval_2 = 15 Timeout_2 = 3 IP_3 = N/A Interval_3 = 15 Timeout_3 = 3 IP_4 = N/A Interval_4 = 15	× 8.88

- 3. After configuring the "Ping\_ip" function in the "I/O Boards" window, 50 "BOOL" Input variables will be automatically added to the "Variables" window. While connecting with PAC, the status of the connection for the remote device will be displayed.
  - TRUE: The connection is ok.
  - FALSE: Connection failed or cable problem.

Variables								X
<b>Wame</b>	Туре	Dim.	Attrib.	Syb.	Init value	User Group	Tag	Description
🗆 🚨 %IX9.0 - Ping i	00 09							^
%IX9.0.0	BOOL		Input					
%IX9.0.1=IP_dev01	BOOL		Input					
IP_dev01			Input					
%IX9.0.3								
%IX9.0.4 Double	-click to set	the r	name ai	nd press E	nter.			
%IX9.0.5	DOOL		mpac					
%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								
🗄 👼 %IX9.2 - Ping i	20 29							
🗄 📕 %IX9.3 - Ping ip	30 39							
🗄 📕 %IX9.4 - Ping ip	40 49							~
<								>

# 4.2 i\_scale: Scale Conversion

The "i\_scale" function provides up to 29 scale conversion settings to convert a physical I/O value to an engineering value. Refer to <u>Chapter4</u> to add this I/O board.

1. Double-click the "i\_scale\_0" (or "i\_scale\_1" or "i\_scale\_2") to open the "Properties" window and then to view the description.

I/O Boards	
0	Close
1 2	9.0: i_scale - Properties
3	Key=6
4	<b>Ref</b> = 16#2A
5	<b>Ch00_X0_reserved</b> = 0.0
6	<b>Ch00_X1_reserved</b> = 0.0 <b>Ch00_Y0_reserved</b> = 0.0
7	Ch00 ¥1 reserved = 0.0
8	Ch01_X0_Min_Physical_Yal=0.0 Ch01_X0_Min_Physical_Yal
9 i_scale	Ch01_X1_Max_Physical_Val=0.0 Ch01_Y0_Engineering_Val For X0=0.0
- 1 i_scale_0	Ch01_Y1_Engineering_Yal_For_X1 = 0.0
	<b>Ch02_X0_Min_Physical_Val</b> =0.0 Ch02_X1_Max_Physical_Val=0.0
10	$Ch02_X1_Max_Firstcal_Val=0.0$ $Ch02_Y0_Engineering_Val_For_X0=0.0$
11	Ch02 Y1 Engineering Val For X1 = 0.0
12	Ch03_X0_Min_Physical_Val=0.0 Ch03_X1_Max_Physical_Val=0.0
13	Setting Description
14	
	ne scaling function No. 01 to 29 for scaling I/O variables .
	le scaling function No. of to 23 for scaling f/o variables .
	ting both value of Ch_X0 and Ch_X1 to 0.0, it means the relative scaling function No. is disabled. X0 is greater than or equal to Ch_X1, the setting is wrong.

Parameters: ('Ch' stands for Ch01 to Ch29; "Ch00" is a reserved item)

Ch_X0_Min_Physical_Val:	The minimum value of an AI (or AO) module (X0).
Ch_X1_Max_Physical_Val:	The maximum value of an AI (or AO) module (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.

- Double-click the desired item and enter a value, then press the "Enter" key to set. Notice:
  - 1. If the **Ch\_X0** and **Ch\_X1** are set to "0.0", which means the scaling function No. is disabled.
  - 2. If the **Ch\_X0** is greater than or equal to **Ch\_X1**, which means the setting is wrong.
  - 3. If the **Ch\_Y0** is equal to **Ch\_Y1**, which means the setting is wrong.

For example, for converting an AI value from 4 to 20 mA to an engineering value from 0 to 10000, set Ch\_**X0** as "4.0", Ch\_**X1** as "20.0", Ch\_**Y0** as "0.0", and Ch\_**Y1** as "10000.0".

For example, for converting an AO value from -10 to +10 V to an engineering value from -50 to 1200, set Ch\_**X0** as "-10.0", Ch\_**X1** as "+10.0", Ch\_**Y0** as "-50.0", and Ch\_**Y1** as "+1200.0".

9.0: i_scale - Properties	×
Key = 6 Ref = 16#2A Ch00_X0_reserved = 0.0 Ch00_X1_reserved = 0.0 Ch00_Y0_reserved = 0.0 Ch00_Y1_reserved = 0.0	<b>∧</b>
Ch01 11 Teserved = 0.0 Ch01 X0_Min_Physical_Yal = -10.0 Ch01 X1_Max_Physical_Yal = +10.0 Ch01 Y0_Engineering_Yal_For_X0 = -50.0 Ch01 Y1_Engineering_Yal_For_X1 = +1200.0	
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	~

- 3. After configuring the "i\_scale" in the "I/O Boards" window, 30 Boolean variables will automatically be added. When connecting to the PAC, the status of the scale conversion will be displayed in the "Variables" window.
  - True: scaling function is ok.

FALSE:	scaling function	is not enabled or setting e	error.
--------	------------------	-----------------------------	--------

Description	User Tag	Init value	Syb.	Dim. Attrib.	Туре	Name
			111-10-		- i_scale_0	🖃 👹 %IX9.0
				Input	BOOL	%IX9.0.0
				Input	BOOL	%IX9.0.1
				Input	BOOL	%IX9.0.2
				Input	BOOL	%IX9.0.3
				Input	BOOL	%IX9.0.4
				Input	BOOL	%IX9.0.5
				Input	BOOL	%IX9.0.6
				Input	BOOL	%IX9.0.7
				Input	BOOL	%IX9.0.8
				Input	BOOL	%IX9.0.9
					- i_scale_1	🖻 👹 %IX9.1
				Input	BOOL	%IX9.1.0
				Input	BOOL	%IX9.1.1
				Input	BOOL	%IX9.1.2
				Input	BOOL	%IX9.1.3
				Input	BOOL	%IX9.1.4
				Input	BOOL	%IX9.1.5
				Input	BOOL	%IX9.1.6
				Input	BOOL	%IX9.1.7
				Input	BOOL	%IX9.1.8
				Input	BOOL	%IX9.1.9
					- i_scale_2	🖻 👹 %IX9.2
				Input	BOOL	%IX9.2.0
				Input	BOOL	%IX9.2.1
				Innut	BOOL	%IX9 7 7

### 4.3 RPAC\_2000\_LED: Control the L1 and L2 LED on the RPAC-2658M

The RPAC\_2000\_LED function can be used to control the L1 and L2 LED on the front panel of the RPAC-2658M.

- Note: Only L1 and L2 on the PAC can be controlled by software. Refer to <u>Chapter4</u> to add this I/O board.
- 1. Double-click the "RPAC\_2000\_LED" to open the "Properties" window and then to view the description.

IIII I/O Boards	$\times$
3 4 5 6 7 8 RPAC_2000_LED 9 10 11 11 12	8: RPAC_2000_LED - Properties Key = 7 Ref = 16#9A Reserved0 = 0 Reserved1 = 0 Reserved2 = 0 Reserved3 = 0
13 14 15 16	RPAC_2000_LED : 2-Ch LED output on the front panel of the RPAC-2000 . Note: Only L1 and L2 LED on the RPAC-2000 PAC can be controlled by software. 2-Ch Boolean outputs: Ch0 : L1 LED Ch1 : L2 LED Nov.20,2020 ICP DAS , Taiwan

#### Parameters:

**Ch0, Ch1:** (Data Type: "BOOL") indicates L1 and L2 LED.

2. After adding the "RPAC\_2000\_LED" function in the "I/O Boards" window, two Boolean variables will automatically be added. When connecting to the PAC, the status of the LED will be displayed in the "Variables" window.

Va	riables									X
T	Name	Туре	Dim.	Attrib.	Syb.	Init value	User Group	Tag	Description	
	🗆 🚨 %IX8 - RPAC 2000	LED								^
	%IX8.0	BOOL		Input						
	%IX8.1	BOOL		Input						¥
	<								>	

# 4.4 RPAC\_PAC\_state: Detect the LAN state of the RPAC-2658M

The RPAC\_PAC\_state function can be used to detect the state of the RPAC-2658M. Refer to <u>Chapter4</u> to add this I/O board.

1. Double-click the "RPAC\_PAC\_state" function to open the "Properties" window and then to view the description.

8: RPAC_state - Properties	
1       Key = 7         2       Ref = 16#P000         3       Reserved0 = 0         4       Reserved1 = 0         5       Reserved3 = 0         6       7         8       RPAC_state         9       10         11       12         13       13	
14     The following PAC support RPAC_state.       15     RPAC-2658M	^
4-Ch DINT inputs : represent the state of the RPAC Ch.0 : State_1 of This PAC Ch.1 : State_1 of Passive PAC (If not enable i_redundancy, it will always 0). Ch.2 : Reserved.	~

### Parameters:

4-Ch DINT inputs that represent the state of the RPAC PAC

- Ch.0: State\_1 of This PAC
- Ch.1: State\_1 of Passive PAC (If the i\_redundancy is disabled, it will always 0).
- Ch.2, Ch.3: Reserved.

State :

bit0 to bit2: The status of LAN1, LAN2, and LAN3. 0: OK, 1: Failed or Not wired. bit3: The status of the hardware (e.g., EEPROM, Disk, RAM). 0: OK, 1: Failed. bit4: The status of I/O slot - 0: OK, 1: Failed. (Only for local I/O slots of PAC)

2. After configuring the "RPAC\_PAC\_state" function in the "I/O Boards" window, four DINT variables will automatically be added. When connecting to the PAC, the status of the PAC will be displayed in the "Variables" window.

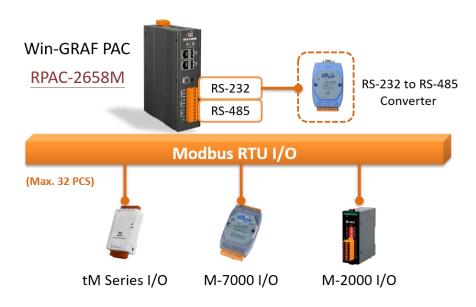
Va	riables									Х
T	Name	Туре	Dim.	Attrib.	Syb.	Init value	User Group	Tag	Description	
	🗆 📕 %ID8 - RPAC state	2								^
	%ID8.0	DINT		Input						
	%ID8.1	DINT		Input						
	%ID8.2	DINT		Input						
	%ID8.3	DINT		Input						~
	<								>	

# Chapter 5 Modbus Master: connecting to Modbus Slave Devices

This chapter lists the way to set the Win-GRAF PAC to act as a Modbus Master to connect Modbus RTU/ASCII Slave or Modbus TCP/UDP Slave devices.

### 5.1 Enabling Win-GRAF PAC as a Modbus RTU/ASCII Master

### **Application Diagram:**



#### Follow the steps below:

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

🚾 Win-GRAF - Test_01			
<u>File E</u> dit <u>V</u> iew Insert Project	Tools Window Help		
127 🖬 🕑 🛃 👗 🖻 🛍	<u>X 🔍 🏷   0 🦭 🚮 👪</u>	🖥 🕵 🚮 🛋 💱g	🕒 🕍 🎝 I
Workspace	10 Drivers		🖬 🛛 🗙
⊡ 🗊 Test_01		Name Value	Type Type
🗄 🚞 Exception programs			🚮 Global variables 🛛 🔺
🗄 🚞 Programs			🚽 RETAIN variables 📃
🖸 Main			🗋 Main 👘
🚊 🚞 Watch (for debuggi			🎦 pOnBadIndex 💳
Soft Scope			🎦 pOnDivZero 🗸
📰 Initial values			
📟 📆 Binding Configuration		п	Name Value
🔤 😽 🚽 😽 🚽 😽	<u>چ</u>		
🔤 🚰 Variables			<
E Types	Build		×
	Dette Organization	Outline I. Desited a	n District and the based of D
I	Build Cross references Runtime	Call stack Breakpoint	s _ Digital sampling trace _ Prompt _

2. Set the PAC to act as a Modbus Master. Click the "Insert Configuration" button on the left of the "IO Drivers" window and click the "MODBUS Master", then click "OK".

10.1	Orivers								4X
眉				Name Value	• 7	7 1	Vame	Туре	
	×							Global variables	^
***								RETAIN variable	s
*1		Add	l Configuration						×
		ſ	Choose a configuration —						
			⊞⊷ (All)						Ζ Ι
			⊨ MODBUS					Cancel	
¢,			- MODBUS Mas	ter 🖕					
			MODBUS Slav	e					

3. **Choose the serial port.** Click the "Insert Master/Port" button on the left side to open the settings window. Select "Serial MODBUS-RTU" and set the COM Port and the Delay time, then click OK.

### COM Port:

When using the Modbus **RTU** protocol, enter **COM2:9600,N,8,1**. When using the Modbus **ASCII** protocol, enter **ASCII:COM2:9600,N,8,1**.

### Delay (ms):

Recommended to set as 10 (ms), the value can be 0 to 10000.

	's *					X
· ···· ľ	1👩 MODBUS Maste	:r	Name Value	T Name	Туре	
	MODBUS Master	Port			Hobal variables	^
	Included Interest	Ton		and the second se	ETAIN variables! 1ain	=
-	O MODBUS on	Ethernet			OnBadIndex	
	Address:	r		Cancel	OnDivZero	~
					>	
Mo	Port:	502			Value	
× (	Pr <u>o</u> tocol:	TCP - Open MOD				>
		UDP - MODBUS R UDP - Open MOD				
	l	Ma.				
		JS-RTU				
	⊙ <u>S</u> erial MODBI <u>C</u> om. port:	JS-RTU COM2:9600,N,8,	1			
		COM2:9600,N,8,	1		ng the respond	-
	<u>C</u> om. port:	COM2:9600,N,8,	1		ng the respond nd the next cor	-
	<u>C</u> om, port: Delay between r	COM2:9600,N,8,	1		-	-
	<u>C</u> om. port: Delay between r <u>D</u> elay (ms):	COM2:9600,N,8, equests 10			-	-
	<u>C</u> om. port: Delay between r Delay (ms): ▼ Try to <u>r</u> econner	COM2:9600,N,8,		10 ms to ser	-	nmand

4. Create a data block. Click the "Insert Slave/Data Block" button to open the settings window.

10 Drivers				
E - Ma MODBUS Master	Name		Value	
문 RTU: COM2:9600,N,8,1	Mode		RTU	
	Address		COM2:9600,N,8,1	
	Port		502	
	Reconnect a	ifter error	<ul> <li>Image: A start of the start of</li></ul>	
MODBUS Master Request		ostics		
Request		n requests (ms)	10	
Description:	OK			
Slave/Unit:	Cancel			
Slave/Unit: 0 a		a. <u>Slave/U</u>	nit:	
MODBUS Request			ne Net-ID of the S	lave device.
<1> Read Coil Bits <2> Read Input Bits		b. <u>MODBL</u>	IS Request:	
<3> Read Holding Registers		The rea	d/write command	ds.
Data block		c. <u>Base ad</u>	dress:	
C Base address: 1		Start fro	om "1" by default	
Nb items: 1		<u>Nb item</u>	<u>is</u> :	
( d )		Amount	t of data to read/	write.
Activation		d. Activatio	n.	
			-	loct
On call (on error)			ing to send a requ :: Send periodical	
Misc.			pr" means how lo	-
e Timeout: 3000 ms			after an exceptio	-
Nb trials: 1		On call:		
		The req	uest is sent only v	when the
f		progran	n calls.	
Declare variables		<u>On char</u>	<u>ige</u> :	
Prefix: V% BOOL		The req	uest is sent only v	when the da
From: 1 Auto changed a	according	is chang	ged.	
v1v1 to the Request		e. Timeou	t: If there is no re	sponse in
		specific	time that means	abnormal.

f. Declaring single or several variables with consecutive numbers according to the Request and Nb items. If "Nb items" is set to 5, check the box to add V1-V5 BOOL variables.

#### Note:

- If it is necessary to change the "Base address", right-click on "MODBUS Master" and click "MODBUS Master Addresses" to modify the value.
- Also, refer to Section 5.1.6 to use the "MBRTU\_M\_disable" function in the program to disable the Modbus RTU/ASCII Master port.

IO Drivers									
	US Master 🍗								
MODBUS Master add	MODBUS Master addresses								
First valid MODBUS a	ddresses	ОК							
Input <u>b</u> its:	1	Cancel							
<u>⊂</u> oil bits:	1								
Input <u>r</u> egisters:	1								
Holding registers:	1								
		J							

(Modbus RTU/ASCII: 200 to 1000 ms)

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

Function Code	Modbus Request	Description		
2	Read Input Bits	5.1.1 <u>Read DI data</u>		
5	Write single coil bit	5.1.2 Write DO data		
4	Read Input Registers	5.1.3 <u>Read Al data</u>		
6	Write single holding register	5.1.4 <u>Write AO data (16-bit)</u>		
16	Write Holding Registers	5.1.5 <u>Write AO data (32-bit)</u>		

### 5.1.1 Read DI data

### Description:

To read 16 DI values from the device (Slave ID = 1) by sending the request every two seconds. It will be sent after 15 seconds if an exception occurs. No response for 250 ms can be regarded as abnormal.

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

MODBUS Master	ned rear			
Request	1			ОК
Description:				
<u>S</u> lave/Unit:	1	a		Cancel
MODBUS Reques	t		b	
<1> Read Coil	Bits		~	
<2> Read Inpu				
<3> Read Hold			~	
Leas load loo	IF USAICTOR	5		
Data block				
Base <u>a</u> ddress:	1			
	1000			
<u>N</u> b items:	16			
	16			
	16			
<u>N</u> b items:	16	ms	15000	
Nb items: Activation	(	ms		
Nb items:	(	ms	15000 (on error)	
Nb items: Activation	(	ms	(on error)	n is not availab
Nb items: Activation O Periodic: O On call	(	ms	(on error) This optior	
Nb items: Activation ⊙ Periodic: ○ On call ○ On change - Misc.	2000	ms	(on error) This optior	n is not availab ead" request.
Nb items: Activation Periodic: On call On change	(		(on error) This optior	n is not availab ead" request.

<u>Note:</u> Refer to step4 in Section 5.1 for the description of items.

2. Next, open the "Variables" window to add variables.

Workspace	10 Drivers		
⊡ 🗊 Test_01	📙 🖃 Mo MODBUS Master	Name	Value
🗄 🔤 Exception programs	표	Request	<2> Read Input Bits
🖨 🔤 Programs	* 2> Read Input Bits (1) [1 16]	Slave/Unit	1
🛄 편 Main		Address	1
🖻 🔤 Watch (for debuggi		Nb Item	16
Soft Scope	Tips: Press "E1" for Help	Activation	Periodic
📰 Initial values		Period (ms)	2000
📶 🚮 Binding Configuration		Period on error	15000
🚽 😽 🚽 Global defines	¢jp	Timeout (ms)	250
🚮 Variables 🛌		Number of trials	1
E Types	目+	Description	
Double-click it to open t	he window.	Offset	Mask
	<		>

"Boo\_01 to Boo\_16" (BOOL): Add 16 Boolean variables to read data.
"Status" (DINT, Dim. = 5): Add one Array variable to display the access status of data.

After completing the settings, the results are shown below.

۷	ariables									ΖX	
7	Name	Туре	Dim.	Attrib.	Syb.	Init value	User	Tag	Description		
L	🗉 🚮 Global	variables								^	
L	Bool_01	BOOL									
L	Bool_02	BOOL									
L	Bool_03	BOOL									
L	Bool_04	BOOL									
L	Bool_05	BOOL									
L	Bool_06	BOOL				D	40		000		
L	Bool_07	BOOL				Boo	_16	6	BOOL		
L	Bool_08	BOOL			Ε	Stat	us	0	DINT		
L	Bool_09	BOOL					RETA	IN var	iahlae		5
L	Bool_10	BOOL			Π		NE 17	ura var	lables		
L	Bool_11	BOOL									按 Enter
L	Bool_12	BOOL									按 Enter
L	Bool_13	BOOL			1	Boo	16	F	300L		
L	Bool_14	BOOL	_	L J	↗(		_			ſ.	• • • • • • • • • • • • • • • • • • •
L	Bool_15	BOOL				Stat	us		DINT		04]
L	Bool_16	BOOL		Ļ			/				
	<			-						_	
4	IO Drivers	Variables									

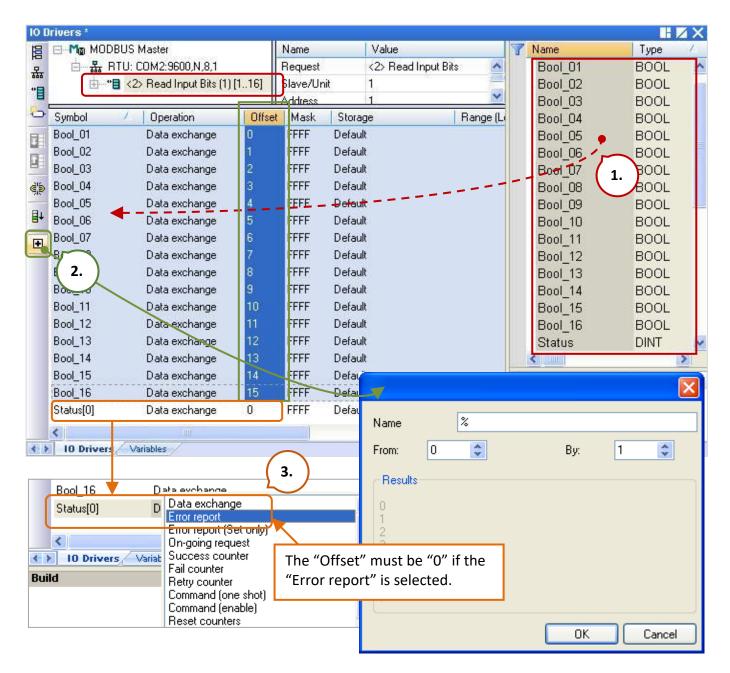
3. **Choose variables.** In the "IO Drivers" window, drag "Bool\_01" to "Bool\_16" and "Status" variables to the address mapping area of the data block.

<u>Note</u>: The "Status" is an Array variable, including Status[0] to Status[4]. Only **Status[0]** is required in the address mapping area.

4. **Configure Offset values.** Select the "Offset" field from **"Boo\_01"** to **"Boo\_16"** and click the "Iterate Property" button on the left side to set the "Offset" value (From: "0"; By: "1").

### 5. Configure Operation status.

Set the "Operation" column of "**Status[0]**" to "**Error report**" to return an error code when fails to read data. The value will be reset to "0" when a successful read.



### Also press "F1" in the "IO Drivers" window for more details on the 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	The server is busy.	130	RS-232 communication error.

### 5.1.2 Write DO Data

Description:

To write a single DO data to the device (Slave ID = 2) by sending the request when the program calls. No response for 250 ms can be regarded as abnormal.

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

	MODBUS Master	Request		X	
	Request Description: Slave/Unit:	2 a		OK Cancel	
	MODBUS Reques	ding Registers ut Registers	b		
(c	Data block Base <u>a</u> ddress: <u>N</u> b items:	1			
d	Activation O Periodic: O On call O On change	0 ms	0 (on error)		
e	Misc. <u>T</u> imeout: Nb trials:	250 ms		Refer to Section to create a dat	

2. Open the "Variables" window and add the needed variables.

Workspace	10 Drivers *		
🖃 📲 Test_01	📙 🖃 Mo MODBUS Master	Name	Value
🗄 🔤 Exception programs	品 BTU: COM2:9600,N,8,1	Request	<5> Write single coil bit
🛓 🛄 Programs	□	Slave/Unit	2
🛄 편 Main	*	Address	1
🛓 📄 🔁 Watch (for debuggi		Nb Item	1
Soft Scope		Activation	On Call
🔤 Initial values	Tips:	Period (ms)	0
📲 🚮 Binding Configuration	Press "F1" for Help.	Period on error	0
🚽 🖇 🚽 🚽 🚽 🚽	ظ	Timeout (ms)	250
		Number of trials	1
E Types	<b>■</b> +	Description	

Double-click it to open the window.

DO\_0 (BOOL): Add one Boolean variable to write DO data.

Act\_0 (BOOL): Add one Boolean variable to active the "On call" procedure.

After completing the settings, the results are shown below.

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

3. **Choose variables.** In the "IO Drivers" window, drag "DO\_0", "Act\_0" and "Status" variables to the address mapping area of the data block.

<u>Note</u>: The "Status" is an Array variable, including Status[0] to Status[4]. Only **Status[1]** is required in the address mapping area.

### 4. Configure Operation status.

- 1) Set the Operation of **Status[1]** to "**Error report**" which means to return an error code when fails to read data and be reset to 0 when a successful read.
- 2) Set the Act\_0 entry to "Command (one shot)" which means to send a request once when "Act\_0" is set to TRUE. After that, set it automatically to FALSE. "Command (Enable)" means that the request will be sent continuously when "Act\_0" is set to "TRUE". Set it to FALSE to stop sending a command.

10 E	)rivers *							HZ	iX
眉	🖃 📲 MODBUS Master	Name		Value	7	Name	Туре	Dim.	Att
뮮	🖮 🚠 RTU: COM2:9600,N,8,1	Request		<5> Write single coil bit		Bool_11	BOOL	1	^
	庄 📲 <2> Read Input Bits (1) [116]	Slave/Un	it	2		Bool_12	BOOL		
*8		Address		1		Bool_13	BOOL		
0		Nb Item		1		Bool_14	BOOL		
		Activation	E.	On Call		Bool_15	BOOL		
		Period (m	s)	0		Bool_16	BOOL		
H#		Period on	error	0		Status	DINT	[04]	
¢.		Timeout (	ms)	250		DO 0 🌻	BOOL		
		Number o	f trials	1		Act 0	BOOL		V
∎+		Descriptio	n			<		>	
<b>H</b>	Symbol Operation Off	set Mask	Stor	age – Ran	Na	ame	Value		
			-Defa				1.000		1990
	Status[1] Error report 0 D0_0 Data exchange 0	FFFF	Defa						
6	Act_0 Command (one shot) 0	FFFF	Defa						
l	Dista evoluando	EFEE	Dela						
	Error report			>					
< >	IO Drivers Varia Error report (Set only)								
	On-going request Success counter								
	Fail counter		=						
	Retry counter								
	Command (one shot) Command (enable)								
	Reset counters		_						
	Slave: last error	•							
	Slave: last error date star								
	Slave: last error time stan		~						

### 5.1.3 Read AI Data

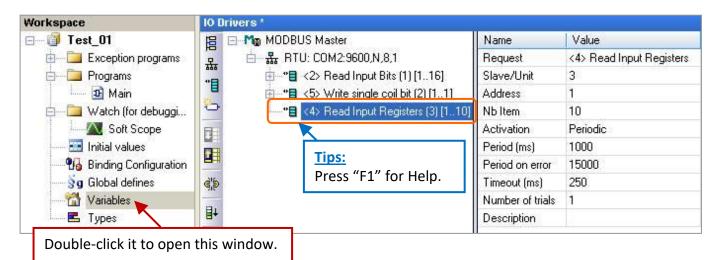
Description:

To read 10 AI data from the device (Slave ID = 3) by sending the request every second. It will be sent after 15 seconds if an exception occurs. No response for 250 ms can be regarded as abnormal.

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

MODBUS Master	Request			
Request Description: Slave/Unit:	3 a		OK Cancel	
MODBUS Reques	ut Bits ding Registers ut Registers	b		
Data block Base <u>a</u> ddress: <u>N</u> b items:	1			
Activation Periodic: On call On change	1000 ms	15000 (on error)		
Misc. Timeout: Nb trials:	250 ms		Refer to Sectic to create a dat	
	1			

2. Open the "Variables" window and add the needed variables.



Word\_1 to Word\_6 (WORD):Add six WORD (16-bit) variables to read AI data.Long\_1 (DINT):Add one DINT (32-bit) variable to read the AI data.Real\_1 (REAL):Add one REAL (32-bit) variable to read the AI data.

After completing the settings, the results are shown below.

Name	Туре	D. 🔽	Attrib.	Syb.	Init value	User	Tag	Description
Long_1	DINT		1					
VVord_1	WORD							
Word 2	WORD							
Word 3	WORD							
Word 4	WORD							
VVord_5	WORD							
Word 6	WORD							
Real_1	REAL							
<								>

### 3. Choose variables.

In the "IO Drivers" window, drag "Word\_1" to "Word\_6", "Long\_1", "Real\_1", and "Status" variables to the address mapping area of the data block.

<u>Note</u>: The "Status" is an Array variable, including Status[0] to Status[4]. Only **Status[2]** is required in the address mapping area.

Image: Symbol       Operation       Offset       Mask       Storage         Word_4       Data exchange       0       FFFF       Default       Main         Word_6       Data exchange       0       FFFF       Default       Main         Word_6       Data exchange       0       FFFF       Default       Main         Word_6       Data exchange       0       FFFF       Default       Main         Word_1       Dota exchange       0       FFFF       Default       Main         Word_1       Dota exchange       0       FFFF       Default       Main         Word_1       Data exchange       0       FFFF       Default       Main         Word_1       Data exchange       0       FFFF       Default       Main         Word_1       Data exchange       0       FFFF       Default       Main         Word_2       WORD       Word_1       Word_1       Word_1       Word_1         Word_4       Data exchange       0       FFFF       Default       Main         Word_5       Data exchange       0       FFFF       Default       Main         Word_1       Data exchange       0       FFFF	🗉 Mo Modbu	JS Master		Name	Value	7	Name	L	Туре	Dim
Symbol       Operation       Offset       Mask       Storage         Word_4       Data exchange       0       FFFF       Default         Word_5       Data exchange       0       FFFF       Default         Word_6       Data exchange       0       FFFF       Default         Word_1       Data exchange       0       FFFF       Default         Word_6       Data exchange       0       FFFF       Default         Word_1       Data exchange       0       FFFF       Default         Word_6       Data exchange       0       FFFF       Default       Name         Vord_1       Data exchange       0       FFFF       Default       Name       Value		U: COM2:9600,N,8,1		Request	<4> Read Input Regis		DO O		BOOL	~
Symbol       Operation       Offset       Mask       Storage         Word_4       Data exchange       0       FFFF       Default         Word_5       Data exchange       0       FFFF       Default         Word_6       Data exchange       0       FFFF       Default         Word_1       Data exchange       0       FFFF       Default         Word_5       Data exchange       0       FFFF       Default         Word_6       Data exchange       0       FFFF       Default         Word_1       Data exchange       0       FFFF       Default         Word_5       Data exchange       0       FFFF       Default         Word_1       Data exchange       0       FFFF       Default         Word_1       Data exchange       0       FFFF       Default         Word_6       Data exchange       0       FFFF       DWORD (Low · High)         DWORD (Low · High)       DWORD (Low · High)       2	III	<2> Read Input Bits (1) [1	16]	Slave/Unit	3		Long_1		DINT	
Activation       Periodic         Period (ms)       1000         Period on err       15000         Timeout (ms)       250         Number of tri       1         Description       1         Word_4       Data exchange       0         Vord_5       Data exchange       0         Vord_6       Data exchange       0         Long_1       Data exchange       0         FFFF       Default         WORD       Low - High)         Real       Data exchange       0		<5> Write single coil bit (2	2) [1.]	Address	1		Real_1		REAL	
Activation       Periodic         Period (ms)       1000         Period (ms)       1000         Period on err       15000         Timeout (ms)       250         Number of tri       1         Description       1.         Word_4       Data exchange         Word_5       Data exchange         Vord_6       Data exchange         Uong_1       Data exchange         0       FFFF         Default       Name         Name       Value	* 	<4> Read Input Registers	; (3) [	Nb Item	10		Status		DINT	[C
Period (ms)       1000         Period on err       15000         Timeout (ms)       250         Number of tri       1         Description       1.         Word_4       Data exchange       0         Vord_5       Data exchange       0         Vord_6       Data exchange       0         Long_1       Data exchange       0         PFFF       Default         WORD_1       WORD         WORd_6       Data exchange       0         Data exchange       0       FFFF         Default       Name       Value	1		Ĩ	Activation	Periodic		Word 1		WORD	0
Fellod of ell       1 5000         Timeout (ms)       250         Number of tri       1         Description       1.         Word_4       Word_5         Word_4       Description         Word_4       Data exchange         0       FFFF         Word_5       Data exchange         0       FFFF         Default       Main         Name       Value         Name       Value				Period (ms)	1000		Word 2		WORD	
Number of tri       1       Word_5       WORD         Description       0       0ffset       Mask       Storage         Word_4       Data exchange       0       FFFF       Default       Main         Word_5       Data exchange       0       FFFF       Default       Name       Value         Word_6       Data exchange       0       FFFF       Default       Name       Value         Long_1       Data exchange       0       FFFF       DWORD (Low - High)       Z	Ħ			Period on err	15000		Word_3		WORD	
Number of tri       1       Word_5       WORD         Description       1.       Word_6       WORD         Symbol       Operation       Offset       Mask       Storage         Word_4       Data exchange       0       FFFF       Default       Main         Word_5       Data exchange       0       FFFF       Default       Name       Value         Word_6       Data exchange       0       FFFF       Default       Name       Value         Long_1       Data exchange       0       FFFF       DWORD (Low - High)       Z         Real_1       Data exchange       0       FFFF       DWORD (Low - High)       Z	> -			Timeout (ms)	250	1	Word 4		WORD	
Description       Word_6       WORD         Symbol       Operation       Offset       Mask       Storage         Word_4       Data exchange       0       FFFF       Default       Main         Word_5       Data exchange       0       FFFF       Default       Name       Value         Word_6       Data exchange       0       FFFF       Default       Name       Value         Long_1       Data exchange       0       FFFF       DWORD (Low - High)       Z         Real_1       Data exchange       0       FFFF       DWORD (Low - High)       Z				Number of tri	1 (1	$\sum$	Word 5		WORD	
Symbol     Operation     Orrset     Mask     Storage       Word_4     Data exchange     0     FFFF     Default       Word_5     Data exchange     0FEFF_     Default       Word_6     Data exchange     0     FFFF     Default       Long_1     Data exchange     0     FFFF     DWORD (Low - High)       Real_1     Data exchange     0     FFFF	•			Description		)			WORD	
Word_4       Data exchange       0       FFFF       Default         Word_5       Data exchange       0      FEFF_       Default       Name       Value         Word_6       Data exchange       0       FFFF       Default       Name       Value         Long_1       Data exchange       0       FFFF       DWORD (Low - High)       2         Real_1       Data exchange       0       FFFF       DWORD (Low - High)       2	Symbol	Operation	Offs	et Mask 9	Storage		the second second second second second	AIN V	/ariables	~
Word_5       Data exchange       0FEFF_       Default       Name       Value         Word_6       Data exchange       0       FFFF       Default       Name       Value         Long_1       Data exchange       0       FFFF       DWORD (Low - High)       Provide the second se	Word_4	Data exchange	0	FFFF D	efault		A CONTRACTOR OF A CONTRACTOR OFTA CONT			>
Word_6     Data exchange     0     FFFF     Default       Long_1     Data exchange     0     FFFF     DWORD (Low - High)       Real_1     Data exchange     0     FFFF     DWORD (Low - High)	Word_5	Data exchange	0	FEFF D	efault			14		1
Real_1 Data exchange 0 FFFF DWORD (Low - High)	Word_6	Data exchange	0	FFFF D	efault	INA	ame	V	alue	_
	Long_1	Data exchange	0	FFFF D	WORD (Low - High)					
Status[2] Error report D FFFF Default 2.	Real_1	Data exchange	0	FFFF D	WORD (Low - High)					
	Status[2]	Error report	þ	FFFF D	efault	<u>.</u> )				

### 4. Configure Storage format.

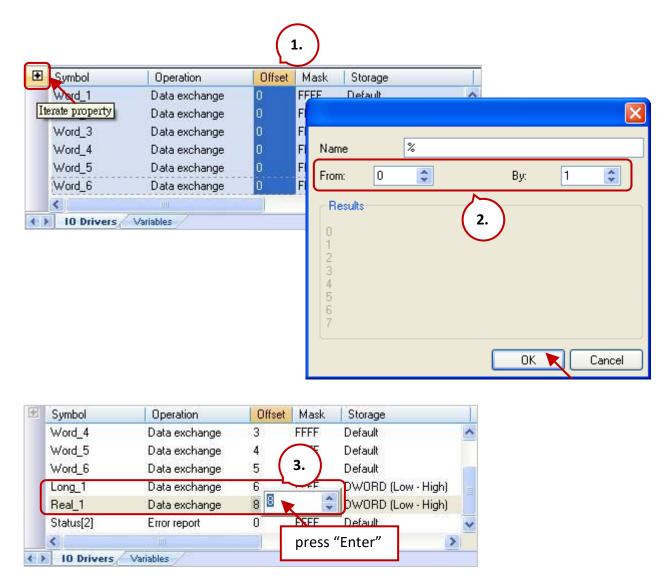
Set the Storage of Long\_1 and Real\_1 (32-bit) to "DWORD (Low – High)", mapping two Word (16-bit).

### 5. Configure Operation status.

Select the Operation of **Status[2]** to "**Error report**" which means to return an error code when fails to read data and reset to 0 when a successful read.

### 6. Configure Offset values.

- 1) Select the "Offset" field from "Word\_1 to Word\_6" and click the "Iterate Property" button on the left side to set the "Offset" value (From: "0"; By: "1").
- 2) Double-click the Offset field of "Long\_1" (32-bit) and fill in "6", then press the "Enter" key.
- 3) Double-click the Offset field of "Real\_1" (32-bit) and fill in "8", then press the "Enter" key.



Note: Two Modbus addresses are required for 32-bit data.

### 5.1.4 Write AO Data (16-bit)

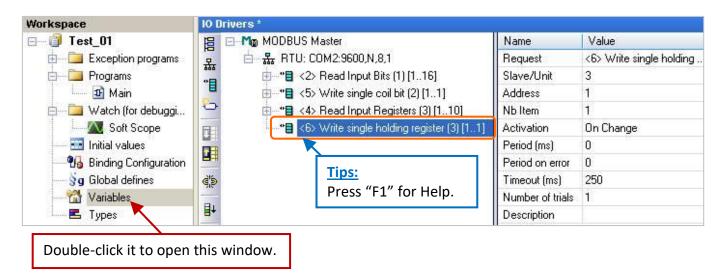
### Description:

To write a single AO data to the device (Slave ID = 3) by sending the request when the data is changed. No response for 250 ms can be regarded as abnormal.

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

MODBUS Master	r Request		X	
Request Description: Slave/Unit:	3 a	)	OK Cancel	
MODBUS Reque	out Registers gle coil bit gle holding register	b		
C Data block Base <u>a</u> ddress: <u>N</u> b items:	1			
d Activation Periodic: On call On change	0 ms	0 (on error)		
e Misc. Timeout: Nb trials:	250 ms		Refer to Section to create a data	
ND (rials:				

2. Open the "Variables" window and add the needed variables.



Word\_Write\_1 (WORD): Add one WORD (16-bit) variable to write the AO data.

After completing the settings, the results are shown below.

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

3. **Choose variables.** In the "IO Drivers" window, drag "Word\_Write\_1" and "Status" variables to the address mapping area of the data block.

Note: The "Status" is an Array variable, including Status[0] to Status[4]. Only **Status[3]** is required in the address mapping area.

### 4. Configure Operation status.

Set the **Status[3]** entry to "**Error report**" which means to return an error code when fails to read data and reset to 0 when a successful read.

10 0	lrivers *								H J	XΝ
围		JS Master		Name	Value	7	Name	$\nabla$	Туре	
뮮	📥 🏭 🚠 RTU	J: COM2:9600,N,8,1		Request	<6> Write single holdi	2	🖽 🚮 Glob:	al variat	oles	^
	<b>*</b> ∎	<2> Read Input Bits (1	) [116]	Slave/Unit	3		Word_Wri	ite_1	WORD	
"目	÷*=	<5> Write single coil b	it (2) [11]	Address	1		Word 6	-	WORD	
0	÷*	<4> Read Input Regis	ters (3) [110]	Nb Item	1		Word_5		WORD	
	±+∎	<6> Write single holdir	ig register (3) [1	1) Activation	On Change		Word_4	3	WORD	
				Period (ms)	0		Word_3	8	WORD	
				Period on err	0		Word_2	8	WORD	
ġ.				Timeout (ms)	250		Word 1		WORD	_
- 22				Number of tri	1		Status		DINT	
∎+				Description	1.	)	Dool 1		DEAL	X
			1					-		
	Symbol	Operation		ask Storage	Range	(Lc N	ame	Valu	e	_
	Word_Write_1	Data exchange		F Default						
	Status[3]	Error report	0 FFF	FF Default						
	1201		2.			am.				
-	<	,101	2.			>				
< >	10 Drivers	Variables								

### 5.1.5 Write AO Data (32-bit)

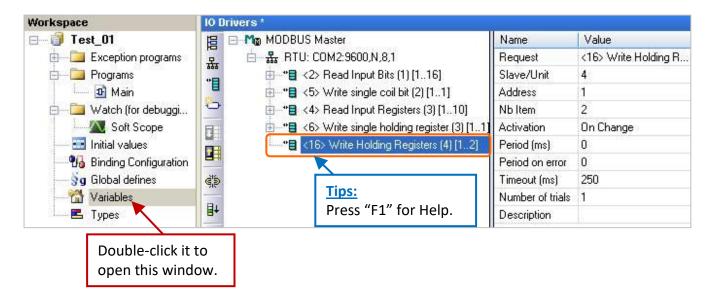
### **Description**:

To write a single AO data to the device (Slave ID = 4) by sending the request when the data is changed. No response for 250 ms can be regarded as abnormal.

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

MODBUS Master R	lequest			
Request         Description:         Slave/Unit:	4 a	)	OK Cancel	
<15> Write Coil <16> Write Hold Data block Base <u>a</u> ddress:		b		
d Activation	0 ms	0 (on error)		
Turesar	250 ms		Refer to Sectio to create a data	

2. Open the "Variables" window and add the needed variables.



Real\_Write\_1 (REAL): Add one REAL (32-bit) variable to write AO data.

After completing the settings, the results are shown below.

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

3. **Choose variables.** In the "IO Drivers" window, drag "Word\_Write\_1" and "Status" variables to the address mapping area of the data block.

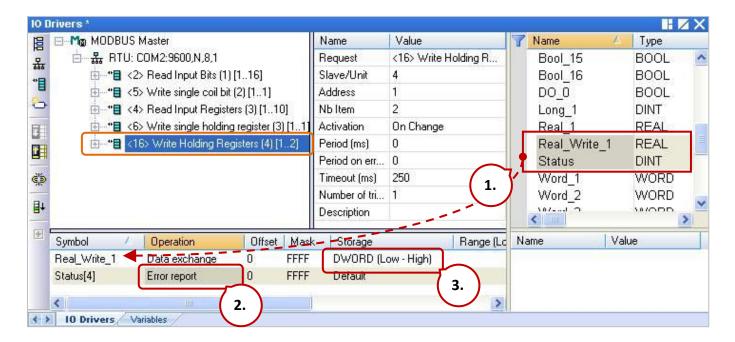
<u>Note</u>: The "Status" is an Array variable, including Status[0] to Status[4]. Only **Status[4]** is required in the address mapping area.

### 4. Configure Operation status.

Set the Operation of **Status[4]** to "**Error report**" which means to return an error code when fails to read data and reset to 0 when a successful read.

### 5. Configure Storage format.

Set the Storage of Real\_Write\_1 (32-bit) to "DWORD (Low – High)", mapping two Word (16-bit).



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

The Modbus RTU/ASCII Master port that is enabled in the Win-GRAF "Fieldbus Configuration" - "IO Drivers" window will be activated automatically whenever the PAC is powered on.

If the user wants to disable the Modbus **RTU/ASCII** Master port while the program is running, using the "**MBRTU\_M\_disable**" function as follows.

(\* Declare To\_disable as BOOL \*)
If To\_disable then
To\_disable := FALSE ;
MBRTU\_M\_disable(2) ;
End\_if;

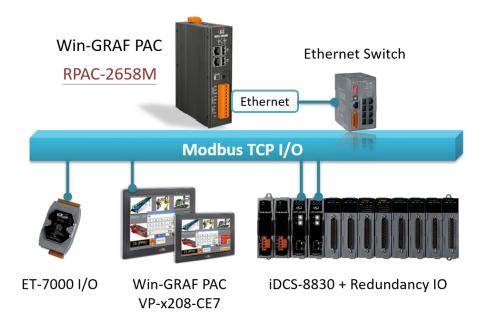
As with the above codes, when "To\_disable" is set to "TRUE", the Modbus RTU/ASCII Master port (i.e., COM2) will be disabled. Also, using the **"MBRTU\_M\_enable"** function to enable it as follows.

(\* Declare To\_enable as BOOL Declare Status\_com2 as BOOL \*) If To\_enable then To\_enable := FALSE ; MBRTU\_M\_enable(2) ; End\_if; Status\_com2 := MBRTU\_M\_status(2) ;

The **"MBRTU\_M\_status"** function listed above is used to get the status of the Modbus RTU/ASCII Master port. True: Enabled; False: Disabled.

# 5.2 Enabling the Win-GRAF PAC as a Modbus TCP/UDP Master

### **Application Diagram:**



### Follow the steps below:

- 1. Click the tool icon "Open Fieldbus Configuration" to open the "IO Drivers" window.
- 2. Set the PAC to act as a Modbus Master. Click the "Insert Configuration" button on the left of the "IO Drivers" window and click the "MODBUS Master", then click "OK".

Wm Win-GRAF - Test_01				
<u>File E</u> dit <u>V</u> iew Insert Project	<u>T</u> ools <u>W</u> indow <u>H</u> elp			
🖆   🛃 🖻   🛃   🕹 🗈 🛱	४ 🕵 🗁 🗠 🖂 🚟 👪 🔚	😼 😨 🟠 🛋 💱 🗄	皆 🕍 🖓	
Workspace	10 Drivers			H 🛛 X
⊡ 🗃 Test_01		Name Value	🍸 Name	Туре
🗄 🚞 Exception programs			🔄 🚮 Globa	al variables 🛛 🔺
e Programs	Add Configuration		X	IN variables
<ul> <li>□ Watch (for debuggi</li> <li>Soft Scope</li> <li>□ Initial values</li> <li>□ Binding Configuration</li> <li>○ g Global defines</li> <li>□ Variables</li> </ul>	Choose a configuration		OK Cancel	adIndex ivZero v Value
IIII Types	Build			×
	Build Cross references Runtime	Call stack Breakpoints	Digital samplin	ng trace Prompt

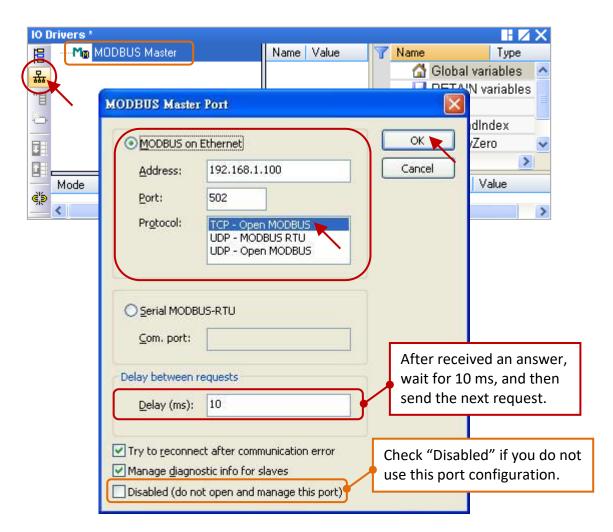
### <u>Note</u>:

The Modbus Master configuration allows using several Modbus TCP/UDP or Modbus RTU/ASCII ports. If you want to disable the Modbus TCP/UDP port while the program is running, refer to Section 5.2.4.

3. **Choose the Ethernet port.** Click the "Insert Master/Port" button on the left side to open the settings window. Select "MODBUS on Ethernet" and set the Address, Port, Protocol, and the Delay time, then click OK.

Address:	Enter the IP Address of the Modbus Slave device (e.g., "192.168.1.100").
<u>Port</u> :	TCP port number of the Modbus Slave device.
<u>Protocol</u> :	When using Modbus TCP, select the "TCP – Open MODBUS".
	When using Modbus UDP, select the "UDP – Open MODBUS".

<u>Delay</u>: Enter the delay time of the request (e.g., 10 ms, can be 0 to 10000).



4. Create a data block. Click the "Insert Slave/Data Block" button to open the settings window.

10	Drivers *		
臣	⊡Mg MODBUS Master	Name	Value
노	<b>윲</b> Open MODBUS: 192.168.1.100:502	Mode	Open MODBUS
		Address	192.168.1.100
		Port	502
		Reconnect after error	
T		Slaves diagnostics	✓
		Delay between requests (ms)	10
		Disabled	

### Read AI Data

To read 4 AI data from the device (Slave ID = 1) by sending the request every second. It will be sent after 15 seconds if an exception occurs. No response for 1000 ms can be regarded as abnormal.

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

Request	-11-				
Description:		_		ОК	
<u>S</u> lave/Unit:	1	a		Cancel	
MODBUS Reques	t		b		
<2> Read Inpo <3> Read Hold	ut Bits ding Registr	erc	^		
<4> Read Inp	ut Register				
ZEN Weite sins	alo coil bit	~			
Data block	1		Rofor	to Section 5.1	- Stop / for
Base <u>a</u> ddress:	1				
<u>N</u> b items:	4		more	details about t	the data bio
Activation	· Le				
• Periodic:	1000	ms 1	5000		
On call	1000		1005		
On change		(0	n error)		
Misc.					
	1000	me	_	a	
Timeout:	1000	ms		mmended to se	
			40	00 to 3000 ms.	

2. Open the "Variables" window and add the needed variables.

Workspace	IO Drivers *	1	10000
E Test_2	E ⊡-Mo MODBUS Master	Name	Value
🖮 🚞 Exception programs	品 Open MODBUS: 192.168.1.100:502	Request	<4> Read Input Registers
📥 🦳 Programs	**** <4> Read Input Registers (1) [14]	Slave/Unit	1
💷 🖭 Main		Address	1
🖮 🛅 Watch (for debuggi		Nb Item	4
Soft Scope	Tips:	Activation	Periodic
📰 Initial values		Period (ms)	1000
📶 🚮 Binding Configuration		Period on error	15000
🦳 😽 🚽 Global defines	di s	Timeout (ms)	1000
🚮 Variables 🦕		Number of trials	1
E Types	<b>■</b> +	Description	

"Word\_1 to Word\_4" (WORD): Add 4 WORD (16-bit) variables to read AI data. "Status" (DINT, Dim.=5): Add one Array variable to display the status for accessing data. After completing the settings, the results are shown below.

Name	Д Туре	Dim.	Attrib.	Syb.	Init value	User	Tag	Description
Status	DINT	[04]				1		
Word_1	WORD	1970 34						
Word_2	WORD							
Word_3	WORD							
Word_4	WORD							
<						ľ		

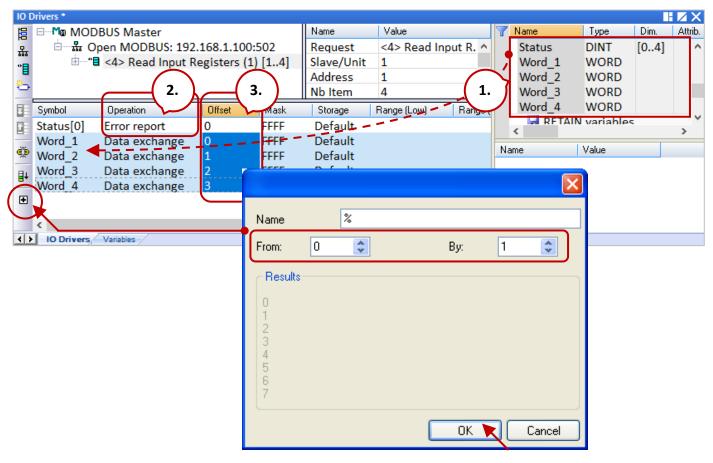
3. **Choose variables.** In the "IO Drivers" window, drag "Word\_1 to Word\_4" and "Status" variables to the address mapping area of the data block.

<u>Note</u>: The "Status" is an Array variable, including Status[0] to Status[4]. Only **Status[0]** is required in the address mapping area.

#### 4. Configure Operation status.

Set the Operation of the "Status[0]" to "Error report" which means to return an error code when fails to read data and reset to 0 when a successful read.

5. **Configure Offset values.** Select the "Offset" field from **"Word\_1"** to **"Word\_4"** and click the "Iterate Property" button on the left side to set the "Offset" value (From: "0"; By: "1").



#### Note:

Using a similar way to create the data block to read/write IO data, also refer to Section 5.1.1 to 5.1.5.

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

The ET-7000 series is a web-based Ethernet I/O module. Win-GRAF PAC can be set as a Modbus TCP Master to connect several ET-7000 modules. It is recommended that one RPAC-2658M can connect up to 100 ET-7000 modules.

For more information about ET-7000 modules, please visit the website: https://www.icpdas.com/en/product/guide+Remote\_I\_O\_Module\_and\_Unit+Ethernet\_I\_O\_Mod ules+ET-7000\_ET-7200

### Configure an ET-7000 Module

First, configure the ET-7000 module by using the web browser. The factory settings of ET-7000 are IP address = 192.168.255.1 and Mask = 255.255.0.0. Make sure that both the module and your PC are on the same network segment (e.g., IP address = 192.168.255.100, Mask = 255.255.0.0).

### **Notice:** The Dip Switch on the rear of the ET-7000 must be set as "Normal".

1. Open the browser and enter the IP address of the ET-7000 to connect it.



2. Enter the user name and password to log into ET-7000.

	連線到 192.168.2	255.1	? 🛛
		E	
Username: Admin Password: Admin (Case sensitive)	及密碼。 警告:此伺服器要 式傳送(基本驗證	的伺服器 192.168.255.1 需要做 求您的使用者名稱及密碼以不 ,不含安全連線)。	
e sensitive)	使用者名稱(U):	🖸 Admin	~
	密碼(2):	****	
		<ul><li>✓記憶我的密碼(R)</li><li>確定</li></ul>	取消

• • http://192.168.255.1/			794	▶ ▶ ▶ ▶ ▶ ■ ▶ 百度一下,你就知道		
我的最愛 🏉 PET-7000 Web page			🕥 🏠 🔹 🗟 👘 🖾 🖶 🔹 網頁 🕑 🗸 安全性 🖏 🗸 🏾			
ICP DAS p://www.icpdas.com			-			
Main Menu	Modbus Setting	is (Al Basic Setting)	2.			
Configuration		Range (40427)	$\mathcal{A}$	Enable (00595)		
Network Settings 1.	Ch0	[05] +/-2.5V	~	OFF ON		
Basic Settings	Ch1	[04] +/-1V	~			
Module I/O Settings	Ch2	[0F] T/C K-type, -270°C ~ 1372°C	×	OFF ON CON Set as "ON" to		
Authentication	Ch3	[0F] T/C K-type, -270°C ~ 1372°C	~	off on enable it.		
Pair Connection	Ch4	[05] +/-2.5V	~	OFF ON C		
More Information	Ch5	[05] +/-2.5V	~	OFF ON .		
	Ch6	[05] +/-2.5V	~	OFF ON .		
	Ch7	[05] +/-2.5V	~	OFF ON .		
	Ch8	[05] +/-2,5V	~	OFF ON •		
	Ch9	[05] +/-2.5V	~			
	Modbus Address	Function		Action		
	00629	50/60 Hz rejection for AI	3.	OFF C ON C 0=50/1=60 Hz (default=1)		
	00631	AI Data Format		OFF ON O= Hex/1=Engineeer (default=0)		
	00632	Recover AI Calibration Values to Facto	ory Setting	1=Clear		
		Sul	bmit	•		

3. Click "Module I/O Settings" under the "Configuration" folder and set the range of channels and then click "Submit". When using the AO module (e.g., ET-7018Z), set "AI Data Format" to "ON" (Engineering).

For example,

The data -10000 to +1000 stands for **-1 to +1 V.** The data -25000 to +25000 stands for **-2.5 to +2.5 V.** The data 258 stands for **25.8 (°C)**.

Type Code	Range	Data Format	Minimum	Maximum
04	-1 to +1 V	Engineering	-10000	+10000
04	-1 10 +1 V	2's comp HEX	8000h	7FFFh
05	-2.5 to +2.5 V	Engineering	-25000	+25000
05	-2.5 to +2.5 v	2's comp HEX	8000h	7FFFh
	Type M	Engineering	-20000	+10000
18	Thermocouple - <b>200 to 100°C</b>	2's comp HEX	8000h	4000h

### **Restore and Modify the Win-GRAF Project:**

Win-GRAF <u>demo programs</u> as described below can be downloaded from the <u>website</u>.

To restore the project to Win-GRAF workbench, click the "File - Add Existing Project - From Zip". (Refer to Section 11.4 if necessary).

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

12] 🛃 🛃 🛃 🐁 🖻 🖻 🗙 🖘 🏷 🔊 🛯 🟭 🏭 💁 💁 🗳 🖬 🚱 🚹 🖷 🚱 🕼 🖀 🖓								
🔁 🖬 🖻 🛃 🐰 🖻 🖻 🗙 🛒		1						
Workspace	IO Drivers *							
💷 🚺 demo_et7060 🔵	E - Ma MODBUS Master	Name Value 🝸 Name	Туре					
Exception programs	品 · · · 品 · · 品 · Open MODBUS: 10.0.255.2:502 ►	Mode Open ^ ET7060_DI_00	BOOL ^					
🖻 🛄 Programs	** <b>*</b> <2> Read Input Bits (1) [16]	Addre 10.0.2 ET7060_DI_01	BOOL					
🛄 Main	📲 🗄 👘 📲 <15> Write Coil Bits (1) [16]	Port 502 ET7060_DI_02	BOOL					
🖻 🛄 Watch (for debugging)	MoDBUS Slave	Recon ET7060_DI_03	BOOL					
Soft Scope		Slaves ET7060_DI_04	BOOL					
🔤 Initial values	Request Slave/Unit Address Nb Item Ad	ctivation Period (m: ET7060_DI_05	BOOL					
Binding Configuration	<2> Read Input Bits 1 1 6 Pe	riodic 50 ET7060 status	DINT Y					
🚽 🕯 🛛 Global defines	216	n Change 0	>					
🖓 Variables		Name Value						
🗖 Types	ID Drivers, Main							

 Double-click "Open Modbus: IP:502" to open the "MODBUS Master Port" window. In this case, a Modbus TCP Master is enabled to connect to an ET-7000. Enter the IP address of the ET-7000 and the "Port" is fixed "502", also set the "Protocol" to "TCP - Open Modbus".

MODBUS Master	r Port	×	
MODBUS on     Address:	Ethernet	OK Cancel	
Port: Protocol:	502 TCP - Open MODBUS UDP - MODBUS RTU UDP - Open MODBUS	iter the current the ET-7000.	IP address
O Serial MODB Com. port: Delay between r			
Delay (ms):	5		
Manage diagno	ect after communication error ostic info for slaves ot open and manage this por		

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

ET-7060 is an Ethernet I/O module with 6 DI and 6 Relay Output channels. The Win-GRAF project describes in this section is "demo\_ET7060.zip". Before using the module, make sure you have completed the module and port settings noted in <u>Section 5.2.1</u>.

### **Description of the Demo:**

To connect an ET-7060 via LAN3 of the RPAC. There are two data blocks, one is used to read 6 DI data, the other is used to write 6 DO data.

1. Click (or double-click) the first Data Block (i.e., <2> Read Input Bits) to view the settings.

10 D	Drivers *								ΗZ	×
	E MoDBUS Master	r		Name		Value	$\mathbf{T}$	Name	Туре	
暴	🖻 🚠 Open MODBl	JS: 10.0.255.2:502		Reque	est	<2 ^		ET7060_DI_00	BOOL	^
*	⊞— <b>*目</b> <2> Read	Input Bits (1) [16		Slave,	/Unit	1		ET7060_DI_01	BOOL	
	🗄 🗝 📲 <15> Writ	e Coil Bits (1) [16		Addre	ss	1		ET7060_DI_02	BOOL	
8	MoDBUS Slave			Nb Ite	em	6 🗸		ET7060_DI_03	BOOL	
	Cumbal	0	Offset	ll a a	1.01			ET7060_DI_04	BOOL	
_	Symbol	Operation		Mask	Storag			ET7060_DI_05	BOOL	
	ET7060_DI_00	Data exchange	0	FFFF	Defau			ET7060_status	DINT	
¢,5	ET7060_DI_01	Data exchange	1	FFFF	Defau	ılt ┥		ET7060 RL 00	BOOL	
- 214	ET7060_DI_02	Data exchange	2	FFFF	Defau	ult		ET7060 RL 01	BOOL	
₽Ļ	ET7060_DI_03	Data exchange	3	FFFF	Defau	ılt		ET7060 RL 02	BOOL	
	ET7060_DI_04	Data exchange	4	FFFF	FFFF Default			ET7060 RL 03	BOOL	
÷	ET7060_DI_05	Data exchange	5	FFFF	Defau	ult		ET7060 RL 04	BOOL	
	ET7060_status	Error report	0	FFFF	Defau	ılt		<	DOOL	
	<					>				_
< >	IO Drivers Main									

### Description:

To read **6** DI data from the device (Slave ID = 1) by sending the request every **50 ms**. It will be sent after **5** seconds if an exception occurs. No response for **500 ms** can be regarded as abnormal.

DBUS Master	Request			
Request Description:				ОК
<u>Slave/Unit:</u> MODBUS Reques	1			of a Modbus TCP T-7000) is fixed "1".
<1> Read Coil <2> Read Inp <3> Read Hole <4> Read Inp	ut Bits <del>Jing Regist</del>		)	
Data block				
Base <u>a</u> ddress:	1			
<u>N</u> b items:	6		]	
Activation				
• Periodic:	50	ms	5000	
◯ On caḷ ◯ On change	9 <del>0</del>		(on error)	
Misc.	22			Refer to Section 5.1 – <u>Step 4</u> fo
<u>T</u> imeout:	500	ms	]	detals about the data block.
Nb trials:	1			

2. Click (or double-click) the second Data Block (<15> Write Coil Bits) to view the settings.

10 D	Drivers *							ΗZ	×
冒	E MoDBUS Master			Name	Value	Y	Name	Туре	
뮮	🖻 🚠 Open MODBl	JS: 10.0.255.2:502	2	Req	<15> Wr ^		ET7060_DI_01	BOOL	^
**	⊞* <b>≣</b> <2> Read	Input Bits (1) [16	5]	Sla	1		ET7060_DI_02	BOOL	
	⊞* <b>⊟</b> <15> Write	e Coil Bits (1) [1	6] 🦕 👘	Add	1		ET7060_DI_03	BOOL	
8	MoDBUS Slave		$\sim$	Nb I	6		ET7060_DI_04	BOOL	
				يت م ال	On Cha		ET7060_DI_05	BOOL	
	Symbol	Operation	Offset	Mask	Storage		ET7060 status	DINT	
	ET7060_RL_00	Data exchange	0	FFFF	Default		ET7060_RL_00	BOOL	
¢,	ET7060_RL_01	Data exchange	1	FFFF	Default		ET7060 RL 01	BOOL	
	ET7060_RL_02	Data exchange	2	FFFF	Default		ET7060 RL 02	BOOL	
∎+	ET7060_RL_03	Data exchange	3	FFFF	Default	9	ET7060 RL 03	BOOL	
÷	ET7060_RL_04	Data exchange	4	FFFF	Default 🔫		ET7060 RL 04	BOOL	
	ET7060_RL_05	Data exchange	5	FFFF	Default		ET7060_RL_05	BOOL	~
			$\square$	,			<	>	
	<				>	NI-	amo Utalua		
< >	IO Drivers Main								

## Description:

To write **6** DO data to the device (Slave ID = 1) by sending the request when the data is changed. No response for **500 ms** can be regarded as abnormal.

MODBUS Master	Request		<b>X</b>	
Request Description:			ОК	
<u>S</u> lave/Unit:	1		f a Modbus TCP -7000) is fixed "1"	
ADDBUS Reques	ile coil bit <del>Ile holding regis</del>	<sup>stor</sup>		
Base <u>a</u> ddress:	6			
Activation O Periodic:		ms 5000 (on error)		
<ul> <li>On change</li> <li>Misc.</li> <li>Timeout:</li> </ul>	500	ms	Refer to Section	
Nb trials:	1		detals about the	data block.

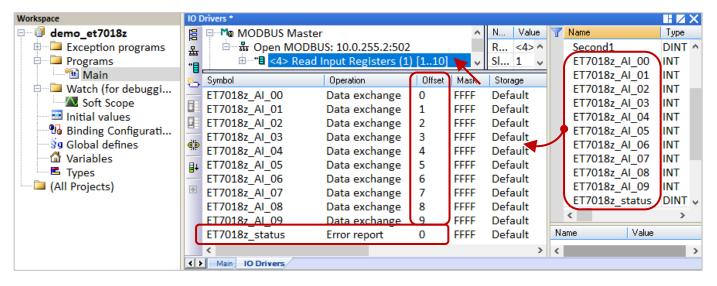
## 5.2.3 Connecting the ET-7018Z (10 AI)

The ET-7018Z is an Ethernet I/O module with 10 AI channels. The Win-GRAF project describes in this section is "demo\_ ET7018z.zip". Before using it, make sure you have completed the module and port settings noted in <u>Section 5.2.1</u>.

### **Description of the Demo:**

To connect an ET-7018Z via LAN3 of the RPAC. There is one data block used to read 10 AI data.

1. Click (or double-click) the data block (i.e., <4> Read Input Registers) to view the settings.



## Description:

To read **6** DI data from the device (Slave ID = 1) by sending the request every **50 ms**. It will be sent after **5** seconds if an exception occurs. No response for **500 ms** can be regarded as abnormal.

DBUS Master	Request					
Request Description:				ОК		
<u>S</u> lave/Unit: MODBUS Reques	1			f a Modbus T T-7000) is fixe		
<2> Read Inp <3> Read Hole <4> Read Inp <5 White	ut Bits <del>Jing Regist</del>		)	< >		-
Data block						
Base <u>a</u> ddress:	1					
<u>N</u> b items:	10		)			
Activation						
• Periodic:	50	ms	5000			
◯ On call ◯ On change	1.		(on error)			
Misc.	1004			Refer to Se	ction 5.1	– Step 4 fo
Timeout:	500	ms	J	detals abo		
Nb trials:	1		-			

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

The Modbus TCP/UDP Master port enabled in the Win-GRAF "Fieldbus Configuration" - "IO Drivers" window will be activated automatically whenever the PAC is powered on.

If you want to disable the Modbus **TCP/UDP** Master port while the program is running, using the "**MBTCP\_M\_disable**" or "**MBUDP\_M\_disable**" function as follows.

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

When "To\_disable" is set to "TRUE", the Modbus TCP Master port that connects to a TCP Slave (IP address = "10.0.255.2" and Port\_No = 502) will be disabled. If it is necessary, using the "**MBTCP\_M\_enable**" (or "**MBUDP\_M\_enable**) function to enable it as follows.

(\* Declare To\_enable as BOOL
 Status\_tcp as BOOL \*)
If To\_enable then
 To\_enable := FALSE;
 MBTCP\_M\_enable ( '10.0.255.2' , 502 );
End\_if;
Status\_tcp := MBTCP\_M\_status ( '10.0.255.2' , 502 );

The **"MBTCP\_M\_status"** or **"MBUDP\_M\_status"** function can be used to get the status of the Modbus TCP/UDP Master port. (True: enabled ; False: or disabled)

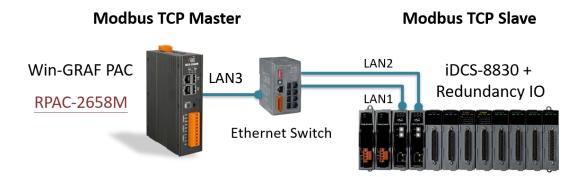
For easy maintenance and error avoidance, declare a STRING variable (e.g., "IP\_addr2", the Length is set to "20") and set its initial value (e.g., "10.0.255.2"). As follows:

```
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 a Modbus TCP Slave Device with two IP addresses

This section describes how to create the redundant "Modbus Master Request", i.e., when one IP address of a Modbus TCP Slave is unavailable, the other IP address still can be used to read/write data.

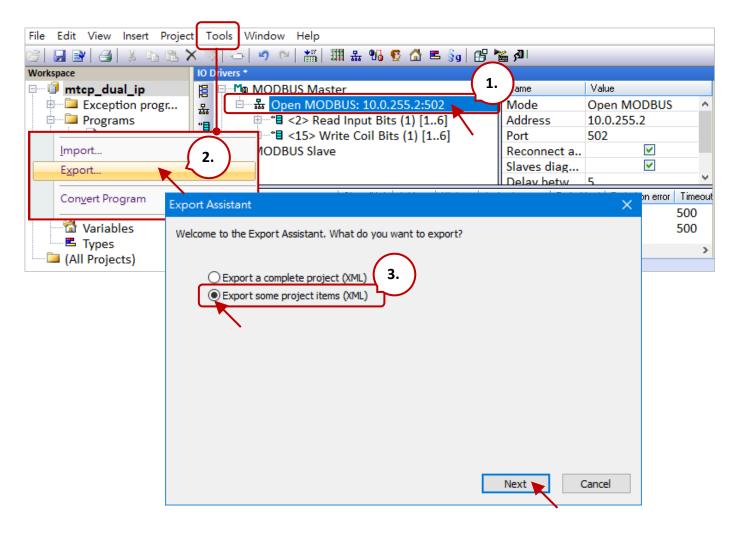
## **Application Diagram:**



### Follow the steps below:

Only one IP address of the Modbus Slave can be specified in the Modbus Master setting. The following describes how to make a copy of the read/write settings and then access data via the second Slave IP address.

- 1. Click on "Open MODBUS:" in the IO Drivers window and click the menu command Tools Export.
- 2. In the "Export Assistant" window, select "Export some project items (XML)" and click "Next".



3. Check only the "MODBUS Master" configuration and click the "Next" button.

Export Assistant	×
Export Assistant         Select items to be exported:         Programs and UDFBs         Main         pOnBadIndex         pOnDivZero         pShutDown         pStartup         Data types         Configurations         MODBUS Master         MODBUS Slave	
Export referenced global variables	:el

4. Specify the file location (default in C:\Win-GRAF\DATA\CONFIG) and enter a file name (e.g., Master\_2.xml), then click the "Save" button. Finally, click "Finish" to export the file.

	🖌 Save As						×
	$\leftrightarrow \rightarrow \cdot \uparrow$	C:\Win-GRAF\DA	TA\CONFIG	ڻ v	, Search CON	IFIG	
	Organize 🔻	New folder				== -	?
	Name	^	Date mod	lified	Туре	Size	^
Export Assistant		Ν	lo items match you	r search.			~
Export is complete.	<						>
	File na	me Master_2.xml					~
	Save as ty	pe: XML files (*.xml)					~
	∧ Hide Folders				Save	Cancel	
				Finish			

5. Open the ".xml" file in Notepad and copy the code from <modbusport> to </modbusport> tags. Then, paste the code above the </modbus> tag and change the address to the second IP address of the Modbus Slave device (e.g., "10.0.255.2"), then save and close the file.

F	Edit Format View Help Copy
	modbus>
	<modbusport address="10.0.255.2" disabled="0" kind="master" port="502" reconnect="1" reqdelay="5" rtu="0" sdiags="1"></modbusport>
	<pre><modbusreq address="0" call="Periodic(50,5000)" mode="Read Input Bits" name="" nbitem="6" retry="1" slave="1" timeout="500"></modbusreq></pre>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="DI_00"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="1" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="DI_01"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="2" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="DI_02"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="3" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="DI_03"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="4" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="DI_04"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="5" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="DI_05"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Status" rangehi="" rangelo="" sighi="" siglo="" symbol="status"></modbusitem>
	<modbusreq <="" address="0" call="OnChange(5000)" mode="Write Holding Bits" name="" nbitem="6" retry="1" slave="1" td="" timeout="500"></modbusreq>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="RL_00"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="RL_01"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="RL_02"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="RL_03"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="RL_04"></modbusitem>
	<modbusitem nbword<="" offset="0" ope="Data" symbol="RL_05" td=""></modbusitem>
	Paste
	<modbusport address="&lt;mark&gt;10.0.255.100&lt;/mark&gt;" disabled="0" kind="master" port="502" reconnect="1" reqdelay="5" rtu="0" sdiags="1"></modbusport>
	<modbusreq address="0" call="Periodic(50,5000)" mode="Read Input Bits" name="" nbitem="6" retry="1" slave="1" timeout="500"></modbusreq>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="DI_00"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="1" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="DI_01"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="2" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="DI_02"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="3" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="DI_03"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="4" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="DI_04"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="5" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="DI_05"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Status" rangehi="" rangelo="" sighi="" siglo="" symbol="status"></modbusitem>
	<modbusreq <="" address="0" call="OnChange(5000)" mode="Write Holding Bits" name="" nbitem="6" retry="1" slave="1" td="" timeout="500"></modbusreq>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="RL_00"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="RL_01"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="RL_02"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="RL_03"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="RL_04"></modbusitem>
	<modbusitem mask="ffff" nbword="1" offset="0" ope="Data" rangehi="" rangelo="" sighi="" siglo="" symbol="RL_05"></modbusitem>

6. Click the menu command **Tools - Import i**n the Win-GRAF Workbench.

File Edit View In	sert Project	Tools Window	Help										
2 🔄 🛃 😢 🛃 👗 🔁 📉 🔁 🔍 🔍 🔚 🎬 🏭 🐇 % 🧟 🖓 🖷 🗞 🕼 🖉 🖓													
Workspace		IO Drivers *											
🖃 🗐 mtcp_d	-								Name		Value		
🗄 🖾 Exce	Import				10.0.2	255.2:50	)2		Mode		Open MOI	DBUS	~
🖻 🗁 Prog	Evport			_	ut Bits	; (1) [1	.6]		Address		10.0.255.2		
····· 🕂 N	E <u>x</u> port				oil Bit	s (1) [1.	6]		Port		502		
🖻 🗀 Wat	C 10								Reconn	ect a		✓	
📉 s	Convert Pr	ogram		'					Slaves o	liag		✓	
🔤 Initiar va	ues								Delay h	etw	5		×
📲 😼 Binding (		Request		Slav	e/Unit	Address	Nb Item	Acti	vation	Period	(ms) Period o	n error	Timeout
🚽 🖥 🖁 Global de		🏟 <2> Read Inp	out Bits	1		1	6	Peri	odic	50	5000		500
🖄 Variables		<15> Write C	Coil Bits	1		1	6	On	Change	0	5000		500
Types		<u> </u>											>
🖾 🗀 (All Projects			) Drivers										

7. Click "Import project items (XML)" and the "Next" button in the Import Assistant window.

Import Assistant	×
Welcome to the Import Assistant. What do you want to import? Inport project items (XML) Import ST file Import project items from previous versions	
<ul> <li>Import common definitions</li> <li>Import library definitions</li> <li>Import OEM libraries</li> <li>Import Embedded HMI hardware definition</li> </ul>	
Next Cancel	

8. Select the file you want to import (e.g., "Master\_2.xml") and click the "Open" button, and then click "Finish" to import the file.

	🖌 Open				×
Import Assistant	$\leftrightarrow \rightarrow \cdot \uparrow$	🔜 « DATA → CONFIG	ٽ ~	, ○ Search CONFIG	
Import is complete.	Organize 🔻	New folder			□ ?
	Name	^ <u>2.</u>	Date modified	Туре	Size
	Master_2.xm		2021/3/3 下午 03:05	XML Document	5 KB
		N			
	<				>
		File name: Master_2.xml	~	XML files (*.xml)	~
				Open	Cancel
	<u></u>		<b>_</b>		
		( 3	s. )		
			Finish		

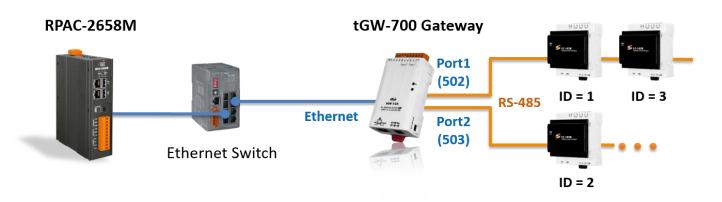
9. By now, it allows accessing data from a Modbus Slave with its second IP address.

IO D	Drivers *
臣	🗏 Ma MODBUS Master
쁆	🖮 🚠 Open MODBUS: 10.0.255.2:502
	⊞*• <2> Read Input Bits (1) [16]
* <b>1</b>	
-0-	( 🖻 👪 Open MODBUS: 10.0.255.100:502
	⊞*• <15> Write Coil Bits (1) [16]
	M& MODBUS Slave

# 5.4 Accessing multiple Modbus RTU Slave via the tGW-700 Gateway

When using Modbus RTU (RS-232/485/422) devices for long-distance transmission, a lower baud rate is set, which allows for stable and longer transmission distance but slower transmission speed. In this case, you can use the tGW-700 series (Modbus TCP to RTU/ASCII gateway) to send data over Ethernet to improve communication efficiency.

This chapter describes how a Win-GRAF PAC communicates with several LC-103 modules via the tGW-700 gateway (see figure below).



The <u>demo program</u> (demo\_tgw725.zip) can be download on the website:

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

**The tGW-700** is a Modbus TCP to RTU/ASCII gateway that enables a Modbus/TCP host 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 <u>tGW-700 series webpage</u> for more information about the product, and refer to Chapter 3 & Chapter 4 of the <u>tGW-700 series User Manual</u> for instructions on tGW-700's network setting, testing methods, and web configuration.

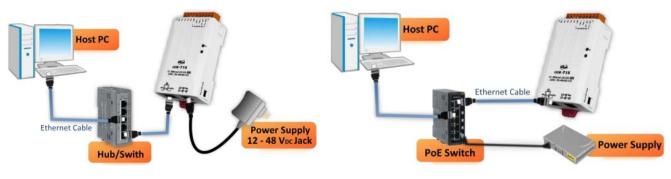
Before using the tGW-700, configuring the network and COM Port setting:

## • 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 PC on the same sub-network, and then power on the tGW-700.



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

**PoE Power Supply** 

• Install "eSearch Utility" on PC and then Search and Configure the Network Setting for the tGW-700.

https://www.icpdas.com/en/product/guide+Software+Utility\_Driver+eSearch\_\_Utility

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		(2)			>
tus	ers Confi	guration (UDP)	😂 Web	Exit	

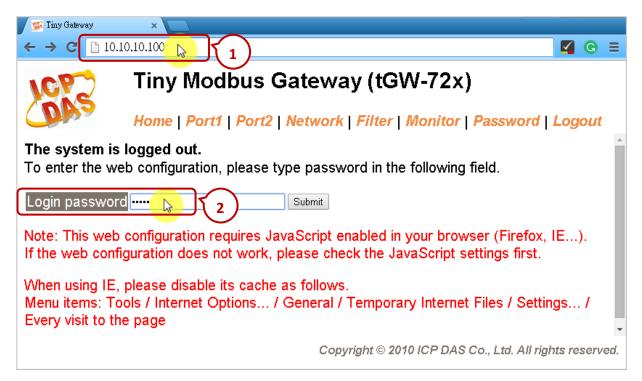
The user can modify the IP Address, Subnet Mask, and Gateway settings. After that, click the "OK" button. New settings will take effect within 2 seconds.

Configure Server (UDP)									
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!									

## Web Configuration

The following describes the COM Port setting:

- 1. Enter the IP address of tGW-700 into the address bar. Note that both the tGW-700 and a PC must on the same sub-network.
- 2. Enter the password (the factory setting is "admin") and click Submit to log in.



3. After that, the current port setting will be displayed on the Home page. Also, change the settings on the "Port1" or "Port2" page.

Tiny Mod	bus Gateway (tGW-72x)	<b>^</b>				
Home   Port1	Port2   Network   Filter   Monitor   Pass	or   Password   Logout				
Initial	Switch OFF	System Timeout (Network Watchdog, Seconds)				
Current port settings:						
Port Settings	Port 1	Port 2				
Baud Rate	e (bps) 9600	9600				
Data Size	e (bits) 8	8				
	Parity None	None				
Stop Bits	s (bits) 1	1				
Modbus Pr	rotocol RTU	RTU				
Slave Timeou	ut (ms) 300	300				
Char Timeout (	(bytes) 4	4				
Silent Tim	e (ms) 0	0				
Read Cach	e (ms) 980	980				
Local TC	P Port 502	503				
TCP Timeout (Sec	conds) 180	180				
Pair-Connection Settings (Master/Slave Mode)	Port 1	Port 2				
Server	Mode Server	Server				
Remote Se	rver IP -	-				
Remote TC	P Port -	-				
RTU Vir	tual ID -	-				
TCP SI	ave ID -	-				

## 5.4.2 Accessing the LC series Module via the tGW-700 Gateway

The following describes how Win-GRAF PAC access data from LC-103 modules via the tGW-725 gateway.

tGW-725 is a tiny Modbus TCP to RTU/ASCII gateway with PoE and 2-port RS-485.

- **LC-103H** is a lighting control module that supports the Modbus RTU protocol and provides 1 channel for digital input and 3 channels for relay output. Before using it, set the ID number of the module,
  - e.g., adjust the rotary switch to "1" at the bottom of the module to set the ID as "1".

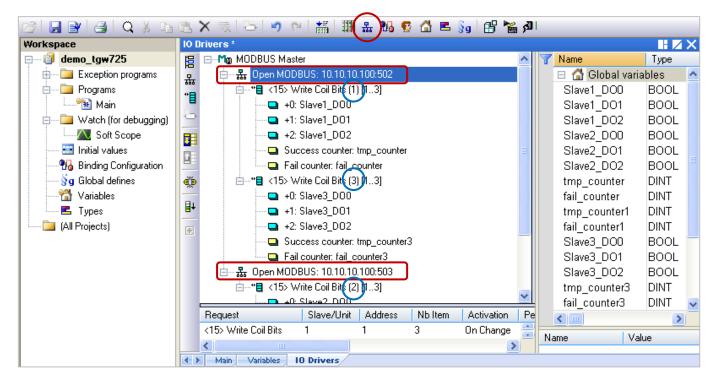


The LC series module user manual: http://www.icpdas.com/en/download/s how.php?num=596&model=LC-103H/

Download the <u>demo program</u> (demo\_tgw725.zip) on the website. Click the menu commands File - Add Existing Project - From Zip... in the Workbench to open the project.

## Description of the Demo: (Refer Section 5.2 to set a Modbus TCP Master)

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



## The table lists the information about each piece of equipment:

Win-GRAF PAC		tGW-725	LC	-103H	
	Modbus <sup>-</sup>	TCP to RTU/ASCII Ga	Modbus RTU Slave		
Modbus TCP Master	IP Address	TCP Port = 502	COM1	Two modules	Slave ID = 1 &3
	10.10.10.100	TCP Port = 503	COM2	One module	Slave ID = 2

 Click (or double-click) the first data block under port 502 to view the settings. The "Operation" is set to "Success counter" (or "Fail counter"), which means the value of the variable will be added 1 if succeed (or failed) to send the command. Also, the "Offset" value of these items must be set as "0".

10 0	)rivers *									Х
間	🖃 Modbus	Master				~	T	Name	Туре	
쁆	🖨 🚠 Open I	MODBUS: 10.10.10.100:50	2					😑 🚮 Global vari	ables	^
*		5> Write Coil Bits (1) [13]						Slave1_DO0	BOOL	
		+0: Slave1_D00				_		Slave1_DO1	BOOL	
8	🗖	+1: Slave1_D01	Ň			=		Slave1_DO2	BOOL	
	🗖	+2: Slave1_D02						Slave2_DO0	BOOL	
		Success counter: tmp_co	unter					Slave2_DO1	BOOL	
	<b>D</b>	Fail counter: fail_counter				_		Slave2_DO2	BOOL	
₿,	ia*∎ <1	5> Write Coil Bits (3) [13]						tmp_counter	DINT	
	····· 🗅	+0: Slave3_D00						fail_counter	DINT	
∎+	🗅	+1: Slave3_D01						tmp_counter1	DINT	
	🗖	+2: Slave3_D02						fail_counter1	DINT	
		<u>. Success counter tran, co</u>	unter3			<u> </u>		Slave3_DO0	BOOL	_
	Symbol	Operation	Offset	Mask	Storage	Rang		Slave3_DO1	BOOL	
	Slave1_D00	Data exchange	0	FFFF	Default	^		Slave3_DO2	BOOL	
	Slave1_D01	Data exchange	1	FFFF	Default			tmp_counter3	DINT	
	Slave1_D02	Data exchange	2	FFFF	Default			fail_counter3	DINT	~
	tmp_counter	Success counter	0	FFFF	Default			<	>	
	fail_counter	Fail counter	0	FFFF	Default	~	N	ame Va	lue	
	<					>				
< >	Main Variables	s 10 Drivers								

## Description:

To write **three** DO data to the device (Slave ID = 1) by sending the request when the program calls. No response for **3 seconds** can be regarded as abnormal.

MODBUS Master Request	2	<
Request	ОК	
Slave/Unit: 1	The Slave ID of the device (LC-103H) is "1".	
<5> Write single coil bit <6> Write single holding register <15> Write Coil Bits		
Base address:     1       Nb items:     3		on 5.1 – <u>Step 4</u> for bout the data block.
Activation Periodic: 0 ms On call On change	0 (on error)	
Misc. <u>Timeout:</u> 3000       Mb trials:     1	Recommender 1000 to 3000	

3. Click (or double-click) the second data block under port 502 to view the settings.

The "Operation" is set to "Success counter" (or "Fail counter"), which means the value of the variable will be added 1 if succeed (or failed) to send the command. Also, the "Offset" value of these items must be set as "0".

IO D	rivers *									×
臣	⊨*∎ <	15> Write Coil Bits (1) [13]				^	7	Name	Туре	
쁆	<b>C</b>	+0: Slave1_D00						😑 🚮 Global varia	ables	^
		+1: Slave1_D01						Slave1_DO0	BOOL	
*8	<b>C</b>	+2: Slave1_D02						Slave1 DO1	BOOL	
ъ		Success counter: tmp_co	unter					Slave1 DO2	BOOL	
		Fail counter: fail_counter				=		Slave2 DO0	BOOL	
	i i*∎ <	15> Write Coil Bits (3) [13]						Slave2_DO1	BOOL	
		+0: Siave3_000	<u> </u>					Slave2_DO2	BOOL	
ġ,þ	<b>C</b>	+1: Slave3_D01						tmp_counter	DINT	
		□ +2: Slave3_D02 fail_counter DINT								
∎+		Success counter: tmp_co	unter3					tmp_counter1	DINT	
÷		Fail counter: fail_counter3						fail counter1	DINT	
		MODBUS: 10 10 10 100:50	3			~		Slave3_DO0	BOOL	_
	Symbol	Operation	Offset	Mask	Storage	Rang		Slave3 DO1	BOOL	
	Slave3_D00	Data exchange	0	FFFF	Default	~		Slave3_DO2	BOOL	
	Slave3_D01	Data exchange	1	FFFF	Default			tmp_counter3	DINT	
	Slave3_D02	Data exchange	2	FFFF	Default			fail_counter3	DINT	~
	tmp_counter3	Success counter	0	FFFF	Default			<	>	
	fail_counter3	Fail counter	0	FFFF	Default	~		ame Val		-
	<		_			>			uc	_
< >	Main Variable	s 10 Drivers								

### Description:

To write **three** DO data from the device (Slave ID = 3) by sending the request when the program calls. No response for **3 seconds** can be regarded as abnormal.

MODBUS Master Request	
Request	ОК
Slave/Unit: 3 MODBUS Request	The Slave ID of the device (LC-103H) is "3".
<5> Write single coil bit <6> Write single holding r <15> Write Coil Bits	register
Data block Base <u>a</u> ddress: 1	 
<u>N</u> b items: 3	
Activation O Periodic: O On call	ms 0 (on error)
• On change	
Misc. Timeout: 3000	ms
Nb trials: 1	

4. Click (or double-click) the first data block under port 503 to view the settings. The "Operation" is set to "Success counter" (or "Fail counter"), which means the value of the variable will be added 1 if succeed (or failed) to send the command. Also, the "Offset" value of these items must be set as "0".

IO D	rivers *									Х
眉	🗖	+1: Slave1_D01				^	7	Name	Туре	
恭	🗖	+2: Slave1_DO2						🗉 🚮 Global vari	ables	^
*		Success counter: tmp_co	unter					Slave1_DO0	BOOL	
510	····· 🔤	Fail counter: fail_counter						Slave1_DO1	BOOL	
0	⊡*∎ <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	
¢,þ	🖸	Success counter: tmp_co	unter3					tmp_counter	DINT	
	🖸	Fail counter: fail_counter3	}					fail_counter	DINT	
∎+	🖻 🚠 Open I	MODBUS: 10.10.10.100:50	)3					tmp_counter1	DINT	
		5> Write Coil Bits (2) [13]						fail_counter1	DINT	
			Co		Loi			Slave3_DO0	BOOL	_
	Symbol	Operation	Offset	-	Storage	Rang		Slave3_DO1	BOOL	
	Slave2_D00	Data exchange	0	FFFF	Default			Slave3_DO2	BOOL	
	Slave2_D01	Data exchange	1	FFFF	Default			tmp_counter3	DINT	
	Slave2_D02	Data exchange	2	FFFF	Default			fail_counter3	DINT	~
	tmp_counter1	Success counter	0	FFFF	Default Default			<	>	
	fail_counter1	Fail counter	0	FFFF	Default	_	N	ame Va	lue	
	<					>				_
<b>*</b> >	Main Variables	s 10 Drivers								

## Description:

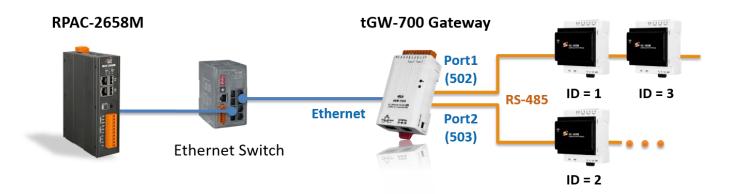
To write **three** DO data to the device (Slave ID = 2) by sending the request when the program calls. No response for **3 seconds** can be regarded as abnormal.

Request       OK         Description:       Image: Constraint of the device (LC-103H) is "2".         MODBUS Request       Image: Constraint of the device (LC-103H) is "2".	
MODBUS Request	
<5> Write single coil bit <6> Write single holding register <15> Write Coil Bits	
Data block         Base address:       1         Nb items:       3    Refer to Section 5.1 – <u>Step 4</u> for more details about the data block	
Activation         O Periodic:       0         On call       (on error)         O on change       0	
Misc.     Recommended to be set as 1000 to 3000 ms.	]

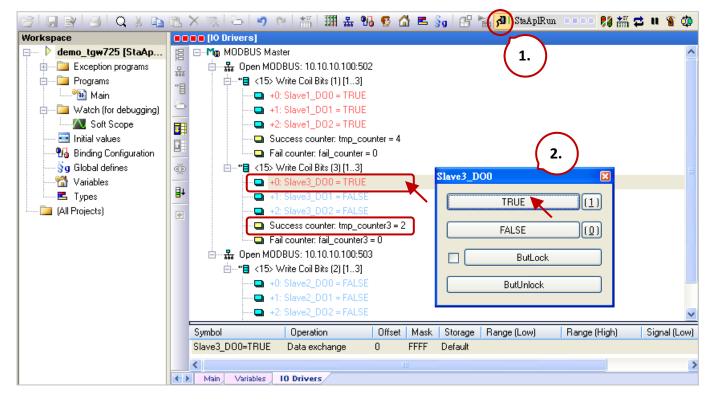
## 5.4.3 Test the Project (demo\_tgw725.zip)

Please download this demo program to your Win-GRAF PAC before testing.

### The Hardware Wiring:



Click the "On Line" button to connect with the PAC. In the "IO Drivers" window, double-click any DO variable to set it as "TRUE". If successes to write data, the "tmp\_counter" value will be added "1".



<u>Note</u>: While the PAC is turned on, it will send a Modbus request to the Modbus slave device. So, you can see the value of "tmp\_counter" is "1" at the beginning, which means succeed to write data.

# Chapter 6 Retain Variable and Data Storage

Retain variables are used to protect important data. The system can retain the latest variable data during an unexpected power failure. The user can either use RETAIN variables in the Variables window or use the RETAIN\_xxx functions.

#### Note:

- To avoid internal data conflict, do not use both RETAIN variables and functions in the same project. Otherwise, functions will invalid.
- Using a great amount of RETAIN variables will cause the lower performance of the system due to access data to a disk frequently.

#### **Add RETAIN variables**

In the **Variables** window, right-click on **RETAIN variables** and click **Add Variable** to add a RETAIN variable. Next, set the name and type for the variable and press Enter.

67 🖬 🖻 🤮 🕹 🕹 🖼 🛪 🦷	<li></li>	, % 🔨 (	3 E	§g	🗄 層 🖓	
Workspace	Variables					
🖃 📲 📴 Retain_variable	🝸 Name	Туре			Dim. Attrib.	Syb. Init valu
🗄 🔤 Exception programs	Global variables					
🖻 🗝 Programs	🗆 🚽 RETAIN variables					
🛄 Main	Retain_BOOL	BOOL	9	<u>U</u> ndo		
🚊 🛁 Watch (for debugging)	Retain_DINT	DINT	6	<u>R</u> edo		
Soft Scope	Retain_REAL	REAL	¥			
🔜 Initial values	Retain INT INT			<u>C</u> ut		
📶 📶 Binding Configuration	Retain I REAL I REAL			Сору		
jg Global defines	NewVar BOOL			Paste		
Variables	Main			Clear		
	POnBadIndex			_		
(All Projects)	POnDivZero		C)	<u>E</u> dit		
	pShutDown			Cance	el Sorting	
	pStartup					
				E <u>n</u> abl	e Changes	Space
				<u>S</u> wap	Global <> Reta	in
			-	A <u>d</u> d V	/ariable	Ins
				Add <u>N</u>	<u>A</u> ulti Variables	

## **Test RETAIN variables**

Download the project to Win-GRAF PAC and change the value online. If the setting is correct, the modified value will be displayed after a reboot.

27 🖬 🗗 (8) 🕹 🖬 🖪 🗙 🗉	※  谷  ら ら  器  翻:	# 😘 😨 🔂 🛋 §g	🕄 🎽 🚮 RUN	- n 🛱 🐘 💶	🐒 🥸
Workspace	[Variables]				
🖃 🕨 Retain_variable [RUN]	T Name	Value	Туре	Dim. Attrib.	Syb.
🗄 🚞 Exception programs	🚮 Global variables				
🚊 🚞 Programs	🗉 🛃 RETAIN variables				
🔛 🎦 Main	Retain BOOL	TRUE	BOOL		
📄 🚞 Watch (for debugging)	Retain DINT	2147483647	DINT		
Soft Scope	Retain REAL	6563.249023	REAL		
🔜 Initial values	Retain INT	32767	INT		
	Retain LREAL	36597.246876	LREAL		
	Retain_STRING	'abcd12345'	STRING(255)		
	Main				

# 6.1 Using the RETAIN\_xxx Function

This chapter lists the way to use the "RETAIN\_ARY", "RETAIN\_FLAG\_CLR", "RETAIN\_FLAG\_GET", "RETAIN\_FLAG\_SET"and "RETAIN\_VAR" Functions. The Win-GRAF PAC has a built-in storage memory for users to save retain variable data that will not be lost when the PAC is shutdown. The data still exists after the PAC is turned on.

## Note:

- 1. To avoid internal data conflict, do not use both RETAIN variables and functions in the same project. Otherwise, functions will invalid.
- 2. 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. When a function calls in another Cycle, it will return "FALSE".
- 3. The user must specify an initial value for the retain variable. If not, an invalid value will be received after executing the function in the program on the PAC.

Download <u>the demo program</u> (**demo\_retain.zip**) on the website. Click the menu commands **File - Add Existing Project - From Zip...** in the Workbench to open the project.

## ST Program:

The demo describes Function Retain\_Var() and Retain\_Ary().

```
(* "on line change cycle" is declared as DINT, a non-zero value indicates that on-line-change
  is being executed in that Cycle. "retain done" is declared as BOOL with an initial value
  "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 or set variables in the "Variables" window.

Name /	Туре	Dim.	Attrib.	Syb.	Init value	User	Tag	De
🗉 🚮 Global variables								^
BOOL_1	BOOL							
BOOL_2	BOOL							
DINT_1	DINT							
DINT 2	DINT							
DINT_3	DINT							
INT_ARY	INT	[09]						
LINT_1	LINT							
LREAL_1	LREAL							
on_line_change_cycle	DINT							
REAL_1	REAL							
REAL_ARY	REAL	[0 101	17					
retain_done	BOOL	20	\$					
tmp_bool	BOOL	-						Y
<							3	

## 6.1.1 RETAIN\_VAR: Retain a Variable



## 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.

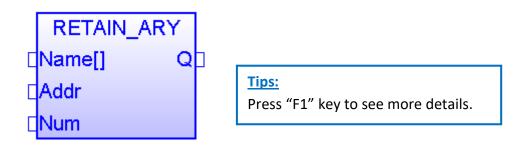
## Q:

Data Type: BOOL. TRUE: Ok; FALSE: Error.

## Note:

- 1. One variable (or one element of the array) can be stored at only one address. Do Not assign the same address to two or more variables, or the variable data will be wrong.
- 2. When using the 64-bit data type (e.g., LINT or LREAL), set the address to 10001 to 12000.

## 6.1.2 RETAIN\_ARY: Retain an Array Variable



## Name[]:

An ARRAY variable name (DO NOT use String). The data 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 number of elements in the array variable to be retained.

For example:

If an array has 10 elements, setting the Num as 1 to 10 is correct. But, set it larger than 100 is wrong. If an array has 5 elements, setting the Num as 1 to 5 is correct. But, set it larger than 5 is wrong.

## Q:

Data Type: BOOL. TRUE: Ok; FALSE: Error.

#### Note:

- 1. One variable (or one element of the array) can be stored at only one address. Do Not assign the same address to two or more variables, or the variable data will be wrong.
- 2. When using the 64-bit data type (e.g., LINT or LREAL), set the address to 10001 to 12000.

## 6.1.3 RETAIN\_FLAG\_SET/GET/CLR (Set/Get/Clear the Retain Flag)

## How to Use:

"**Retain Flag**" is a flag (TRUE/FALSE) that is used to check if the retained data is a valid value and is stored in the memory. For the program to function properly, the user must specify an initial value for all retain variables. If not, the data read from the memory after turning on the PAC is normally a random value.

If "Retain\_Flag\_Set()" is set to TRUE which means all retain data have been specified an initial value. "Retain\_Flag\_Get()" is used to get the state of the flag and "Retain\_Flag\_Clr()" is used to clear the state of the flag.

## ST Program:

(\* "on\_line\_change\_cycle" is declared as DINT, a non-zero value indicates that on-line-change is being executed in that Cycle.

"retain\_done" is declared as BOOL with an initial value "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 ;

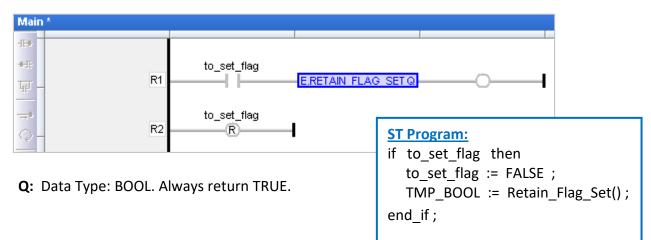
(\* When all Retain variables are assigned proper values, remember to set the "to\_set\_flag" to "TRUE" for calling "Retain\_Flag\_Set() once, so that, when next time you use the "Retain\_Flag\_Get()", it can return "TRUE". \*)

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

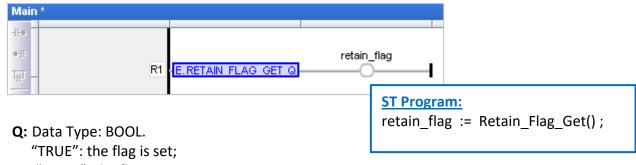
## LD Program:

(Press the "F1" key to view the description of the setting.)

**RETAIN\_FLAG\_SET:** Set the retain flag.

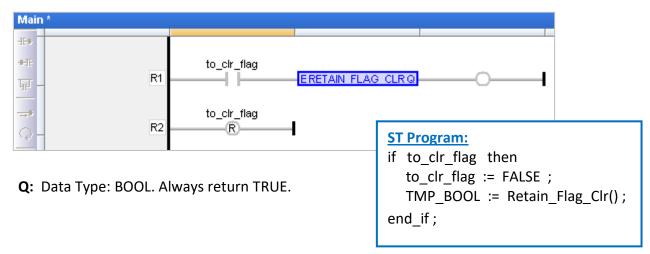


## **RETAIN\_FLAG\_GET:** Get the state of the retain flag.



"FALSE": the flag is not set.

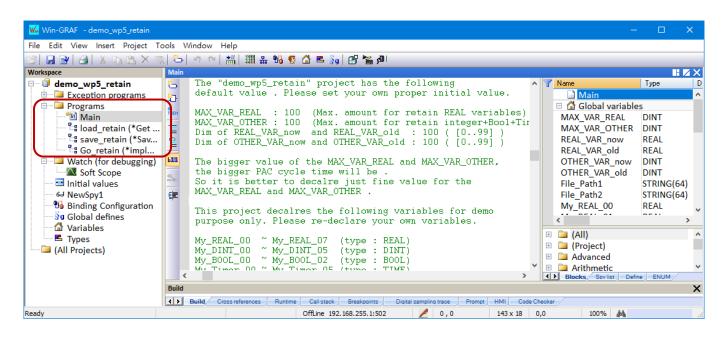
RETAIN\_FLAG\_CLR: Clear the retain flag.



# 6.2 Save Retain Variables and Data to a File

Download <u>the demo program</u> (**demo\_wp5\_retain.zip**) on the website and then run the File menu commands "Add Existing Project" and "From Zip..." in the Win-GRAF Workbench to open the project.

This project includes an ST main program (i.e., Main) and three ST sub-programs (i.e., load\_retain, save\_retain, and Go\_retain).



## Limitation :

This project is not applicable for retaining variable data that changes frequently, e.g., the data will be changed every second or every minute. In this case, the retained data is stored in a file under the /System\_Disk path. If the data changes frequently, updating the file will consume lots of CPU time, also reduce the performance of the PAC.

The table lists the setting value used in the "demo\_wp5\_retain". Also, change to a proper value to meet specific application requirements.

Variable Name	Value	Description
MAX_VAR_REAL	lnit. = 100	The maximum amount of REAL retains variables can be used.
MAX_VAR_OTHER	Init. = 100	The total maximum amount of Integer, BOOL, and TIMER retain variables can be used.
REAL_VAR_now REAL_VAR_old	Dim. = 100	[099], which must be the same as the "MAX_VAR_REAL" value.
OTHER_VAR_now OTHER_VAR_old	Dim. = 100	[099], which must be the same as the "MAX_VAR_OTHER" value.

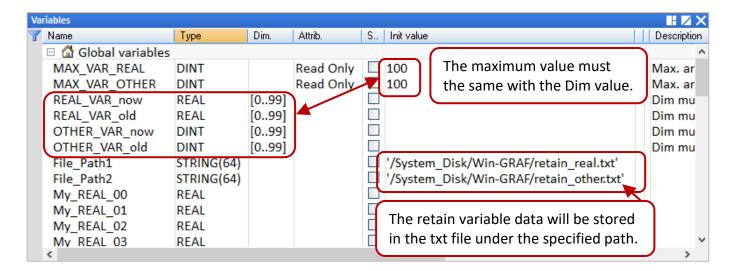
## Note:

The bigger the value of the MAX\_VAR\_REAL and MAX\_VAR\_OTHER, the larger the PAC cycle time will be. So, specify the proper amount of MAX\_VAR\_REAL and MAX\_VAR\_OTHER.

The following variables are declared in this project.

```
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 retain data. The following code can be modified by using the customized variables

```
(* Add your REAL variables for retain here *)
(* ------ *)
My_REAL_00 := REAL_VAR_now[0];
My_REAL_01 := REAL_VAR_now[1];
My_REAL_02 := REAL_VAR_now[2];
My_REAL_03 := REAL_VAR_now[3];
My_REAL_04 := REAL_VAR_now[4];
My_REAL_05 := REAL_VAR_now[5];
My_REAL_06 := REAL_VAR_now[6];
My_REAL_07 := REAL_VAR_now[7];
(* ------ *)
```

(\* Add your integer, BOOL, Timer variables for retain here \*)
(\* ...... \*)
My\_DINT\_00 := OTHER\_VAR\_now[0];
My\_DINT\_01 := OTHER\_VAR\_now[1];
My\_DINT\_02 := OTHER\_VAR\_now[2];
My\_DINT\_03 := OTHER\_VAR\_now[3];
My\_DINT\_04 := OTHER\_VAR\_now[4];
My\_DINT\_05 := OTHER\_VAR\_now[5];

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

### (\* 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:

Make sure that the communication IP is correct and download the project to the PAC. In the Spy List window, all values are "0" (or "FALSE"). Change some values, and the .txt file (i.e., retain\_real.txt or retain\_other.txt) will automatically be created in the specified folder on PAC, and then write data to the file.

**Note:** "Save\_file\_counter" displays the cumulative number of times for updating the file. The project does not apply to the needs of updating the file frequently under the "/System\_disk/..." path.

🎹 🏭 😘 😨 🏠 🛋 💱 🛛	🕒 🅍 🚮 RUN 👘	🗱 👬 🔹 u 🔏 🌼
NewSpy1.spl		Cycle time (ms):
Name	Value	Last - 1
Save_file_counter	5	Allowed = 0 Maximum = 13
— My_REAL_00	4.65	Overflow = 0
I My_REAL_01	0.0 W/r	ite data five times.
My_REAL_02	0.0	0
My_REAL_03	0.0	CRC=16#7f47c4c4 Target application:
My_REAL_04	0.0	V16 - 23/02/2021 - 16:04:10
My_REAL_05	0.0	CRC=16#7f47c4c4 RAM = 312008 bytes
My_REAL_06	0.0	Elapsed: 6m29s
My_REAL_07	0.0	
My_DINT_00	1357	
My_DINT_01	2468	
My_DINT_02	0	
My_DINT_03	0	
My_DINT_04	0	
My_DINT_05	0	
My_BOOL_00	TRUE	
My_BOOL_01	FALSE	
My_BOOL_02	FALSE	
My_Timer_00	t#1h30m50	5
My_Timer_01	t#0s	
My_Timer_02	t#0s	
My_Timer_03	t#0s Do	uble-click to enter the value.
My_Timer_04	t#0s	
My_Timer_05	t#0s	
Go retain Variables loa	ad retain 🖉 save retain	NewSpy1

To check files in the Win-GRAF PAC by using the SSH tool (refer to Section 13.2)

Browse File System on Remote Host	×
/System_Disk/Win-GRAF retain_other.txt retain_real.txt 5.cod	•
File name: Open	Cancel

## 6.3 Save the Retain Data to a FRAM

RPAC-2658M built-in a FRAM for users to read/write data that will not be lost due to PAC shutdown.

### FRAM has the following features:

Advantages: Besides the retain variable, there is another way to save important data.

### **Disadvantages:**

When using the retain variable, the CPU time is much less than 1 ms. It takes much more CPU time to read or write data in a FRAM. **Do not** call the "EEP\_Read" or "EEP\_Write" function frequently, or it will increase the cycle time of the PAC.

### <u>Note</u>:

EEP\_Read() and EEP\_Write() functions can be used to read/write data from/to a FRAM or EEPROM.

### ST Program:

```
(* 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 memory 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 programming: write to the memory 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;
```

#### (\* Hazardous: The memory may be destroyed very soon. \*)

```
(* 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 memory once in the first Cycle. *)
if FIRST_CYCLE then
FIRST_CYCLE then
FIRST_CYCLE := FALSE; (* means it is not the first Cycle any more *)
tmp_bool := EEP_Read (1, New_Val);
end_if;
(* Hazardous programming: Write the "New_Val" value to the memory one time in every Cycle. *)
```

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

## 6.3.1 EEP\_READ (Read a Value from the FRAM)



Tip: Press "F1" key to see the detailed setting descriptions.

Addr: (Data Type: "DINT")

The address can be 1 to 1200.

When using the 64-bit data type (e.g., LINT or LREAL), set the "Addr" to 1001 to 1200.

## @Name :

Specify a variable name to store data read from the FRAM.

DO NOT use string variable. The data 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 data type of the variable is REAL or LREAL but the read data is not a REAL value or an error occurs, "Q" will return FALSE. The variable value "0.0" indicates the data is not a REAL value.

## 6.3.2 EEP\_WRITE (Write a Value to the FRAM)



Addr: (Data Type: "DINT")

The address can be 1 to 1200. If the variable type of the "Value" parameter is 64-bit data (e.g., LINT or LREAL), the "Addr" can be 1001 to 1200 only.

## Value :

Specify the value to write to the FRAM.

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:

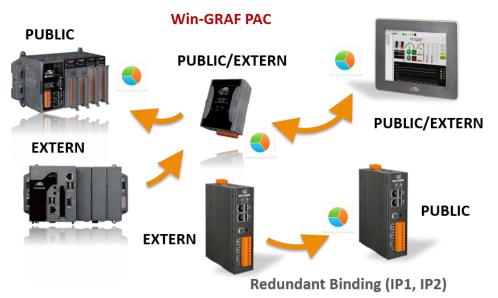
Data Type: BOOL. TRUE: Ok; FALSE: Error.

# **Chapter 7** Exchange Data between PACs (Data Binding)

The Binding feature allows multiple Win-GRAF PACs to exchange data to each other in an event-triggered manner that is more efficient than polling. Win-GRAF offers two ways to configure Binding:

- **PUBLIC:** To make the data publicly available for other PACs or the C program in the same PAC.
- **EXTERN:** To get data from the other PAC.

#### **Application Diagram:**



#### Note:

The maximum number of "Binding" features available for each RPAC-2658M is 32 (EXTRN).

The "PUBLIC" Setting: Make the variable data accessible to other PACs.

1. Click the "Open Binding Configuration" button to open the "Binding" window.

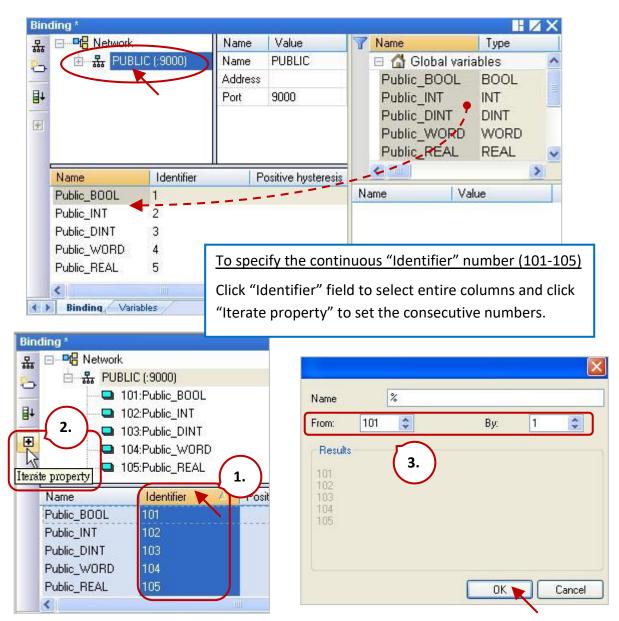
"PUBLIC (: 9000)" indicates that the variable data is made public via the port number "9000" (the number is fixed, do not change it).

Win-GRAF - demo_binding	Open	Binding Configura	ation – 🗆 🗙
File Edit View Insert Project To	ools Window Help		
23   🛃 🖻   🎒   🔏 🖪 🛍 🗙 🤿	:   🏷 🕑 🔚 🏢 👪 🚺 🦉 🙆	🖻 🛐 📴 🚰 🔊	
Workspace	Binding		
🖓 🔮 demo_binding	品 ⊡ <mark>G</mark> Network	Name Value	Type
Exception programs	📇 PUBLIC (:9000)	Name PUBLIC	🗆 🔂 Global variables 🛛 🔺
🖻 🛄 Programs		Address	Public_BOOL BOOL
🛄 Main	≣+	Port 9000	Public_INT INT
🖃 🛄 Watch (for debugging)	· · · · · · · · · · · · · · · · · · ·		Public_DINT DINT
Soft Scope			Public_WORD WORD
🔤 Initial values	Keep the	Address blank.	Public REAL REAL V
Binding Configuration			< >
🛛 🕺 🛛 Global defines	Name Identifier	Positive hysteresis N	Name Value
📲 🛗 Variables			· · · · · · · · · · · · · · · · · · ·
- Types			
(All Projects)		>	
	Variables Binding		

2. Before making the variable data public, adding these variables first. The following table lists all the necessary variables in this example.

Variables Name	Data Type
Public_BOOL	BOOL
Public_INT	INT
Public_DINT	DINT
Public_WORD	WORD
Public_REAL	REAL

- 3. Click on the "PUBLIC (: 9000)" and drag-and-drop the needed variables into the ID mapping area, then the ID will automatically be generated. For other PACs, using the same ID to read/write data on this PAC.
  - Note: Up to 8192 variables can be used in the Public setting. The number of "Identifier" can only be "1 to 8192".



The "EXTERN" Setting: To get data from the other PAC.

4. Click the "Insert Master/Port" button on the left and the "Group" window will appear. Follow the description below to set these fields and click the "OK" button.

	Binding				
	<mark>뮸</mark> ⊟ ■ <mark>문</mark> Network		Name	Value	
	🛒 🗼 🗄 🖳 🏪 PUBLIC (:900	)0)	Name	PUBLIC	
			Address		
Group	<b>D</b> .	×	Port	9000	
Gloup		^			
Name		ОК 🍗			
EXTERN:1		Conned			
		Cancel			
Address				PAC1	PAC2
Address	192.168.255.1/10.0.255.1				
				- <b>-------------</b>	्रिक्
Port	9000		LAN3	LAN	N1

Name: Enter a custom name.

Address: Enter the IP address of the PAC that you want to get data from (e.g., "192.168.255.1"). Two IP Addresses (LAN1: 192.168.x.x, LAN3: 10.0.x.x) can be set if the PAC has two Ethernet ports. When one IP address is nonfunctional, it will try to connect to the second IP address.

Port: The fixed port number "9000", do not change it.

5. In the Variable area, add variables for getting data. The table lists the needed variables in this example.

Variables Name	Data Type
PAC1_BOOL	BOOL
PAC1_DINT	DINT
PAC1_REAL	REAL
Error_Status	DINT

6. Drag-and-drop these variables into the ID mapping area of the "EXTERN:1" and the identifier will automatically be generated. Change to the same ID as the public variable in remote public PAC.

Bin	ding *									∡×
恭	⊡ <b>¤¦a</b> Network	¢		Name	Value	7	Name	Туре		
5	🗄 🚟 🖁 PUBI	UC (:9000)			Name	EXTERN:1		Public REAL	REAL	^
	🖻 品 EXTE	RN:1 (192.10	58.255.1/10.0.255.1:9	000)	ddress	192.168.255.1		PAC1_BOOL	BOOL	
∎∔					Port	9000		PAC1_DINT	DINT	
+								PAC1_REAL	REAL	
							/1	Error_Status	DINT	J~
		Lin ee a			11	/				>
	Name	Identifier 🛆	Destination			11	L NL	ame Variable	area 📘	
	PAC1_BOOL	1	Exchange data					ame		
	PAC1_DINT	2	Exchange data		`					
	PAC1_REAL	3	Exchange data							
	Error_Status	4	Exchange data							
< >	Variables Bind	ling/								

7. The "Error\_Status" variable is used to know the communication status of the PAC. Set the ID to "0" and double-click its "Destination" field, then set it as "Connection error status".

Bind	ding *									
恭	□ <sup>□</sup> <sup>□</sup> <sup>□</sup> <sup>□</sup> <sup>□</sup> <sup>□</sup> <sup>□</sup> <sup>□</sup> <sup>□</sup>	k	Name	Value						
8	🖻 🚠 PUB	LIC (:9000)		Name	EXTERN:1					
	🗄 🚠 EXTE	ERN:1 (192.1)	58.255.1/10.0.255.1:9000)	Address	192.168.255.1					
₽Ļ			Port	9000						
+										
	Name	Identifier 🔺	Destination							
	PAC1_BOOL	1	Exchange data							
	PAC1_DINT	2	Exchange data							
	PAC1_REAL	3	Evenandata							
	Error_Status	0	Exchange data Variable error status							
			Variable date stamp							
	Variable time stamp Connection error status 🝗									
< >										
	variables bin	unių								

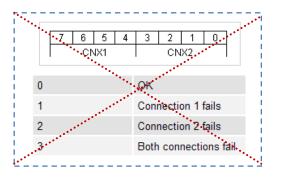
### Note:

If two IP addresses are set in the EXTERN group, "Error\_Status" will return two communication statuses (8-bit). Bit 0-3 represents the connection status of the first IP and its value is 15 if all bits are 1; Bit 4-7 represents the connection status of the second IP and its value is 240 if all bits are 1. As long as the value is not equal to "0" which means the communication is abnormal.

IP2	Connec	tion Sta	itus	IP1	Connec	tion Sta	atus	Status Description				
27	<b>2</b> <sup>6</sup>	<b>2</b> <sup>5</sup>	<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				
	(	)			(	כ	Connection OK					
	(	)			≠0 (1	~ 15)		IP1 Connection error				
	≠ 0 (16 ~ 240)				(	)		IP2 Connection error				
	≠	0			≠	0		IP1 and IP2 Connection error				

#### Notice:

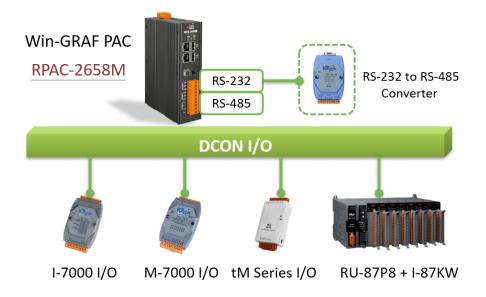
In "HTML Help" (press "F1"), the description of the **"Connection status**" and **"Variable status"** in the topic "Networked applications - Dual binding on redundant ETHERNET" **is wrong**, please ignore it. It does not conform to the usage of Win-GRAF PAC.



# Chapter 8 Connecting DCON I/O Modules

Win-GRAF PACs can connect to remote DCON I/O modules via the COM Port (RS-485). Up to 16 DCON ports can be enabled on a PAC, and up to 50 remote DCON modules can be connected to each port (no more than 32 recommended).

For "I-87KW" series I/O modules, an RS-485 I/O expansion unit is required (e.g., RU-87P4/8).



Product Page - <u>Remote I/O Module and Unit</u>

Product Page - DCON Utility Pro

Before using remote DCON I/O modules, configuring each one with "DCON Utility Pro" software.

Parameter	The Setting Value
Protocol	DCON
Address	1 to 255
Baudrate	'9600' is recommended and must the same as the PAC setting
Checksum	'Enabled' is recommended and must the same as the PAC setting
Data format and I/O settings	Set them according to application requirements

#### Note:

- 1. When using Analog Input (AI) modules, set the Data format to "2's Complement".
- 2. When using the Analog Output (AO) module, set the Data format to "Engineering".

# 8.1 Setting the "DCON" I/O Board

"DCON" I/O board allows enabling one RS-485 port to connect remote DCON I/O modules. Up to 16 DCON ports can be enabled for one PAC.

1. Click the "Open I/Os" button to open the "I/O Boards" window.

					<u>W</u> indow				$\frown$								
2	2	3 X	te C	X	x 8	5	3	<b>T</b>	-	뷺	96	5	2	§g	ß	1	ß
										$\overline{}$							

2. Double-click the slot number to add the "DCON" I/O board, and then double-click "DCON" to open the "Properties" window.

IIII I/O Boards	
0	Close
2 3 4 5 6 7 8 DCON 9 10 11 12 13	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
14 15 16 17	Setting Descriptions Enable one serial port (RS-485) to connect remote DCON I/Os (I-7000 series modules , RU-87PE Note: 1. 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".

#### Parameters:

<u>Note:</u> When using DCON protocol, set the data format of I/O modules as "N,8,1" which stands for no parity-bit, eight data-bit, and one stop-bit.

Port:	COM port number (1 $\sim$ 37, depends on the PAC.)		
Baud_rate:	Communication baud rate in bps, can be set to 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200 (bps). '9600' will be used if setting the other value.		
Host_watchdog_Enabled:	1: Host-watchdog is enabled. 0: Disabled. '1' will be used if setting a non-zero value.		

Watchdog_timeout:	Unit: ms, can be set to 3000 to 25500 (i.e., 3 to 25.5 seconds). '3000' will be used if the setting is less than 3000; '25500' will be used if the setting is larger than 25500. If "Host_watchdog_Enabled" is set to "0", this setting will be ignored.
Checksum_enabled:	1: Checksum is enabled. 0: Disabled.
	'1' will be used if setting a non-zero value.
	(For communication security, it is recommended to enable Checksum)
Delay_ms_between_polls:	Unit: ms, the default value is 0. Valid range is 0 to 1000.
	'0' will be used if the setting is less than 0;
	'1000' will be used if the setting is larger than 1000.
	If no wireless module is connected, set to a smaller value (e.g., 0 to 10).
	If <u>wireless modules</u> are connected (e.g., the ZigBee I/O), set to a larger
	value (e.g., 30 to 100 or other values). The larger the value, the slower
	the Polling efficiency will be.

3. Double-click any item to enter the value.

3: DCON - Properties			×
Key = 6 Ref = 16#5 Port = 2 Baud_rate = 9600 Host_watchdog_Enabled = 1 Watchdog_timeout = 3000 Checksum_enabled = 0			
<b>Delay_ms_between_polls</b> =10	Delay ms between polls	×	
Reserved0 = 0			
Reserved1 = 0	10		
Reserved2 = 0 Reserved3 = 0			

4. After adding the "DCON" in the "I/O Boards" window, a "BOOL" input variable will be automatically added to the "Variables" window. The variable is used to display the communication status of the COM Port on Win-GRAF PAC. (TRUE: OK; FALSE: error.).

Va	riables										
7	Name	A	Туре	Dim.	Attrib.	Sy	<b>D</b> .	Init value	User	Tag	Description
	🖽 🚮 Globa	al variable	es								
	🛃 RETA	NN variat	oles								
ſ	🗉 📓 %IX8	- DCON									
L	%IX8.0		BOOL		Input	Γ					

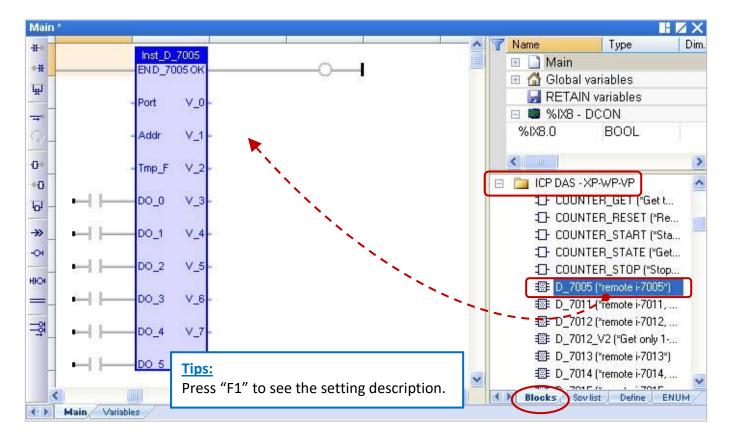
# 8.2 Using I/O Function Blocks

Win-GRAF supports many remote DCON I/O modules, open the "Library Manager" (Refer to Section 1.2.2) or the Help (press "F1") to find out the description of these function and function blocks.

This section describes the usage of "<u>D 7065</u>", "<u>D 7018Z</u>", "<u>D 7083</u>", "<u>D 87084 freq</u>", "<u>D 87084 cnt4</u>", "<u>D 87084 cnt8</u>", "<u>D 100T485</u>", and "<u>D GPS721</u>" function blocks.

Model	Description
<u>I-7065D-G</u>	4-ch Isolated DI and 5-ch Power Relay Module with LED Display
<u>I-7065AD-G</u>	4-ch Isolated DI and 5-ch AC-SSR Output Module with LED Display
<u>I-7065BD-G</u>	4-ch Isolated DI and 5-ch DC-SSR Output Module with LED Display
<u>I-7018Z-G/S</u>	10-ch Thermocouple Input Module
<u>I-7083D-G</u>	3-axis, 32-bit Encoder/Counter Input Module with LED Display
<u>I-7083BD-G</u>	3-axis, 32-bit Encoder/Counter Input Module with Virtual Battery Backup and LED Display
<u>I-87084W-G</u>	4/8-ch Counter/Frequency/Encoder Input Module
DL-100T485	IP66 Remote Temperature and Humidity Data Logger with LCD Display (RS-485)
<u>GPS-721</u>	GPS Receiver Module with RS-485 (Asia Only)

In the **Blocks** pane of the LD Program, there are many functions and function blocks in the "ICP DAS - XP-WP-VP" folder. Drag and drop the desired function to the program editing area to use it.



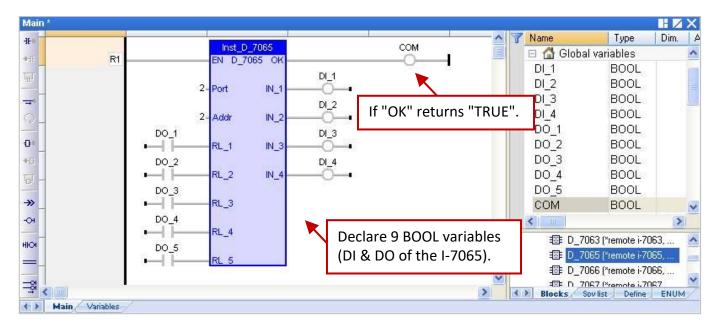
## 8.2.1 "D\_7065" Function Block

"D\_7065": Used to connect a remote I-7065D, I-7065AD, or I-7065BD DCON module.

### Note:

- 1. Make sure that the I/O module has been configured by using DCON Utility Pro (refer to Chapter 8).
- 2. Make sure that "DCON" has been added in the "I/O boards" window (refer to Section 8.1).
- Download <u>the demo program</u> (demo\_d\_7065.zip) on the website. Click the menu commands
   File Add Existing Project From Zip... in the Workbench to open the project.
- 4. Only when the communication status of the module is normal (TRUE), the received DI data is correct.

## Example: Connect the I-7065 (Addr. = 2) via COM2 on PAC. It has 4 DI and 5 Relay output channels.



Input Parameters	Data Type	Description
EN	BOOL	TRUE: Enabled; FALSE: Disabled.
Port	DINT	COM port number (can be 1 to 37, depends on PAC). Note that specify a constant value, not a variable value.
Addr	DINT	The Net-ID address of the module can be 1 to 255. Note that specify a constant value, not a variable value.
RL_1 to RL_5	5-Ch DO values.	

Output Parameters	Data Type	Description
ОК	BOOL	TRUE: Communication is Ok. FALSE: Communication failed.
IN_1 to IN_4	BOOL	4-Ch DI values.

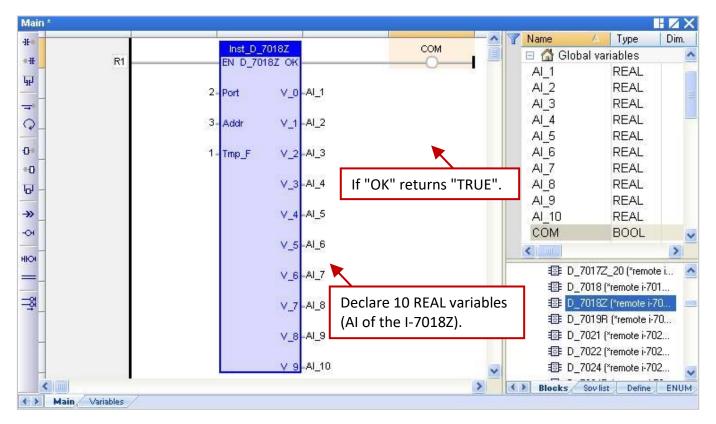
### 8.2.2 "D\_7018Z" Function Block

"D\_7018Z": Used to connect a remote I-7018Z DCON module to measure voltage, current, or temperature.

#### Note:

- Make sure that the I/O module has been configured by using DCON Utility Pro (refer to Chapter 8). Set the data format of the AI module to "2's complement", or Win-GRAF cannot read data correctly.
- 2. Make sure that "DCON" has been added in the "I/O boards" window (refer to Section 8.1).
- Download <u>the demo program</u> (demo\_d\_7018z.zip) on the website. Click the menu commands
   File Add Existing Project From Zip... in the Workbench to open the project.
- 4. Only when the communication status of the module is normal (TRUE), the received AI data is correct.

#### Example: Connect the I-7018Z (Addr. = 3) to measure the Celsius temperature via COM2 on PAC.



Input Parameters	Data Type	Description	
EN	BOOL	TRUE: Enabled; FALSE: Disabled.	
Port	DINT	COM port number (can be 1 to 37, depends on PAC). Note that specify a constant value, not a variable value.	
Addr	DINT	The Net-ID address of the module can be 1 to 255. Note that specify a constant value, not a variable value.	
Tmp_F	DINT	Temperature Format can be 1 or 2: 1: Celsius. 2: Fahrenheit. 'Celsius' will be used if setting it as another value.	

Output Parameters	Data Type	Description
ОК	BOOL	TRUE: Communication is Ok. FALSE: Communication failed.
		<ul><li>10-Ch AI value. The type code can be set in DCON Utility Pro.</li><li>If a Temperature type is set, the unit of data is 'Degree'.</li><li>E.g., the value of 25.75 stands for 25.75 degrees.</li></ul>
V_0 to V_9	REAL	If a Voltage type is set, the unit of data is 'V'. E.g., the value 0.85421 stands for 0.85421 V or 854.21 mV. If a Current type is set, the unit of data is 'mA'.
		E.g., the value of 1.5567 stands for 1.5567 mA.

#### **Open-wire Detection:**

If the temperature data is greater than "9000.0", it means that

- 1. The temperature sensor may be broken-line.
- 2. The temperature sensor may be broken.
- 3. The setting on the DCON module does not match it on the temperature sensor.
- 4. The resistance measured by the sensor is incorrect.

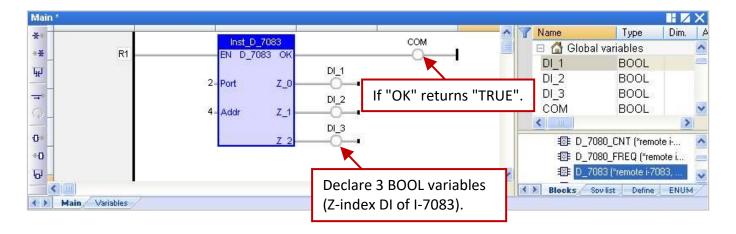
### 8.2.3 "D\_7083" Function Block

"D\_7083": Used to connect a remote I-7083D or I-7083BD DCON module.

#### Note:

- Make sure that the I/O module has been configured by using DCON Utility Pro (refer to Chapter 8). Set the data format of the AI module to "2's complement", or Win-GRAF cannot read data correctly.
- 2. Make sure that "DCON" has been added in the "I/O boards" window (refer to Section 8.1).
- Download <u>the demo program</u> (demo\_d\_7083.zip) on the website. Click the menu commands
   File Add Existing Project From Zip... in the Workbench to open the project.
- 4. Only when the communication status of the module is normal (TRUE), the received DI data is correct.
- 5. To get the Encoder value from the I-7083D/BD module, using the "**D\_7083**" function block and the **"Counter\_xxxx"** function (refer to Section 8.3) to operate the Encoder function.

#### Example: Connect the I-7083 (Addr. = 4) via COM2 on PAC. It has 3 DI channels



Input Parameters	Data Type	Description	
EN	BOOL	TRUE: Enabled; FALSE: Disabled.	
Port	DINT	COM port number (can be 1 to 37, depends on PAC). Note that specify a constant value, not a variable value.	
Addr	DINT	The Net-ID address of the module can be 1 to 255. Note that specify a constant value, not a variable value.	

Output Parameters	Data Type	Description	
ОК	BOOL	TRUE: Communication is Ok. FALSE: Communication failed.	
Z_0 to Z_2	BOOL	3-axis Z-index DI value.	

### 8.2.4 "D\_87084\_FREQ" Function Block

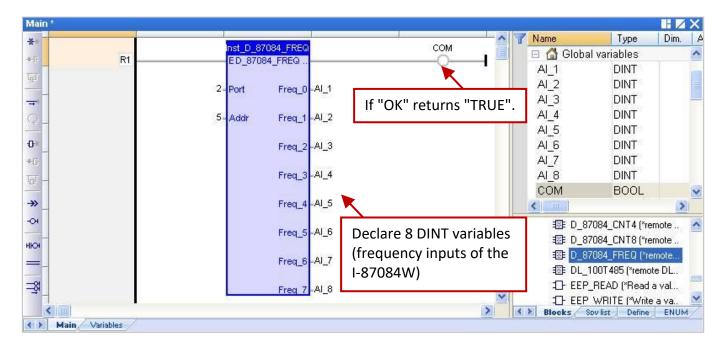
"D\_87084\_freq": Used to connect a remote I-87084W in an RS-485 I/O expansion unit (e.g., RU-87P4/8 or I-87K4/5/8/9) to measure frequency values.

#### Note:

- Make sure that the I/O module has been configured by using DCON Utility Pro (refer to Chapter 8). Set the frequency format of I-87084W to "Hex format", or the function will not work.
- 2. Make sure that "DCON" has been added in the "I/O boards" window (refer to Section 8.1).
- Download <u>the demo program</u> (demo\_d\_87084\_fr.zip) on the website. Click the menu commands
   File Add Existing Project From Zip... in the Workbench to open the project.
- 4. Only when the communication status of the module is normal (TRUE), the received AI data is correct.

#### Example:

Connect the I-87084W (Addr. = 5) via COM2 on PAC to measure frequency values of eight channels.



Input Parameters	Data Type	Description
EN	BOOL TRUE: Enabled; FALSE: Disabled.	
Port	DINT	COM port number (can be 1 to 37, depends on PAC). Note that specify a constant value, not a variable value.
Addr	DINT	The Net-ID address of the module can be 1 to 255. Note that specify a constant value, not a variable value.

Output Parameters	Data Type	Description	
ОК	BOOL	TRUE: Communication is Ok. FALSE: Communication failed.	
Freq_0 to Freq_7	DINT	8-Ch frequency value, unit is Hz.	

### 8.2.5 "D\_87084\_CNT4" Function Block

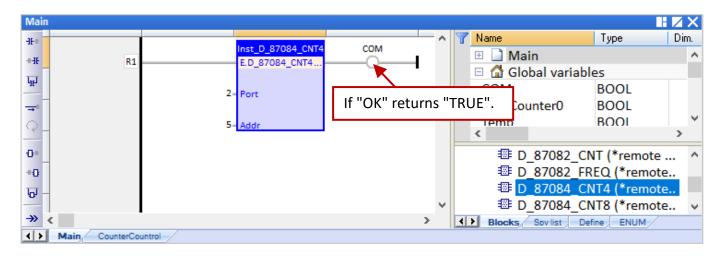
"D\_87084\_CNT4": Used to connect a remote I-87084W in an RS-485 I/O expansion unit (e.g., RU-87P4/8 or I-87K4/5/8/9) to measure the counter value of four channels.

#### Note:

- Make sure that the I/O module has been configured by using DCON Utility Pro (refer to Chapter 8). Set the data format of I-87084W to "Hex format", or the function will not work.
- 2. Make sure that "DCON" has been added in the "I/O boards" window (refer to Section 8.1).
- 3. Download <u>the demo program</u> (demo\_d\_87084\_c4.zip) on the website. Click the menu commands File - Add Existing Project - From Zip... in the Workbench to open the project.
- 4. Only when the communication status of the module is normal (TRUE), the count value is correct.
- 5. To get the counter value from the I-87084W module, using the **"D\_87084\_CNT4"** function block and the **"Counter\_xxxx"** function (refer to Section 8.3) to operate the Counter function.

#### Example:

#### Connect the I-87084W (Addr. = 5) via COM2 on PAC to measure the counter value of four channels.



Input Parameters	Data Type	Description
EN	BOOL TRUE: Enabled; FALSE: Disabled.	
Port	DINT COM port number (can be 1 to 37, depends on PAC). Note that specify a constant value, not a variable value.	
Addr	DINT	The Net-ID address of the module can be 1 to 255. Note that specify a constant value, not a variable value.

Output Parameters	Data Type	Description
ОК	BOOL	TRUE: Communication is Ok. FALSE: Communication failed.

### 8.2.6 "D\_87084\_CNT8" Function Block

"D\_87084\_CNT8": Used to connect a remote I-87084W in an RS-485 I/O expansion unit (e.g., RU-87P4/8 or I-87K4/5/8/9) to measure the counter value of eight channels.

#### Note:

- Make sure that the I/O module has been configured by using DCON Utility Pro (refer to Chapter 8). Set the data format of I-87084W to "Hex format", or the function will not work.
- 2. Make sure that "DCON" has been added in the "I/O boards" window (refer to Section 8.1).
- Download <u>the demo program</u> (demo\_d\_87084\_c8.zip) on the website. Click the menu commands
   File Add Existing Project From Zip... in the Workbench to open the project.
- 4. Only when the communication status of the module is normal (TRUE), the count value is correct.
- 5. To get the counter value from the I-87084W module, using the **"D\_87084\_CNT8"** function block and the **"Counter\_xxxx"** function (refer to Section 8.3) to operate the Counter function.

#### Example:

#### Connect the I-87084W (Addr. = 5) via COM2 on PAC to measure the counter value of eight channels.

Main						ZΧ
-1[-+		COM	^ 7	Name	Туре	Dim.
+-IF R1	Inst_D_87084_CNT8 ED 87084 CNT8)			🖽 🗋 Main		^
<u> </u>		× •		🗆 🚮 Global vari		_
	2- Port			<u>- com</u>	BOOL	
₽		If "OK" returns	"TRU	E". Counter0	BOOL	~
Q	5- Addr					>
-0+				Image: D_87082	_FREQ (*re	~
-					_CNT4 (*re	
+0				<b>I D_8708</b> 4	_CNT8 (*re	
ਯ –				Image: Big D_87084	_FREQ (*re	
			× _		1 (*remote	
→» <		>		Blocks Sov list	Define ENUM	
Main CounterContro						

Input Parameters	Data Type	Description
EN	BOOL TRUE: Enabled; FALSE: Disabled.	
Port	DINT COM port number (can be 1 to 37, depends on PAC). Note that specify a constant value, not a variable value.	
Addr	DINT	The Net-ID address of the module can be 1 to 255. Note that specify a constant value, not a variable value.

Output Parameters	Data Type	Description
ОК	BOOL	TRUE: Communication is Ok. FALSE: Communication failed.

### 8.2.7 "DL\_100T485" Function Block

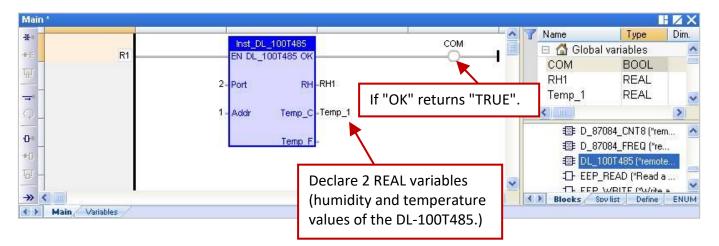
"DL\_100T485": Used to connect a remote DL-100T485 module to get humidity and temperature value.

#### Note:

- Make sure that the I/O module has been configured by using <u>DL Utility</u>. DL-100T485's default value: Address is "1", Baudrate is "9600", and Checksum is "Disable". (<u>Download DL-100T485 user manual</u>)
- 2. Make sure that "DCON" has been added in the "I/O boards" window (refer to Section 8.1).
- Download <u>the demo program</u> (demo\_dl\_100T485.zip) on the website. Click the menu commands File - Add Existing Project - From Zip... in the Workbench to open the project.
- 4. Only when the communication status of the module is normal (TRUE), the AI value is correct.

#### Example:

#### Connect the DL\_100T485 (Addr. = 1) via COM2 on PAC to get humidity and temperature value



Input Parameters	Data Type	Description
EN	<b>EN</b> BOOL TRUE: Enabled; FALSE: Disabled.	
Port	DINT	COM port number (can be 1 to 37, depends on PAC). Note that specify a constant value, not a variable value.
Addr	DINT	The Net-ID address of the module can be 1 to 255. Note that specify a constant value, not a variable value.

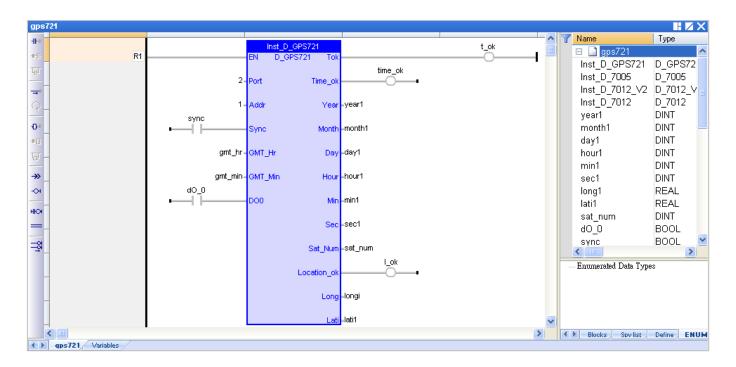
Output Parameters	Data Type	Description
ОК	BOOL	TRUE: Communication is Ok. FALSE: Communication failed.
RH	REAL	"Relative humidity"; the unit is 1%. E.g., the value of 45.7 stands for 245.7 %.
Temp_C	REAL	The temperature C in degrees Celsius. E.g., the value of 25.7 stands for 25.7 °C.
Temp_F	REAL	The temperature F in degrees Fahrenheit. E.g., the value of 78.26 stands for 78.26 °F.

### 8.2.8 "D\_GPS721" Function Block

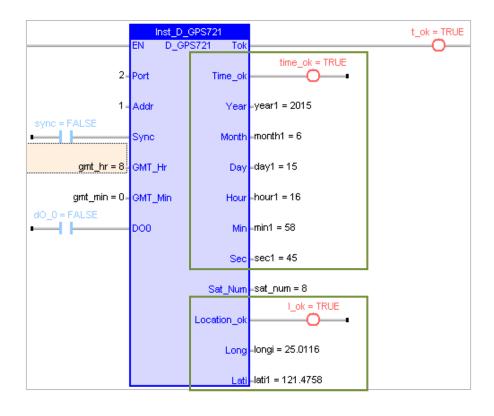
Using the "D\_GPS721" function to link one "GPS-721" remote GPS receiver module (includes one DO and one PPS outputs).

#### Note:

- 1. Only one GPS-721 module can be used for one PAC.
- 2. Make sure that the module has been configured by using DCON Utility Pro (refer to Chapter 8). GPS-721's default value: Address is "1", Baudrate is "9600", and Checksum is "Disable".
- 3. Make sure that "DCON" has been added in the "I/O boards" window (refer to Section 8.1).
- Download <u>the demo program</u> (dmeo\_gps721.zip) on the website. Click the menu commands File - Add Existing Project - From Zip... in the Workbench to open the project.



Input Parameters	Data Type	Description
EN	BOOL	TRUE: Enabled; FALSE: Disabled.
Port	DINT	COM port number (can be 1 to 37, depends on PAC). Note that specify a constant value, not a variable value.
Addr	DINT	The Net-ID address of the module can be 1 to 255. Note that specify a constant value, not a variable value.
Sync	BOOL	Time synchronization, TRUE: Enabled; FALSE: Disabled. The PAC time will be automatically corrected if there is a 5-second (or more) time difference between a PAC and a GPS-721.
GMT_Hr & GMT_Min	DINT	Time Zone. for example, Beijing and Taipei are <b>GMT+8</b> ( <b>GMT_Hr = 8</b> , GMT_Min = 0), USA is <b>GMT-6</b> ( <b>GMT_Hr = -6</b> , GMT_Min = 0), and New Delhi is is <b>GMT+5.5</b> ( <b>GMT_Hr = 5, GMT_Min = 30</b> ).
DO0	BOOL	Digital output channel of the GPS-721 module



Output Parameters	Data Type	Description
Tok	BOOL	TRUE: Communication of GPS-721 is Ok; FALSE: Communication failed, the return values below are invalid.
Time_ok	BOOL	TRUE: The value of Year, Month, Day, Hour, Min, and Sec are valid. FALSE: The value of Year, Month, Day, Hour, Min, and Sec are invalid.
Year ` Month Day ` Hour Min ` Sec	DINT	Year (2009 to), Month (1 to 12), Day (1 to 31), Hour (0 to 23), Minute (0 to 59), and (0 to 59). Note: "Time_ok" will be automatically set to "FALSE" from 23:59:00 to 00:00:59 (2 minutes) every day and time synchronization will not be performed.
Sat_Num	DINT	The number of satellites in use. 0: No satellite is found. Or, 1 to 9 satellites are in use.
Location_ok	BOOL	<ul> <li>Note: Only when "Location_ok" is "TRUE", the value of Long and Lati is valid.</li> <li>TRUE:</li> <li>GPS-721 has got the latitude/longitude of the current position.</li> <li>FALSE: the value of Long and Lati is valid.</li> </ul>
Long	REAL	Longitude (Positive: The East, Negative: The West). E.g., the value of 25.0121 stands for 25.0121° E.
Lati	REAL	Latitude (Positive: The North, Negative: The South). E.g., the value of "121.4576" stands for 121.4576° N.

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

This section lists the way to perform the Counter/Encoder function by using the "COUNTER\_START", "COUNTER\_STOP", "COUNTER\_GET", "COUNTER\_STATE", and "COUNTER\_RESET" functions in the LD or ST program.

#### Note:

- Adding the I/O function block before using the "COUNTER\_XXX" function, or it will not work.
- The I-87K I/O module can be plugged into an RS-485 I/Oexpansion unit (e.g., RU-87P4/8).

Model	Function Block	Encoder	Counter	Frequency
<u>I-7080</u>	"D_7080"		$\checkmark$	$\checkmark$
<u>I-7083</u>	"D_7083"	✓ (Section 8.2.3)	1	
<u>I-87082W</u>	"D_87082"		√	~
<u>I-87084W</u>	"D_87084_cnt4" or "D_87084_cnt8"	~	✓ (Section 8.2.5)	✓ (Section 8.2.4 )

#### 8.3.1 COUNTER\_START

**Description**: Start the counting of a counter or encoder.

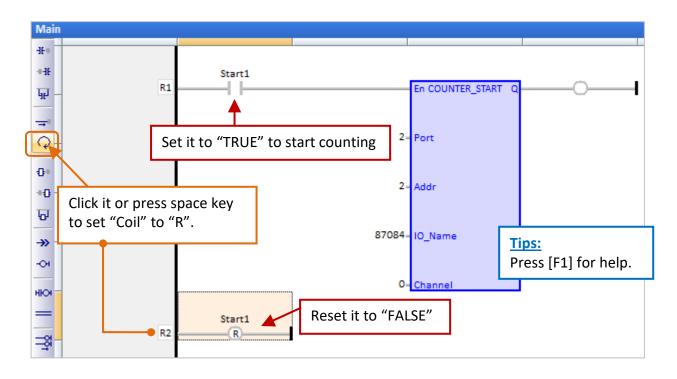
#### ST program:

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

**Note:** "Start1" and "TMP\_BOOL" Boolean variables can be declared in the Variables area.

#### LD program:

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



Input Parameters	Data Type	Description
Port	DINT	COM port number (can be 1 to 37, depends on PAC).
Addr	DINT	The Net-ID address of the module can be 1 to 255.
IO_Name	DINT	The name of relative Counter/Encoder module. it can be set to "87084", "87082", "7083" and "7080".
Channel	DINT	The channel number of the Counter/Encoder module. it can be set to "0", "1", and so on.

Output Parameters	Data Type	Description
Q	BOOL	TRUE: Ok. FALSE: Error.

### 8.3.2 COUNTER\_STOP

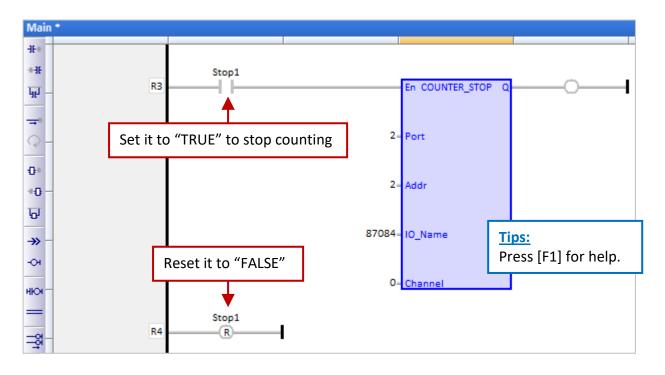
**Description:** Stop the counting of a counter or encoder.

**<u>ST program</u>**: "Start1" and "TMP\_BOOL" boolean variables can be declared in the Variables area.

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

#### LD program:

"Stop1" (BOOL): set it to "TRUE" to stop counting and then reset the "Stop1" to "FALSE".



Input Parameters	Data Type	Description
Port	DINT	COM port number (can be 1 to 37, depends on PAC).
Addr	DINT	The Net-ID address of the module can be 1 to 255.
IO_Name	DINT	The name of relative Counter/Encoder module. it can be set to "87084", "87082", "7083" and "7080".
Channel	DINT	The channel number of the Counter/Encoder module. it can be set to "0", "1", and so on.

Output Parameters	Data Type	Description
Q	BOOL	TRUE: Ok. FALSE: Error.

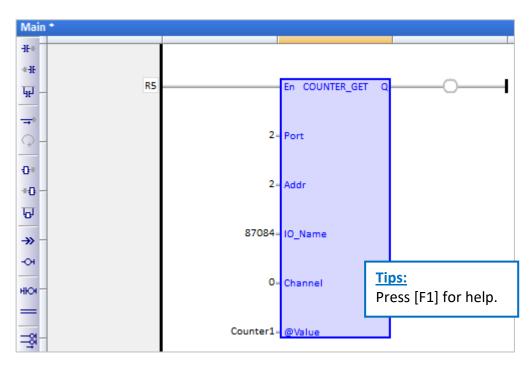
### 8.3.3 COUNTER\_GET

**Description:** Get a counter or an encoder value.

**<u>ST program</u>**: "TMP\_BOOL" (BOOL) and "Counter1" (DINT) can be declared in the Variables area.

TMP\_BOOL := Counter\_Get (Port, Addr, IO\_Name, Channel, Counter1);

#### LD program:



Input Parameters	Data Type	Description
Port	DINT	COM port number (can be 1 to 37, depends on PAC).
Addr	DINT	The Net-ID address of the module can be 1 to 255.
IO_Name	DINT	The name of relative Counter/Encoder module. it can be set to "87084", "87082", "7083" and "7080".
Channel	DINT	The channel number of the Counter/Encoder module. it can be set to "0", "1", and so on.
@Value	"DINT", "UDINT", "DWORD", or "LINT"	Return the value of a counter or an encoder. Refer to <u>Appendix A</u> for the range of data.

Output Parameters	Data Type	Description
Q	BOOL	TRUE: Ok. FALSE: Error.

### 8.3.4 COUNTER\_STATE

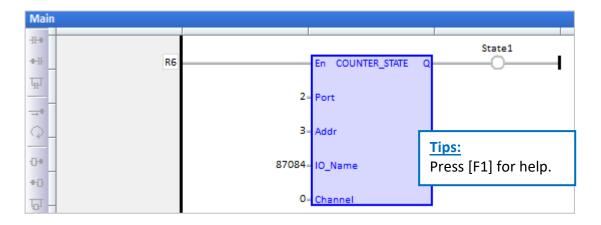
**Description:** Get the counting state of a counter or an encoder.

**<u>ST program</u>**: "TMP\_BOOL" (BOOL) can be declared in the Variables area.

TMP\_BOOL := Counter\_State (Port, Addr, IO\_Name, Channel);

#### LD program:

"State1" (BOOL): Display the status of counting.



Input Parameters	Data Type	Description
Port	DINT	COM port number (can be 1 to 37, depends on PAC).
Addr	DINT	The Net-ID address of the module can be 1 to 255.
IO_Name	DINT	The name of the Counter/Encoder module. it can be set to "87084", "87082", "7083" and "7080".
Channel	DINT	The channel number of the Counter/Encoder module. it can be set to "0", "1", and so on.

Output Parameters	Data Type	Description
Q	BOOL	TRUE: Ok. FALSE: Error.

### 8.3.5 COUNTER\_RESET

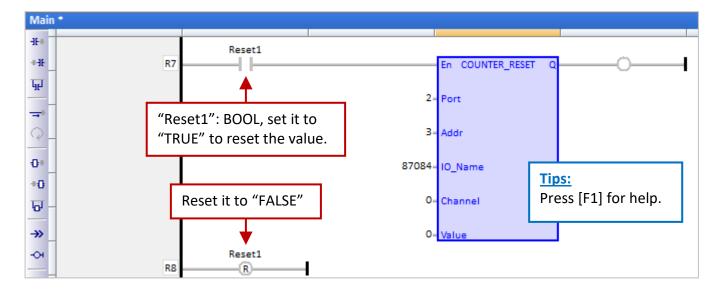
**Description:** Reset a counter or encoder to a specified value.

**<u>ST program</u>**: "Reset1" (DINT) and "TMP\_BOOL" (BOOL) can be declared in the Variables area.

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

#### LD program:

"Reset1" (BOOL): set it to "TRUE" to reset the value and then reset the "Reset1" to "FALSE".



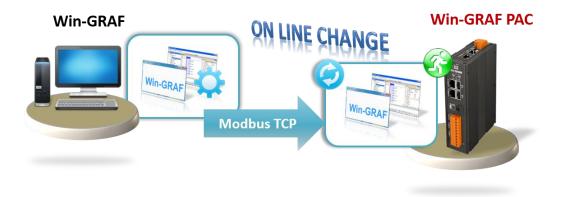
Input Parameters	Data Type	Description
Port	DINT	COM port number (can be 1 to 37, depends on PAC).
Addr	DINT	The Net-ID address of the module can be 1 to 255.
IO_Name	DINT	The name of relative Counter/Encoder module. it can be set to "87084", "87082", "7083" and "7080".
Channel	DINT	The channel number of the Counter/Encoder module. it can be set to "0", "1", and so on.
Value	"DINT", "UDINT", "DWORD", or "LINT"	Specify a new value for the Counter/Encoder.

Output Parameters	Data Type	Description
Q	BOOL	TRUE: Ok. FALSE: Error.

# Chapter 9 On-Line Change

The "On-Line Change" function allows Win-GRAF PAC to update the project with a little modification while it is running. The modified project must be the same name as the one running on the PAC.

This feature is mainly used in emergencies, in some cases, the system must be running for a 24-hour a day and cannot stop working for updating the project. Except for that situation, it is not recommended to do the online modification. The best and safest way is to stop running the project and then download the modified one to the PAC.

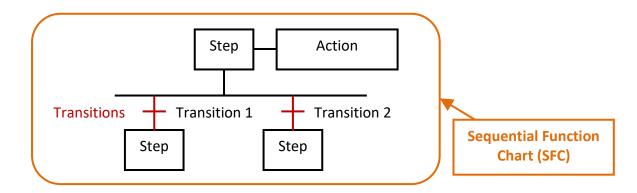


# 9.1 Limitations of "On-Line Change"

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 an SFC transition or the actions of an SFC step.



- Create, rename, or delete global and local variables.
- Create, rename, or delete global and local function block instances.

#### The following kinds of changes are not allowed:

Create, delete, or rename a program.
 (The warning message will be displayed when a program is deleted.)

Main	Workspace
A POU cannot be deleted with On Line Change active. Do-you want to disable it? 星仪) 否例	Test     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.

#### Besides, 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.

<b>Rising Pulse Det</b>	Rising Pulse Detection			
	Not Safe	Safe		
P (False to True)	SW1	SWI Inst R TRIG		
	OUT1	Inst_R_TRIG OUT1		
Decreased Pulse	e Detection			
N	SW1	SW1 Inst F TRIG		
(True to False)	OUT1	Inst_F_TRIG OUT1 CLK F_TRIG_Q		

• Loops in FBD with no declared variable linked.

K You need to explicitly insert a variable in the loop.

#### Using "On-Line Change" 9.2

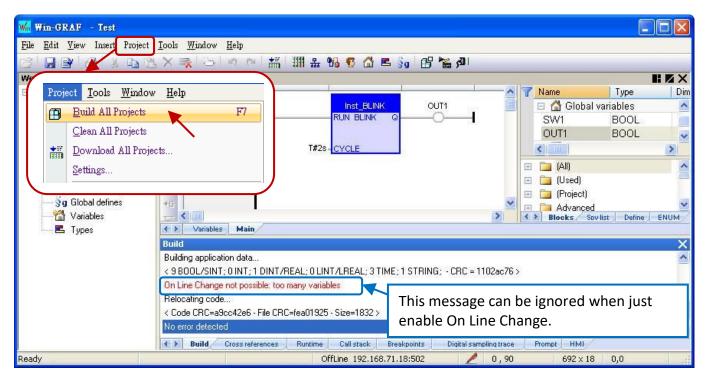
#### Enable the "On-Line Change" Function:

W

1. Click the menu command "Project - Settings..." and double-click the On-Line Change item to set it to Enabled, and then click OK.

📶 Win-GR	AF - Test							
<u>File E</u> dit (	<u>V</u> iew Insert Pr	roject <u>T</u> o	ols <u>W</u> indow	<u>H</u> elp				
23   🛃 🛢	¥ 🖪 🐰 f	Build	All Projects		F7			
Workspace	;	Clean	All Projects					
🖃 🗊 Te:	st 🛃	<u>D</u> own	load All Proje	ots				
÷… 🚞	Exception pro	<u>S</u> ettin						
	Programs							
	Project settin	ngs						
	C:\Win-GRAF\F	Projects\T	est					
	General		Name				Value	1
	Runtime		commu 🗗	nication paramete	rs		192.168.71	.18:502
	Compiler Debugging		🐐 Cycle time			0		
	Advanced		💕 Code G	eneration			Release	
	(AII)		Comple	x variables in a se	parate se	igment	No	
			😅 On Line	Change			Enabled	
			Q Version				VI - 2014/0	06/05 15:48
			Librarie	S			Edit	
			Use ext	ernal objects			Edit	
			Double click t capabilities,	o disable or enab	ie and co	nfigure On Line C	Thange	OK Cancel

2. Click the menu command Project - Build All Projects to compile programs and update the number of variables displayed in the "On Line Change" window.



RPAC-2658M User Manual, V1.0, Mar. 2021 by ICP DAS 126

#### Allocate the number of available variables:

To allow the declaration of new variables or POUs (name of the sub-program or UDFB) after the Online Change function is enabled, you have to define the amount of memory to be allocated in the PAC for each type of data.

3. Follow step 1 to open the "On Line Change" window. As the figure below, click the desired Variable type and set the number of the **Value** or **Margin(%)**, and then click the **Set** button.

Allocate	Description		
Value	Total amount (as the figure on the left). Click the BOOL variable and set it to 30. After clicking Set, the Allocated number will be 30.		
Margin	Used + Used * n%. In this example, click the BOOL variable and set it to 100%. After clicking Set, the Allocated number will be 9+9*100% = 18.		

#### Notice:

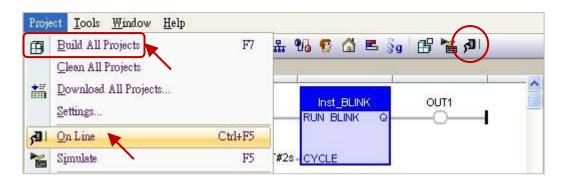
If both "Value" and "Margin" values are set, the larger number will be added, as the figure on the right. In this case, the Margins setting (100%=9) is less than the Value setting (30). After clicking Set, the Allocated number will be 9 + 30 = 39.

On Line Change	×	On Line Change	$(\times)$
- Current Status-		Current Status	
ENABLED	Enable Used/Allocated numbers	ENABLED	Enable Disable
- Number of variab	les allocated	Number of variables allocated	
BOOL/SINT val INT variables DINT/REAL val LINT/REAL va TIME variables Active timers STRING variabl STRING buffers FB instances FB instance dat Published variab Complex variales POUs	iables       5 / 30         ariables       36 / 30         ariables       5 / 30         Set these 3 types to a larger value.       3 / 30         : (character, 17 / 5000       3 / 30         a (bytes - approx.)       0 / 5000         oles       0 / 30	BOOL/SINT variables         INT variables         DINT/REAL variables         LINT/LREAL variables         TIME variables         Active timers         STRING variables         STRING variables         STRING buffers (characters)         FB instances         FB instance data (bytes - approx.)         Published variables         Complex variables         Difference         POUs	9739 5730 36730 5730 3730 1730 2730 1775000 3730 075000 0730 6775000 8730
	ed: Used / Allocated "numbers are according to the last build Value 30 2. Set Margin (%) 3.	Numbers displayed: Used / Allocated Displayed "used" numbers are according to the last build Allocate: Value 30 Margin (%) 100 2.	Set 3.

- There are some "used" numbers by default even if the project is empty.
- "STRING buffers (characters)", "FB instance data (bytes approx.)" and "Complex variables segment (bytes)" need to set a larger number (e.g., "5000").

After completing the settings, click the "X" in the upper right corner of the window to close it.

4. Click the menu command **Project** - **Build All Projects** to compile programs again. Then click "On Line" to connect to the PAC.



 After connecting to the PAC, click the "Download changes" to download the program to the PAC. Note:

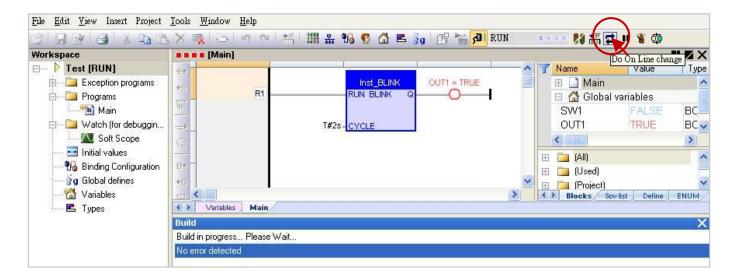
"On-Line Change" only can do the online modification for the same Win-GRAF project. If the project name is different, the user must stop running the project and download the modified one again.

[M	ain]			1100	Dows	nload changes -	TEST
-IE#			19	^	Vame Vame	Value	Тур
-HE		Inst_BLINK	OUT1 = TRUE		🖽 📄 Main		
	R1	RUN BLINK Q	<u> </u>		🗉 🚮 Global	variables	
Ŧ					SW1	FALSE	BC
		T#2s-CYCLE			OUT1	TRUE	BC
0					<	d of source the second	>
					🖽 🧰 (All)		
GH -				THE R.	🖽 🚞 (Used)		-
-0				~	🕀 🧰 (Project)		
				>		list Define	ENUM

6. Click the "Do On-Line Change" button to execute the program.

#### Note:

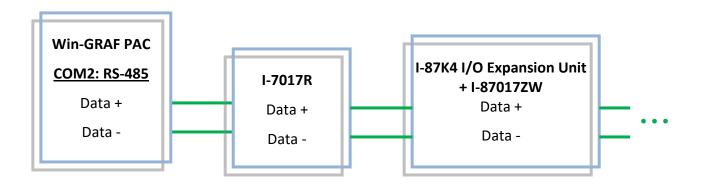
To ensure that the PAC functions properly, there are some restrictions of usage after enabling the "On-Line Change", see Section 9.1. Make sure all programs are correct before running this function.



# Chapter 10 Data/Type Conversion and Using the PAC Time

## 10.1 AI Data Conversion (i\_scale)

Refer to the following settings when connecting to the remote analog input modules, for example, connecting the I-87017ZW or I-7017R through the RS-485 port on the PAC.



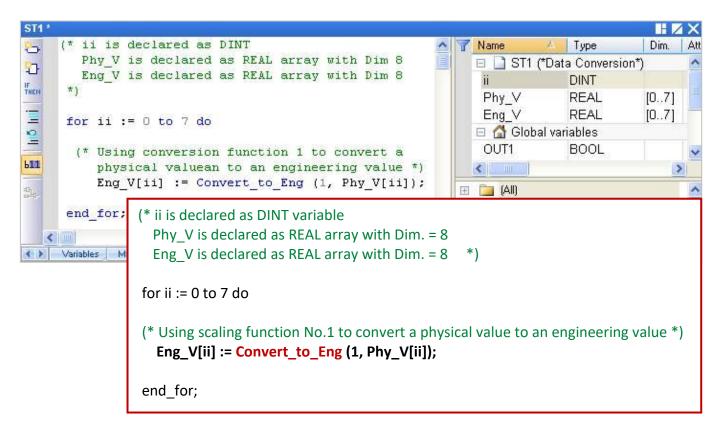
- First, add "i\_scale" in the "I/O Boards" window, and double-click on "i\_scale\_x" to open the Properties window (refer to Chapter4)
   <u>Note:</u> ONLY one "i\_scale" can be used in the Win-GRAF project.
- 2. Set values for the desired scaling function No.

For example, converting the current input (4-20 mA) into engineering values (0-10000) by using the function No. 01.

Ch**01\_**X0\_Min\_Physical\_Val: "4.0" Ch**01\_**X1\_Max\_Physical\_Val: "20.0" Ch**01\_**Y0\_Engineering\_Val\_For\_X0: "0.0" Ch**01\_**Y1\_Engineering\_Val\_For\_X1: "10000.0"

II I/O Boards	9.0: i_scale - Properties
0 1 2 3 4 5 6 7 8 9 i scale 0 i scale 0	Key = 6 Ref = 16#2A Ch00_X0_reserved = 0.0 Ch00_Y1_reserved = 0.0 Ch00_Y1_reserved = 0.0 Ch01_X0_Min_Physical_Yal = 4.0 Ch01_X0_Min_Physical_Yal = 20.0 Ch01_Y1_Engineering_Yal_For_X0 = 0.0 Ch01_Y1_Engineering_Yal = 0.0 Ch02_X0_Min_Physical_Yal = 0.0 Ch02_X1_Max_Physical_Yal = 0.0 Ch02_Y1_Engineering_Yal_For_X1 = 0.0 Ch02_Y1_Engineering_Yal_For_X1 = 0.0 Ch03_X1_Max_Physical_Yal = 0.0 Ch03_X1_Max_Physical_Yal = 0.0
- 1 i_scale_1 2 i_scale_2 10 11 12 13 14	i_scale         Setup the scaling function No. 01 to 29 for scaling I/O variables .
	< >

3. 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]).



## **10.2** AO Data Conversion (i\_scale)

Refer to the following settings when connecting to the remote analog output modules, for example, connecting the I-7024 through the RS-485 port on the PAC.

1. Add "i\_scale" in the "I/O Boards" window, and double-click on "i\_scale\_x" to open the Properties window (refer to Chapter4)

**<u>Note</u>:** ONLY one "i\_scale" can be used in the Win-GRAF project.

I/O Boards	
0	Close
1	0.000
2	 Select
3	
4	Delete
5	
6	 Rename
7	
8	 Properties
9 i_scale	
- 0 i_scale_0	Virtual/Real
- <u>1 i_scale_1</u>	
2 i_scale_2	Move Up
10	
11	Move Do <u>w</u> n
12	 22, 142
13	Help
14	

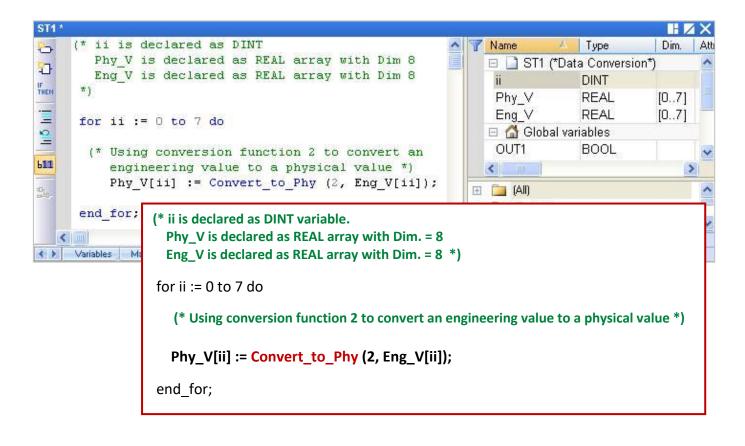
2. Set values for the desired scaling function No.

For example, converting the engineering values (0-20000) into the voltage output (0-10V) by using the function No. 02.

Ch**02**\_X0\_Min\_Physical\_Val: "0.0" Ch**02**\_X1\_Max\_Physical\_Val: "10.0" Ch**02**\_Y0\_Engineering\_Val\_For\_X0: "0.0" Ch**02**\_Y1\_Engineering\_Val\_For\_X1: "20000.0"

	9.0: i_scale - Properties	×
	Key = 6	~
	<b>Ref</b> = 16#2A	
	Ch00_X0_reserved = 0.0	
	<b>Ch00_X1_reserved</b> = 0.0	
	Ch00_Y0_reserved = 0.0	
	<b>Ch00_Y1_reserved</b> = 0.0	200
	Ch01_X0_Min_Physical_Val = 4.0	
	Ch01_X1_Max_Physical_Val = 20.0	
	Ch01 Y0 Engineering Yal For X0 = 0.0 Ch01 Y1 Engineering Yal For X1 = 10000.0	
	Ch01_Y1_Engineering_Ya1_For_X1 = 10000.0	
	Ch02 X0 Min Physical Val=0.0	
Ch02	Ch02_X1_Max_Physical_Val = 10.0	
CHOZ	Ch02_Y0_Engineering_Yal_For_X0 = 0.0	
	Ch02_Y1_Engineering_Yal_For_X1 = 20000.0	
	Chu3_AU_Min_Physical_Val=0.0	(2029)
	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]).



# **10.3** Type Conversion Functions (ANY\_TO\_xxx)

When making comparisons (e.g., > , < , = , <= , >= , <> ) and performing calculations (e.g., +, - , \* , / ) with different types of data or variables in the program, refer to the table below to convert them into the same data type to perform properly.

Type Conversion Functions	Descriptions
ANY_TO_BOOL	Convert to Boolean
ANY_TO_SINT	Convert to Short Integer (8-bit)
ANY_TO_INT	Convert to Integer (16-bit)
ANY_TO_DINT	Convert to Long Integer (32-bit – Default)
ANY_TO_LINT	Convert to Long Integer (64-bit)
ANY_TO_TIME	Convert to Timer
ANY_TO_REAL	Convert to Real
ANY_TO_LREAL	Convert to Double
ANY_TO_STRING	Convert to String
NUM_TO_STRING	Convert Number to String. It allows specifying the number of decimal places to be converted
АТОН	Convert Hexadecimal String to Integer
НТОА	Convert Integer to Hexadecimal String

For example, the ST program is used to convert the type DINT to REAL and then perform the calculation.

REAL\_Val\_1 := ANY\_TO\_REAL (DINT\_Val\_1) \* 3.5 + 4.8 ;

Also, open the "HTML Help" from the menu bar and enter the keyword to search for the relevant instructions.

		💕 HTML Help		
Help	Topics	Hide Locate Back Forward Refresh	Home Prin	
٩	Search T <u>u</u> torials	Contents Search Favorites		
?	<u>A</u> bout	Type in the word(s) to search for:		
	Language	Type Conversion Functions		~ •
		List Topics	D	isplay
		Select topic:	Found: 14	
		Title	Location	Rank
		Type conversion functions	IEC 61131-3	1
		Programming languages - Reference guide	IEC 61131-3	2
			IEC 61131-3 IEC 61131-3	2 3
		Programming languages - Reference guide		-

# 10.4 BCD Conversion (BIN \_TO\_BCD, BCD\_TO\_BIN)

The BCD code uses 4 bits to represent the decimal numbers 0 to 9. Suppose a decimal value is "132" which can be converted to a BCD code of "0001 0011 0010" or a binary value of 10000100 (i.e.,  $2^7 + 2^2 = 128 + 4 = 132$ ).

Note: The BCD code can only be used to represent numbers from 0 to 9 and can not use the value 1010, 1011, 1100, 1101, 1110, and 1111, which will return "0".

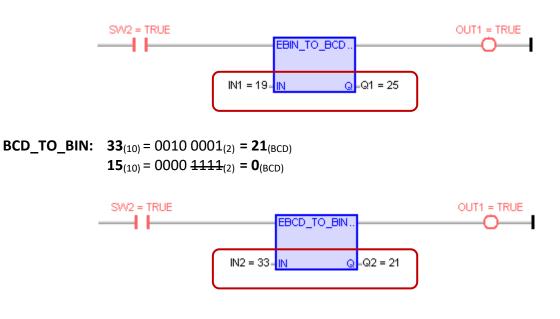
Decimal	BCD				Description		
Decimal	<b>2</b> <sup>3</sup>	<b>2</b> <sup>2</sup>	<b>2</b> <sup>1</sup>	<b>2</b> <sup>0</sup>	Description		
0	0	0	0	0	0		
1	0	0	0	1	2 <sup>0</sup> = 1		
2	0	0	1	0	2 <sup>1</sup> = 2		
3	0	0	1	1	$2^1 + 2^0 = 3$		
4	0	1	0	0	$2^2 = 4$		
5	0	1	0	1	$2^2 + 2^0 = 5$		
6	0	1	1	0	$2^2 + 2^1 = 6$		
7	0	1	1	1	$2^2 + 2^1 + 2^0 = 7$		
8	1	0	0	0	2 <sup>3</sup> = 8		
9	1	0	0	1	$2^3 + 2^0 = 9$		

The table lists functions that can be used to do BCD (Binary Coded Decimal) conversion.

Type Conversion Function	Description		
BIN_TO_BCD	Convert Binary to BCD value		
BCD_TO_BIN	Convert BCD value to Binary		

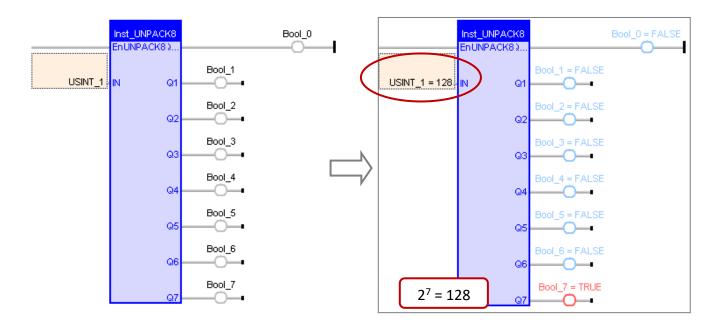
Note: IN1, IN2, Q1, and Q2 are declared as DINT. If the IN is less than or equal to 0, the Q will return 0.

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

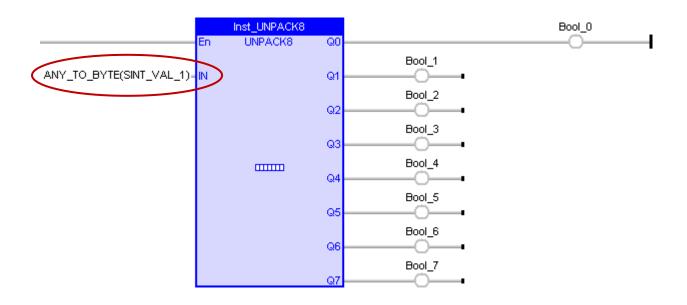


#### **Unpack Integer to Boolean:**

The "UNPACK8" function can be used to unpack one BYTE (or USINT, range: 0 to 255) to 8 Booleans.



To unpack **one SINT** to **8 Booleans**, first, add the "**ANY\_TO\_BYTE()**" with the ST syntax to convert the SINT into BYTE type, as follows:



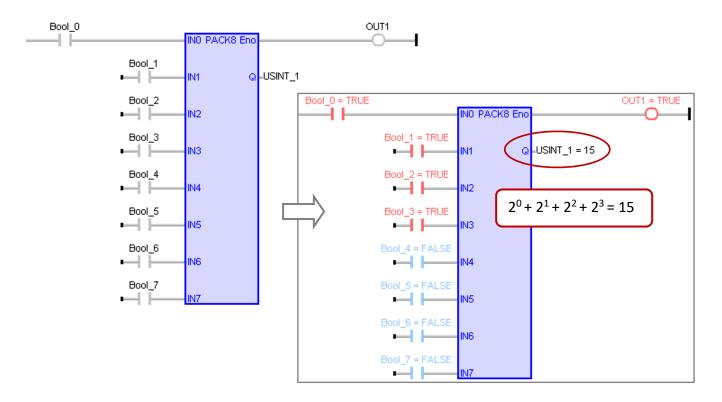
#### Pack Boolean into Integer:

The "PACK8" function can be used to pack 8 Booleans into one BYTE(or USINT, range: 0 to 255).

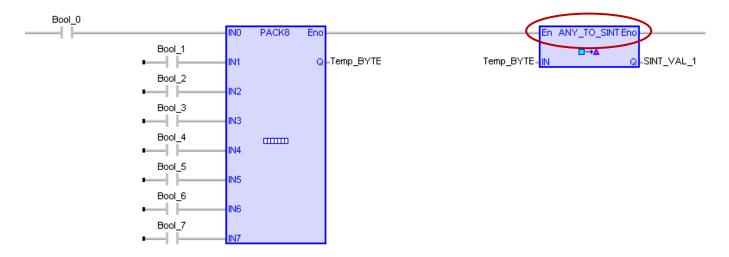
#### ST Program:

USINT\_1 := PACK8 (Bool\_0, I Bool\_1, Bool\_2, Bool\_3, Bool\_4, Bool\_5, Bool\_6, Bool\_7);

#### LD Program:



To pack **8** Booleans into one SINT, first specify a "BYTE" variable to the output(Q) for storing the value temporarily and add an "ANY\_TO\_SINT" function to convert the BYTE to SINT, as follows:



**Note:** If the compilation fails, click the menu command **Project - Settings** and click "Runtime" to see if the setting of "**Complex variables in a separate segment**" is "**Yes**".

Project settings		X
C:\Win-GRAF\Projects	i\Test	
General	Name	Value
Runtime Compiler Debugging Advanced (All)	💷 Runtime system	Little endian
	🖀 Cycle time	0
	💕 Code Generation	Release
	🔀 Exchange IOs while stepping	Yes
	Support variables locking	No
	[i]. Complex variables in a separate segment	Yes

# 10.6 Pack/Unpack BYTE, WORD, DWORD

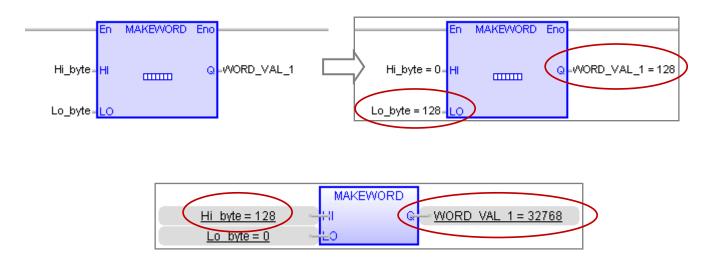
#### Pack Two 8-bit Data into One 16-bit Data

The "MAKEWORD" function can be used to pack 2 Bytes (or USINT) into one WORD (or UINT).

#### ST Program:

WORD\_VAL\_1 := MAKEWORD (Hi\_byte, Lo\_byte);

LD/FBD Program:



To pack **2 SINT** into **one INT**, first, add the "ANY\_TO\_BYTE()" with the ST syntax to convert SINT into BYTE and add the "ANY\_TO\_INT" function to convert the packed WORD into INT type.



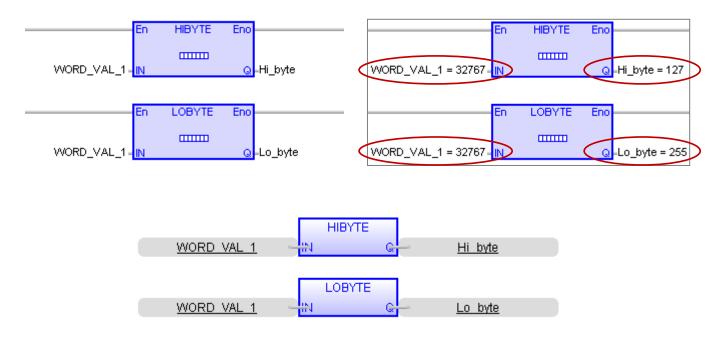
#### Unpack One 16-bit Data to Two 8-bit Data

"HIBYTE" and "LOBYTE" functions can be used to unpack one WORD (or UINT) to 2 BYTEs (or USINT).

ST Program:

Hi\_byte := HIBYTE (WORD\_VAL\_1); Lo\_byte := LOBYTE (WORD\_VAL\_1);

#### LD/FBD Program:



To unpack **one INT** to **2 SINTs**, first, add the "ANY\_TO\_WORD()" with the ST syntax to convert INT into WORD and add the "ANY\_TO\_SINT" function to convert the unpacked BYTE into SINT type.



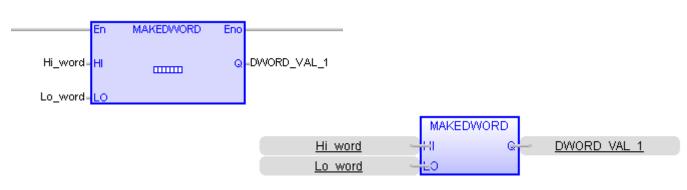
#### Pack Two 16-bit Data into One 32-bit Data

The "MAKEDWORD" function can be used to pack 2 WORDs (or UINT) into one DWORD (or UDINT).

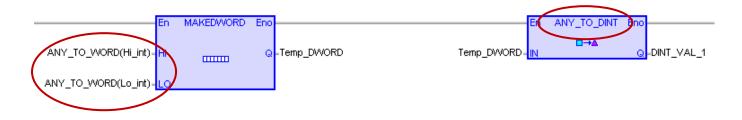
#### ST Program:

DWORD\_VAL\_1 := MAKEDWORD (Hi\_word, Lo\_word);

#### LD/FBD Program:



To pack **2 INTs** into **one DINT**, first, add the "ANY\_TO\_WORD()" with the ST syntax to convert INT to WORD and add the "ANY\_TO\_DINT" function to convert the unpacked DWORD into DINT type.



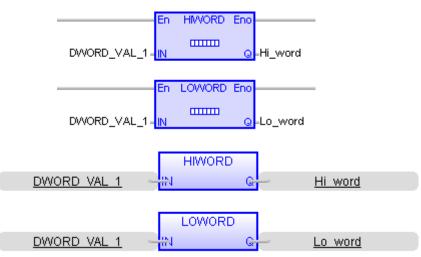
#### Unpack One 32-bit Data to Two 16-bit Data

"HIWORD" and "LOWORD" functions can be used to unpack **one DWORD** (or UDINT) to **2 WORDs** (or UINT).

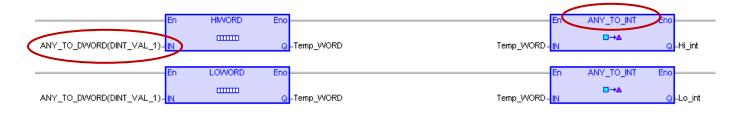
ST Program:

Hi\_word := HIWORD (DWORD\_VAL\_1); Lo\_word := LOWORD (DWORD\_VAL\_1);

LD/FBD Program:



To unpack **one DINT** to **2 INTs**, first, add the "ANY\_TO\_DWORD()" with the ST syntax to convert DINT to DWORD and add the "ANY\_TO\_INT" function to convert the unpacked WORD into INT type.



# 10.7 Unpack Variable to Byte Array or Pack Byte Array into Variable

The "**SerializeOut**" function can unpack a Win-GRAF Variable value to a Byte Array (or USINT Array); The "**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 to at least "8".

2. The STRING variable can not be used in the "Serialize" function.

Also, open the "HTML Help" from the menu bar and enter the keyword to search for the relevant instructions.

		😵 н	IML Help						
<u>H</u> elp		-		$\langle \neg$	⇒	÷.		4	ð-
3	Topics	Hide	Locate	Back	Forwar	d Refresh	Home	Print	Options
Q	Search		-						
	T <u>u</u> torials	Conte	ents Se	arch	Favorites	s			
?	<u>A</u> bout	Type in	the word(	s) to sear	ch for:				
	Language	Seriali	ze						~ ►
						List Topic	s	C	)isplay
		Select	topic:				Found	d: 2	
		Title				Locatio	n		Rank
		Serial	izeln			IEC 61	131-3		1
		Serial	izeOut			IEC 61	131-3		2

If the returned value of **SerializeOut()** and **SerializeIn()** is "0", which means the storage location is incorrect or not enough space for the Array data.

(\* Declare TMP\_DINT as a DINT,

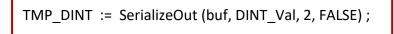
buf as a BYTE Array, Dim. = 10, DINT\_Val as a DINT, Word\_Val as a WORD, REAL\_Val as a REAL \*)

#### Note:

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 location 2 of the BYTE Array in Little-Endian order. \*)



**Note:** The last parameter "FALSE" means that use Little-Endian ordering (i.e., the low byte is stored at the start 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 Big-Endian order. \*)

TMP\_DINT := SerializeOut(buf, Word\_Val, 0, TRUE);

**Note:** The last parameter "TRUE " means that use Big-Endian ordering (i.e., the high byte is stored at the start address).

#### Example 3

(\* To pack the buf[0], buf[1], buf[2], and buf[3] in the Byte Array into one REAL\_Val in Little-Endian order \*)

TMP\_DINT := SerializeIn(buf, REAL\_Val, 0, FALSE) ;

**Note:** The last parameter "FALSE" means that use Little-Endian ordering (i.e., the low byte is stored at the start address).

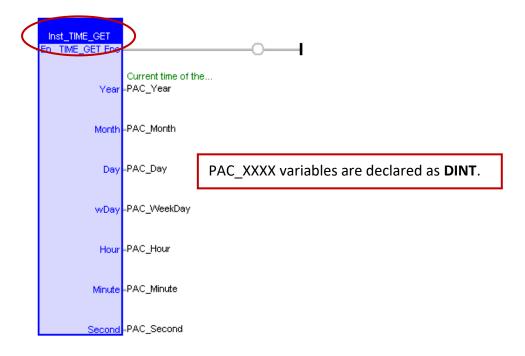
Example 4

(\* To map one DINT\_Val to one REAL\_Val in "Little-Endian" order. \*)

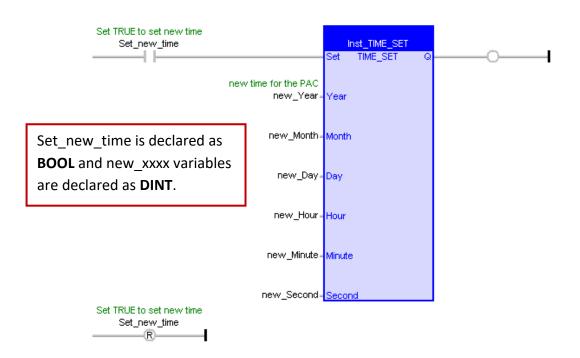
TMP\_DINT := SerializeOut(buf, DINT\_Val, 0, FALSE) ; TMP\_DINT := SerializeIn(buf, REAL\_Val, 0, FALSE) ;

# 10.8 Get/Set the PAC Time

The "**TIME\_GET**" function block can be used to get the current time of a Win-GRAF PAC. (Refer to Section 2.2.1)



The "**TIME\_SET**" function block can be used to set the current time of a Win-GRAF PAC. 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.



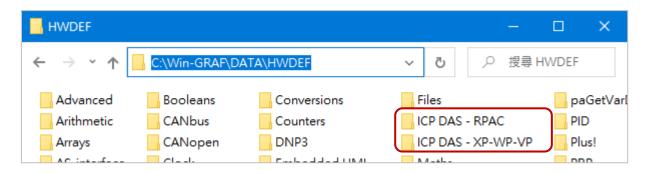
# Chapter 11 Commonly Used Tools and Useful Tips

### 11.1 Upgrade Win-GRAF Libraries

Download the latest version of Win-GRAF libraries (e.g., "win-graf-lib-x.xx.zip") on the website: https://www.icpdas.com/en/download/show.php?num=695

ICP DAS Win-GRAF Libraries, including functions, function blocks, and so on, are stored in the "ICP DAS - XP-WP-VP" and "ICP DAS – RPAC" folders under the "C:\Win-GRAF\DATA\HWDEF" directory.

- 1. Close all Win-GRAF Workbench windows before upgrading. It is recommended to back up the old version of libraries to the other folder (e.g., D:\temp\xxx.zip) before deleting the libraries in the current folder.
- 2. Copy the new version of "ICP DAS XP-WP-VP" and "ICP DAS RPAC" folders to the directory "C:\Win-GRAF\DATA\HWDEF" and execute the Win-GRAF Workbench.



#### Note:

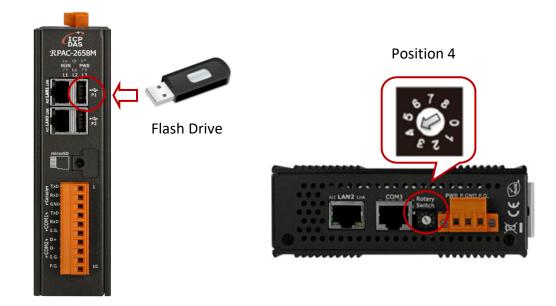
To know the version number of the Win-GRAF library, open the "ICP DAS - XP-WP-VP" library in the "Library Manager" (refer to Section 1.2.2) and click "A\_Version" on the "I/Os" page and then click "Description" to check the version number (e.g., "1.07").

Fil	e Tools Wizard He	lp		
	Open Library 😿	Open Lib	orary	X
	Save Library			
	Use in SAMA		- XP-WP-VP	
•	Exit		6 - RPAC 6 - XP-WP-VP 🝗	Cancel
Library Manager - ICP DAS - XP-WP-VP		Maths		2.
File Tools Wizard Help				
Function and FBs 1/Os Profiles AS-i Types				
□▲_¥ersion (*) iberary velvion: 1.07, Jun.18,2015. Id		ect it . *)	New	
<b>DCON</b> (* Enable one serial port to connect DCON I/O: <b>DIP_switch</b> (* 8-Ch D/I from the Dip switch of PAC)	*)	3.	Rename	
Parameters Description			Delete	
The version number of the Libray "ICP 1	DAS - XP-WP-VP" is 1.0	7.	Store	
This "A_Version" is only for identifica	ation .		Reset Changes	
The PAC's Win-GRAF driver can not rec DO NOT connect this "A_Version" in the	-	~		
<		>		

# 11.2 Upgrade Win-GRAF Driver

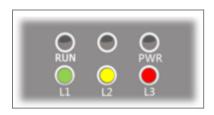
The user can download the latest <u>driver</u> on the website and follow the steps below to update the firmware. https://www.icpdas.com/en/download/index.php?model=RPAC-2658M

Ste	p <u>1</u> :	Format TOSHIBA (H:) X
	Prepare a flash drive.	Capacity: 10.2 GB ~
		File system FAT32 (Default) VITES
b.	Format your flash drive to <b>FAT32</b> format.	FAT32 (Default) EXFAT
		Restore <u>d</u> evice defaults
c.	Put patch files on your flash drive. The file name is <b>patch_vxx.xx.zip.</b>	Volume label TOSHIBA
	patch_v0.11.zip	Format <u>o</u> ptions <u>Q</u> uick Format
_		
Plu	<u>p 2</u> : g your flash drive into the USB port of RPAC-2658M and the rotary switch to position <b>4</b> .	<u>S</u> tart <u>C</u> lose



#### <u>Step 3</u>:

Turn the power on and the firmware update process will be executed automatically. When the firmware update is completed, the **L1**, **L2**, and **L3** LED will be **ON**.



#### <u>Step 4</u>:

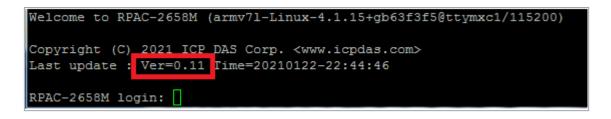
Power off and unplug your flash drive.

#### Step5:

Set the rotary switch to position **0** and power on again. The new firmware version will be shown on the console port or the web page. It means that the firmware update is successful.

Console port:

Refer to Section 13.2 to connect to the PAC by using the SSH tool.



#### Web page:

Enter the IP address of the PAC.

← → C (A	不安全 192.168.79.11	\$	0	* (	<b>)</b> :
	P-DAS_			1	
<u>Login</u> <u>User Account</u> <u>WG_Mgmt</u>	Information OS Version :	1 4.1.15+gb63f3f5 #32 SMP PREEMPT Thu Sep 24 01:54:50 PDT 2020			
Information	APP Version :	0.11			- 1
Reboot	WinGRAF Version :	V0.24 20210120			- 1
Reboot	WinGRAF Info :	Project: demo_file1, Extra_port: None, Elapsed time : 0. 0:15: 0			- 1
	Eth1 IP :	192.168.79.11			- 1
	Eth1 Mask :	255.255.0.0			- 1
	Eth1 Mac :	00:0D:E0:C0:00:28			- 1
	Eth2 IP :	172.16.255.1			- 1
	Eth2 Mask :	255.255.0.0			- 1
	Eth2 Mac :	00:0D:E0:C0:00:29			- 1
	Eth3 IP :	10.0.255.1			- 1
	Eth3 Mask :	255.255.0.0			- 1
	Eth3 Mac :	00:0D:E0:C0:00:2A			-
http://www.icp	das.com		~~~	ď	AS

## 11.3 Spy List

The Spy List is a monitoring function that enables you to watch Win-GRAF variables at runtime. Hundreds of variables may be declared in an application. Workbench allows users to add the specified variable to the Spy List and monitor the current value and status of the variable at runtime.

In online mode, you can write and force variable values in a Spy List to actively influence the PLC application behavior.

Steps:

- 1. Right-click on the project name (e.g., "Demo01") and select Insert New Item.
- 2. Click the Watch category and select the Spy List item, then click Next to the next step.
- 3. Enter a name (e.g., "NewSpy1") and press **OK**.

Works	pace Dem-01	<u>\</u>					
⊡ <b>€</b>	Dem	Set as Startup Project					
-	- 🗀 F	Communication Parame	ters				
	📑 ह्वा	On Line					
ė.	🖌 🕍	Simulate					
	<u></u>	Monitor	🐝 Inser	t New Item			
	- 🖻 ! 🍋	I <u>n</u> sert New Folder	Categor	ies:	Available Items:		
		Inse <u>r</u> t New Program	(All) Program		Spy list		
-	୍ୟା 🔛	Insert HMI Device	Watch Resour		Recipe		
	ğg (	Shortcuts Insert New Item		ded HMI	Soft Scope	2.	
	(1.)	Insert New Tient	Others			$\bigcirc$	
	$\bigcirc$						
1	Propertie	8 \$			X		
	a series						
	Name	NewSpy1	3.				
	Descriptio	n	$\mathcal{I}$			Next	Cancel
			Ok	Cancel		- HUNC	

4. Double-click "NewSpy1" on the left side to open the window and drag the selected variable you want to observe into this window.

Workspace		NewSpy1.spl *							>
🖃 🔐 🗿 🖂 🖂 🗐	同	Name	Value	Description	T	Name	Туре	Dim.	1
🗄 🔤 Exception programs		PAC_Year		Current time of the PAC		🗉 🚮 Global varial	oles		1
📩 🔤 Programs	8	PAC_Month				PAC_Year	DINT		
PAC_Time (*Get / Set P	∎+	PAC_Day				PAC Month	DINT		F
🛄 LD1 (*Blinking*)		PAC_WeekDay	×			PAC Day	DINT		ŀ
🖨 🛅 Watch (for debugging)		PAC_Hour				PAC WeekDay	DINT		L
Soft Scope		PAC_Minute	1 N N			PACHour	DINT		L
📑 NewStructure		PAC_Second		<u>`</u>		PAC Minute	DINT		L
🎫 Initial values		🖽 INT_Val			- 1	PAC Second	DINT		L
		⊞ REAL_Val				INT Val	INT	[02]	L
		1270				REAL Val	REAL	(03)	
Binding Configuration	2	<		>		<		>	1
🚽 🖇 🚽 🚽 🚽	5)	LD1 PAC Time	Variables Ne	w Spy 1					

5. If the Win-GRAF workbench is connected to the PAC then the current variable values are shown.

rkspace		NewSpv1.spl							
Demo01 [RUN]		Name	Value	Description	🍸 Name	Value	Туре	Dim.	Attrib
Exception programs		PAC_Year	2014	Current time of the PAC	🗉 🚮 Global varial	a state of the second second			
Programs	E.	PAC_Month	6		PAC Year	2014	DINT		
PAC_Time (*Get / Set P	目↓	PAC_Day	13		PAC Month	6	DINT		
LD1 (*Blinking*)		PAC_WeekDay	5		PAC Day	13	DINT		
- Watch (for debugging)		PAC_Hour	5		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		
🔜 NewRecipe1		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	10000 57	
🚽 🕺 🚽 Global defines		⊟ REAL_Val			new Month	0	DINT		
		REAL_Val[0]	5.8		new_Day	0	DINT		
E Types		REAL_Val[1]	9.6		new_Hour	0	DINT		
		REAL_Val[2]	3.2		new_Minute	0	DINT		12
		REAL Val[3]	2.5		now Cocond	n	DINT		>

### Back up a Win-GRAF Project:

1. Right-click the project name (e.g., "Demo01") and select "Save Project" and then "To Zip".

Workspace						
Demo01		<u>S</u> et as Startup Project				
🖻 📮 Programs		Communication Parameter	s			
PAC_Tir	الكر	<u>O</u> n Line				
🖳 🛅 LD1 (*E 🔤 Main	*	S <u>i</u> mulate				
		<u>M</u> onitor				
	<u>-</u>	I <u>n</u> sert New Folder				
	•	Inse <u>r</u> t New Program				
		Inser <u>t</u> HMI Device			Also, refer to Win-GRA	.F
		S <u>h</u> ortcuts	•	Ш.	FAQ-003 for the To Dis	k usage.
		Insert Ne <u>w</u> Item				)
		S <u>a</u> ve Project			<u>T</u> o Disk	
	7	Duplicate Project		٩	T <u>o</u> Zip	
	4	Print Project		<b>8</b>	To <u>Z</u> ip and Mail	
	Q	Project Description			To T <u>a</u> rget	

2. Specify the location (e.g., D:\Win-GRAF\_demo\_backup) to save the file and enter the project name (e.g., "Demo01\_0106"), and then click "Save" to complete the backup.

Save in: Win-GRAF_demo_backup   Wame Date modified   No items match your search.   Desktop   Libraries
Name Date modified     No items match your search.     Desktop
Libraries
This PC
Network     K     Demo01_0106.zip     Save       Save as type:     Zip file (*.zip)     Cancel
Save as type: Zip file (*.zip) Cancel Help

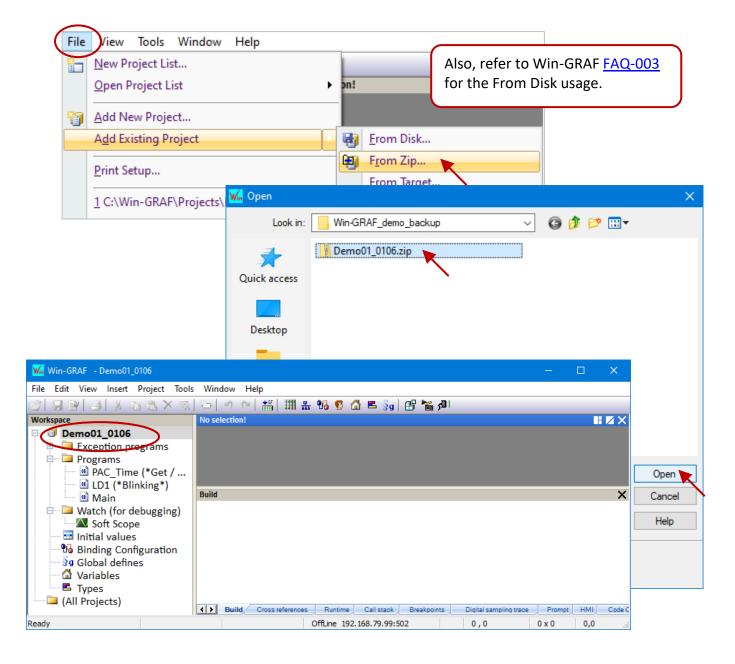
RPAC-2658M User Manual, V1.0, Mar. 2021 by ICP DAS 147

#### **Restore a Win-GRAF Project:**

**Note:** Before restoring, click the menu command **File - Close Project List** to close all projects.

Win	Win-G	Vin-GRAF - Demo01									
File	dit	dit View Insert Project Tools Window Help									
	<u>N</u> ew F	New Project List									
	<u>O</u> pen	Open Project List									
	<u>S</u> ave F	Save Project List									
1	<u>C</u> lose	Project	List 📉								
	File	File dit New F Open Save F	File dit View <u>N</u> ew Project L <u>O</u> pen Project <u>S</u> ave Project L	New Project List Open Project List Save Project List	File dit View Insert Project New Project List Open Project List Save Project List	File Idit View Insert Project Tools           Mew Project List           Open Project List           Save Project List	File Idit View Insert Project Tools Window           New Project List           Open Project List           Save Project List	File Idit View Insert Project Tools Window Help New Project List Open Project List Save Project List			

 Click the menu command "File - Add Existing Project - From Zip", select the project name (e.g., "Demo01\_0106"), and click "Open" to complete the restoring.



## 11.5 Soft Reboot on the PAC

In some cases, the user may want to reboot the PAC by using the software way. Win-GRAF provides the "PAC\_Reboot" function for users to restart the PAC automatically.

### Note:

DO NOT call this Function in every PAC Cycle, or the PAC will reboot all the time.

### Safety Coding:

(* "reset_PAC" is declared as BOOL with an "TMP_BOOL" is declared as BOOL *) (* ONLY when "reset_PAC" is set to "TRUE",	
<pre>if reset_PAC then     reset_PAC := FALSE ;     TMP_BOOL := PAC_Reboot(); end_if;</pre>	reset_PAC Q PAC REBOOT reset_PAC R

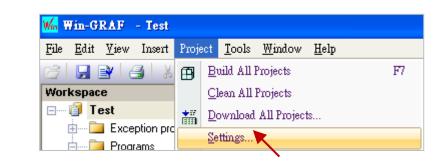
### **Dangerous Coding:**

```
(* "TMP_BOOL" is declared as BOOL *)
(* Dangerous! This code below will cause the PAC to restart repeatedly. *)
TMP_BOOL := PAC_Reboot();
```

**If the PAC keeps rebooting**, simply Set the rotary switch of the PAC to "**1**" (i.e., Safe mode) and reboot. After that, download the project again and Set the rotary switch of the PAC to "**0**" (i.e., Normal mode).

# 11.6 Using ST Syntax in an LD and FBD Program

Win-GRAF Workbench allows users to use a simple ST syntax in the LD or FBD program to facilitate programming. First, click the menu command **Project - Settings** and click **Runtime** and then set the "Complex variables in a separate segment" to "Yes".

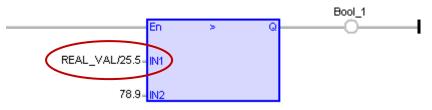


Project settings		X
C:\Win-GRAF\Projects\	Test	
General	Name	Value
Runtime Compiler Debugging	💷 Runtime system	Little endian
	🖀 Cycle time	0
Advanced	🔡 Code Generation	Release
(All)	🔀 Exchange IOs while stepping	Yes
	😽 Support variables locking	No
	[i]. Complex variables in a separate segment	Yes

### Example:

### LD Program:

Using division (REAL\_VAL/25.5).



### FBD Program:

Call a function "ANY\_TO\_BYTE()" to convert the type from "SINT" to "BYTE".

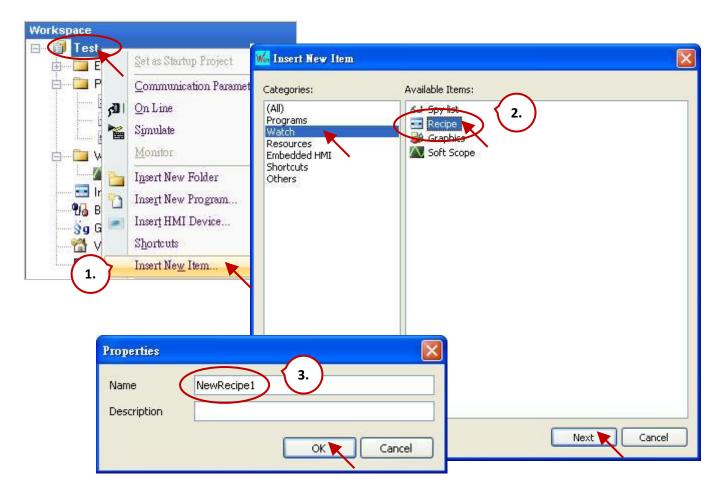


# 11.7 Apply a Recipe on the PAC

For some applications, it needs to change the configuration of the product in the Recipe to meet the requirement of the different manufacturing processes. The user can assign the required variables and values for each product. Next time when producing the other product, the user can choose the desired product configuration to apply to the PAC.

#### Follow the steps below:

1. Create a Recipe. Right-click the project name and select Insert New Item. Click the Watch/ Recipe item and click Next, then enter a name (e.g., "NewRecipe1") and click OK.



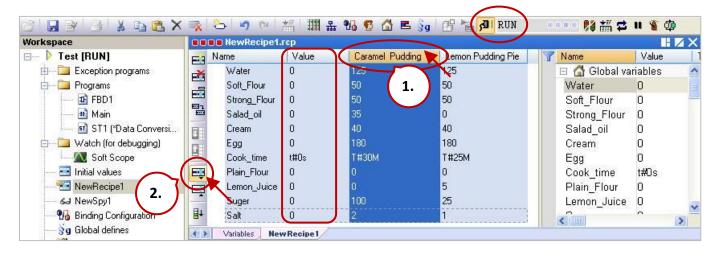
2. Assign the variables. Double-click NewRecipe1 in the Workspace to open the window and drag all needed variables into it.

Workspace	NewRecipe1.rc	p *					X
🖃 📲 Test	E-3 Name	Value	7	Name	Туре	Dim.	A
🗄 🛁 Exception programs	2000 BRIDE 100 BRIDE			🗉 🚮 Global va	ariables	_	~
📩 🛅 Programs	Water Soft_Flo Strong_l	ur		Water	DINT		
🗗 FBD1	Strong_l	Flour		Soft_Flour	DINT		1
📑 Main	Salad_o	il 📉		Strong_Flour	DINT		
🛄 🖬 ST1 (*Data Conversi	Cream			Salad_oil	DINT		
📩 🛅 Watch (for debugging)	Egg			Cream	DINT		
Soft Scope	Cook_tir	ne	-	Egg	DINT		
🔜 Initial values	Plain_Fl	our	 	Cook_time	TIME		
	Lemon_	Juice		Plain_Flour	DINT		
Sel NewSpy1	Suger			Lemon_Juice	DINT		
	<b>∎</b> +			Suger	DINT	-	Y
🚽 🖇 🚽 Global defines				<		>	
- 🚰 Variables	♦ Variables	New Recipe 1	().				

3. **Configure the Recipe.** Click the **Insert Column** button to add a column and enter a name. Double-click the blank cell to enter a value and press the Enter key.

Name	Value	Caramel Pudding	Lemon Pudding Pie	Name	Туре	
Water		125	125	🗉 🚮 Global va	riables	^
Soft_Flour		50	<sup>50</sup> Enter a name.	Water	DINT	
Strong_Flou	JI.	50	50	Soft_Flour	DINT	
Salad_oil		35	0	Strong_Flour	DINT	-
Cream		40	40	Salad_oil	DINT	
Egg		180	180	Cream	DINT	
Egg Cook_time		T#30M	T#25M	Egg	DINT	
Plain_Flour		0	0	Cook_time	TIME	
	e	0	5	Plain_Flour	DINT	
Lemon_Juid		100	25	Lemon Juice	DINT	
Salt		2	1	Suger	DINT	
		Double	-click the cell to enter	Salt	DINT	
Variables	New Recipe 1		and press Enter.	<	- A	>

4. **Apply the Recipe.** Download the project to the PAC and you can see all values are "0". Choose the desired product name and click the "Send Recipe" button to apply the configurations to the PAC.



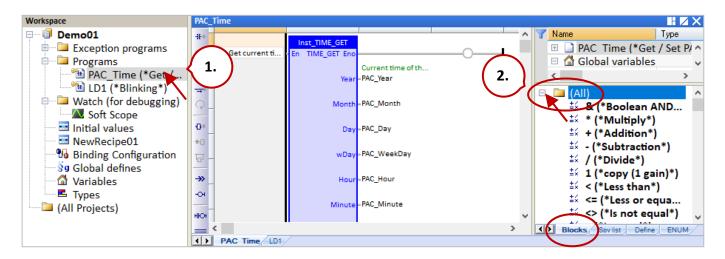
	NewRecipe1.rcp							
E E	Name	Value	Caramel Pudding	Lemon Pudding Pie				
<b>H</b>	Water	125	125	125				
	Soft_Flour	50	50	50				
	Strong_Flour	50	50	50				
E	Salad_oil	35	35	0				
	Cream	40	40	40				
	Egg	180	180	180				
	Cook_time	t#30m	T#30M	T#25M				
E S	Plain_Flour	0	0	0				
<b>E</b>	Lemon_Juice	0	0	5				
	Suger	100	100	25				
∎+	Salt	2	2	1				
<b>&lt; &gt;</b>	Variables New Recipe 1							

Note: Also, refer to Chapter 6 to use Retain Variables to ensure that all values will not disappear due to an unexpected power cut.

## 11.8 Functions and Function Blocks Supported by Win-GRAF PACs

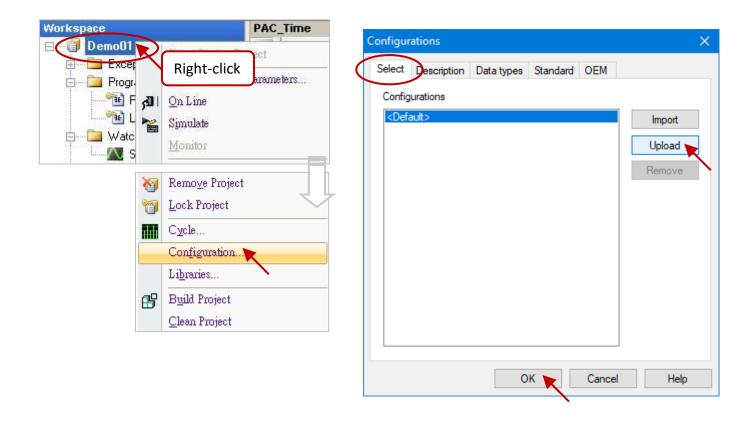
There are many functions and function blocks listed in the Win-GRAF **Block** pane, but some of them are not supported by Win-GRAF PACs. The section describes how to tell which ones are supported or not.

For easy to view all functions and FBs, expand the **All** folder in the **Blocks** pane in any program.



### Follow the steps below to upload the configuration file from PAC:

- 1. Make sure the PAC is powered on and on the same network segment as the Win-GRAF/PC.
- 2. Right-click the Win-GRAF project name and select "Configuration", and then click the "Upload" button on the "Select" page to open the settings window.



3. Set or choose an IP address of the PAC and click the "OK" button to start uploading. By default, the configuration file will be stored in the C:\Win-GRAF\DATA\CONFIG folder. Simply enter a file name (e.g., "vp\_fbs.cfg") and click the "Save" button.

Communication Settings	×
T5 Runtime 192.168.79.101:502 192.168.71.17:502 192.168.79.99:502 Choose	Cal Configurations × Bro Uploading runtime configuration H Please wait.
W Save As	×
← → × ↑ 📙 C:\Win-GRAF\DATA	、CONFIG ~ ひ Search CONFIG
Organize 🔻 New folder	
AUTOMATION ^ Nar	me Date modified
Bitmap CONFIG	No items match your search.
File name(vp_fbs.cfg	~
Save as type: Configuration files (	(*.cfg) ~
∧ Hide Folders	Save Cancel

4. The file name will be displayed in the Configurations window, click OK to exit. Then, check the **Blocks** pane again, the functions and FBs marked in red are not supported by Win-GRAF PAC.

figurations	×	±∞ ANY_TO_STRING ±∞ ANY_TO_TIME (*.
Select Description Data types Standard	OEM	=∝ ANT_TO_TIME (* ≛∞ ANY_TO_UDINT (
Carfin milian		± ANY_TO_UINT (*
Configurations		≛≨ ANY_TO_ULINT (
vp fbs	Import	≛ ANY_TO_USINT (
	Upload	APPLYRECIPECOL
		C ARCOUNT (*Coun.
	Remove	C ARCREATE (*Crea
		C ARFILL (*Fill an ar.
		다 ARFIND (*Search 다 ARMAX (*Get max.
		D ARMAX ("Get max.
		T ARRAYTOSTRING
		ARRAYTOSTRING
		ARRAYTOSTRINGU
		다 ARREAD (*Read f
ОК	Cancel Help	ARSORT (*Sort it
		Blocks Sov list Define EN

RPAC-2658M User Manual, V1.0, Mar. 2021 by ICP DAS 154

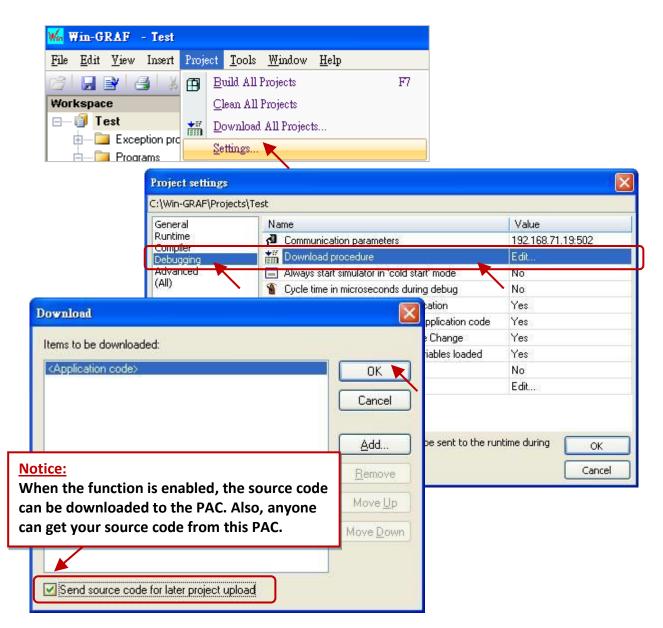
# 11.9 Upload the Win-GRAF Source Code

For some reason like to prevent the source code from being lost or an incomplete code was handed over by a previous worker, so it is necessary to upload the source code from the PAC to PC.

Otherwise, the user needs to enable the **Download Procedure** function and download the project to the PAC once. Next time, the Win-GRAF project can be uploaded with the source code.

### Enable the Download Procedure function:

- 1. Click the menu command **Project Settings** to open the settings window.
- 2. Double-click the **Download procedure** in the **Debugging** setting and check the "**Send source code for later project upload**" box, and then click **OK**.



 Click the menu command Project - Build All Projects to compile the project and download it to the PAC. The source code will be stored in /Systen\_Disk/Win-GRAF/t5.upl on the PAC. The file size varies depending on the content of the project. When the project becomes large and complex, the file size may reach several hundred K Bytes or even more than 1 MB.

### Upload the Source Code from the PAC:

First, click the menu command File - Close Project List to close all Win-GRAF projects.

4. Click the menu commands File - Add Existing Project - From Target and choose (or set) the PAC's IP address, and then set a project name (e.g., "UPL\_project1"). Finally, click "OK" to start uploading.

<u>F</u> ile	Liew <u>T</u> ools <u>W</u> indow <u>H</u> elp	21
	<u>N</u> ew Project List	
	Open Project List	*
1	Add New Project	
	Add Existing Project 💌	From Disk
	Print Setup	Egy From Zip
	1 C:\Win-GRAF\Projects\Test.w51	From Target
	2 C:\Win-	
	Settings	ОК ъ
	T5 Runtime	▼ 192.168.71.19:502 Cancel
	Pathnames	choose (or set)
	Destination folder : C:\Win-GRA	AF\Projects <u>B</u> rowse
	Name: UPL_project	ŧ1

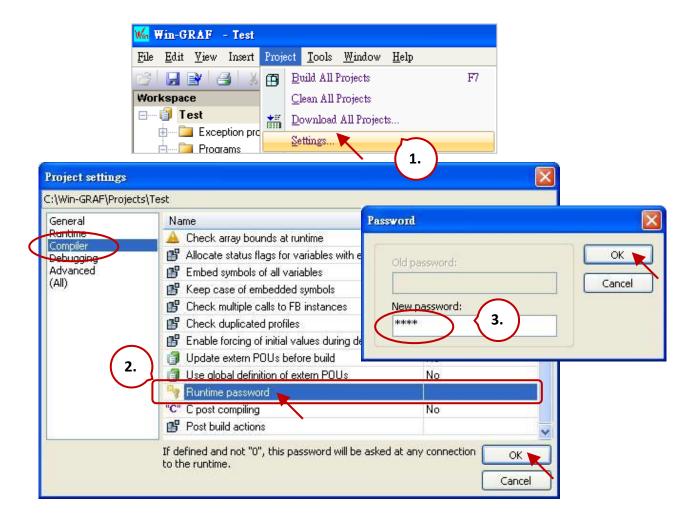
5. After uploading, click the "OK" button to open the Win-GRAF project automatically.

pen pr	oject from target	
From: To:	192.168.71.19:502/t5.upl C:\DOCUME~1\Janice\LOCALS~1\Temp	p\ZIP103.tmp
100 %		ОК
	Win-GRAF - UPL_project1 File Edit View Insert Project Tools	Is Window Help
	Workspace UPL_project1 Exception programs Programs B FBD1 B Main ST1 (*Data Conversi	☆   ☆   ☆   ☆   ☆   Ⅲ 品 96 € ☆   ■ 3g   田 隆 30   No selection! ■ I I X X
	Watch (for debugging)	Build X  Build Cross references Runtime Call stack Breakpoints Digital sa OffLine 192.168.71.19:502

# 11.10 Set Up the PAC Password

To prevent the PAC from being maliciously changed or stopped by a connected PC while running an important program, you can set a password for the PAC to forbid unauthorized operation.

- 1. Click the menu command Project Settings to open the settings window.
- 2. Double-click Runtime password in the Compiler setting to set a password and click "OK".



3. Click the menu command **Project - Build All Projects** to compile the project and download it to the PAC. Next time, the user needs to enter the password after clicking the "On Line" button.

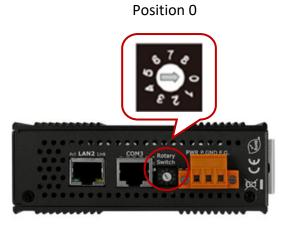
Password	
Password:	ОК
	Cancel

#### Note:

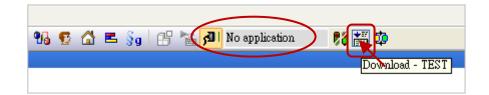
Remember the password after enabling the function or Win-GRAF cannot connect to the PAC.

### The Only Solution:

Set the rotary switch of the PAC to "**1**" (i.e., Safe mode) and reboot. After that, download the project again and Set the rotary switch of the PAC to "**0**" (i.e., Normal mode).



旋轉開關位置	說明
0	Normal Mode
1	Do not perform the user's application
2	LAN1: DHCP Mode
3	LAN1: Static IP Mode IP: 192.168.255.1 Mask: 255.255.0.0
4	Firmware Update Mode



## 11.11 Using Function or Function Block in the ST Program

It is easy to use a Function in the ST program, simply call the Function and assign the corresponding parameters. The example below will open the COM1 at the beginning and then send a `Hello' string through the COM1 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(1) = FALSE THEN
TMP_BOO := COM_open (1, `19200,N,8,1');
END_IF;
IF TMR1 >= T#5s THEN
TMR1 := T#0s;
COM_send_str (1, `Hello: ');
END_IF;
```

To use a <u>Function Block in the ST program</u>, you must declare an FB instance variable in the **Variables** window. After that, the usage is similar to the Function as follows:

For example, unpack one Byte to 8 Booleans.

1. Declare the "MyUnpack" as a "UNPACK8" (FB) variable and "IN" as a "BYTE" variable.

Name	Туре	Dim.	Attrib.	Syb.	Init value	U V	Tag	Description	1
🗉 🚮 Global vari	iables								1
IN Multiment	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 :	
Q5 := MyUnpack.Q5 ; Q6 := MyUnpack.Q6 ;	
Q7 := MyUnpack.Q7 ;	

# 11.12 Protect Your Win-GRAF Program to Avoid Unauthorized Use

When you finish developing a Win-GRAF application and prepare to deliver it to the customer, please think that your application on the PAC may be copied into another PAC with the same model by a third-party?! Be aware! Someone else may steal your hard outcome! The following provides a simple and easy way to protect your application.

### Note:

If you give the source code of a Win-GRAF application to someone, then sorry, the following method cannot protect your program. Anyone who got the source code can modify the code and apply it to another PAC.

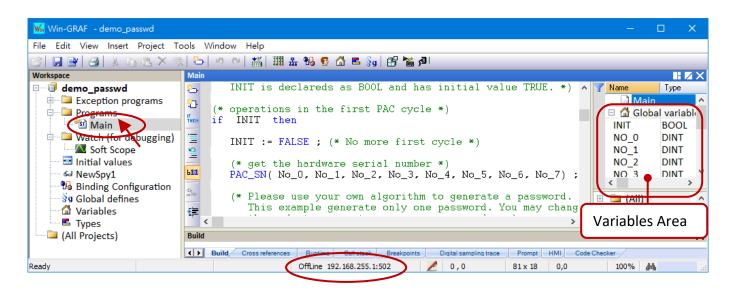
Each ICP DAS Win-GRAF PAC has a unique 64-bit serial number that can be used to combine with the custom algorithm to generate a password. Also, you can add the validation method to the project. When the project is downloaded to the PAC, it will check if the password is correct. If it is incorrect, the program will not be executed.

### **Description of demo programs**:

- 1) "demo\_passwd": Used to generate a password and save it as a file in the PAC.
- 2) "demo\_my\_ap": The project with the password verification function to give to the customer.

The user needs to download the "demo\_passwd" project to the PAC and run it once to generate a unique password. Then, download the "demo\_my\_ap" project to the same PAC. After that, even if the Win-GRAF project is copied to the other PAC with the same model, it will not work due to the password validation failure.

Download <u>the demo program</u> (demo\_passwd.zip and demo\_my\_ap.zip) on the website. Click the menu commands "File - Add Existing Project - From Zip..." in the Workbench to open the project.



### Win-GRAF Project: "demo\_passwd"

First, using the "PAC\_SN" function to read the serial number of the PAC and add the custom algorithm to generate a password, and then store the password in a file. The user can specify the file location.

### Variable Declaration:

Workspace	Variables						🕂 🛛 🗎
🗆 🗐 demo_passwd	🍸 Name	Туре	Dim. Attrib	. Syb.	Init value	User Tag	Description
🗄 🔤 Exception programs	🗉 🚮 Global	variables					
🛓 🔤 Programs	INIT	BOOL			TRUE		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					
Manual Binding Configuration	NO_5	DINT					
gg Global defines	NO_6	DINT					
Variables	NO_7	DINT					
E Types	password	DINT					
	<						>

### ST Program - Main:

- (\* This "demo\_passwd" project will generate a password by the 8-Byte Serial Number of the PAC and save it in the /System\_disk/Win-GRAF/my\_product.pwd on 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;
```

### Win-GRAF Project: "demo\_my\_ap"

First, using the same way as described before to generate a password and compare the password that is stored in the file on the PAC. If the password is incorrect, quit the whole program.

Workspace	Main		
🖃 🗐 demo_my_ap		7 Name	Type 1
🞰 🚞 Exception programs	(* operations in the first PAC cycle *)	🗉 🚮 Global va	riables 🛛 🔺
🖻 — 🦲 Programs	if INIT then	INIT	BOOL
Main		NO D	DINT
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	
Initial values		<	>
😪 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)	1
🚽 🖇 🚽 🚽 🚽 🚽	(No_4 * No_5) + No_6 + No_7 ;	🗄 📄 Advanced	
🔤 🚮 Variables		Arithmetic	×
E Types	Main LD2 NewSpy1	Blocks Sov lis	st Define ENUM

**Note:** Refer to <u>Section 2.1.2</u> to arrange programs in the execution order.

### Variable Declaration:

Workspace	Variables							
∃— 🗊 demo_my_ap	🝸 Name	Туре	Dim. Attrib.	Syb.	Init value	User	Tag	Description
🗄 🛅 Exception programs	🗉 🚮 Global variable	s						
🖨 🦳 Programs	INIT	BOOL			TRUE			init as TRUE
Main	NO_0	DINT						8-byte hardware serial number
102 million	NO_1	DINT						
🖮 🛄 Watch (for debuggi	NO_2	DINT						
Soft Scope	NO_3	DINT						
🔤 Initial values	NO_4	DINT						
NewSpy1	NO_5	DINT						
📲 🚮 Binding Configurati	NO 6	DINT						
Sa Global defines	NO 7	DINT						
C 🖓 Variables 💫	password	DINT						
E Types	PAC password	DINT						
24-2 (	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 in the /System\_disk/Win-GRAF/ my\_product.pwd on 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 *)
```

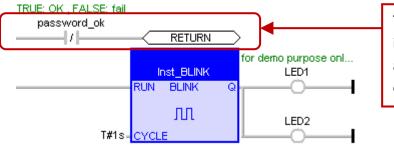
```
(* get the hardware serial number *)
PAC_SN( No_0, No_1, No_2, No_3, No_4, No_5, No_6, No_7);
```

```
(* 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" which means the password is incorrect, it will quit the program. The program can be executed only when the password is correct so that protects your application from unauthorized use by others.



There is only one "LD2" program in this project. The user must add a line of "password\_ok" code for each program.

# Chapter 12 Description of Win-GRAF Demo Projects

The user can download Win-GRAF demo programs on the website:

#### Follow the steps below to execute the demo program:

- 1. Click the menu commands "File Close Project List" to close all open projects.
- 2. Click the menu command "File Add Existing Project From Zip" to restore a demo program (e.g., "demo\_tmr1.zip"). Also, refer to Section 11.4 to back up or restore the project.
- 3. Right-click the project name and select "Communication Parameters" to set the IP address of a PAC.
- 4. Click the menu command "**Project On Line**" to download the project to the PAC. (Refer to Section2.3.4).

	ools <u>W</u> indow <u>H</u> elp	> 1編 I 패 ዱ 166 15 쇼 트 Sg ( đ	4.	
Workspace demo_tm1 Exception programs Programs Wain Watch (for debugging) Soft Scope Initial values WewSpy1 Binding Configuration S Global defines Variables Types	Main (* Set S if STAR ST t: and (* Set S if STOP STOP_ti STOP_ti STOP_ti Stop( end_if; Build (* Set S	T5 Runtime           192.168.71.19:502           192.168.255.1:502           192.168.71.18:502           192.168.71.19:502	Cancel Browse Help S Digital sampling trace	Name Type Main Main Global variables START_tmr BOOL STOP_tmr BOOL RESET_tmr BOOL LED1 BOOL LED2 BOOL The variable area
Ready		OffLine 192.168.71.18:502	🥒 Ln 1, Ch 1	0 x 1 Ln 1, Ch 1 100%

### Before downloading the demo program to the PAC, make sure these settings are correct:

- Win-GRAF communication IP, Modbus TCP Slave IP. Default IP addresses of RPAC - LAN1: 192.168.255.1, LAN2: 172.16.255.1, LAN3: 10.0.255.1
- 2. The setting of COM port or baud rate. COM1: RS-232, COM2: RS-485, COM3: RS-422
- 3. File location (e.g., /mnt/microSD/ or /System\_Disk/Win-GRAF/).

Users can refer to Section 12.1 to modify parameters properly in the following window.

- Main	LD Program	Ħ	I/O Boards
• Main	ST Program	品	I/O Drivers
🔤 🔂 Variables	Variables	93	Binding

# 12.1 The List of Demo Programs

File Name	Description	Chapter		
win-graf-linux-demo-a	II.zip : Download all Win-GRAF demo programs. (674 KB)			
demo_et7060	Connect the remote ET-7060 Ethernet I/O module. I/O Drivers: Modbus Slave IP (10.0.255.2:502)			
demo_et7018z	Connect the remote ET-7018Z Ethernet I/O module. I/O Drivers: Modbus Slave IP (10.0.255.2:502)			
demo_tgw725	To access multiple Modbus RTU Slave via tGW-700 Gateway I/O Drivers: Modbus Slave IP (Port1): 10.0.255.2:502 Modbus Slave IP (Port2): 10.0.255.2:503			
demo_retain	Use retain variables to store last data before a PAC power failure.	<u>6.1</u>		
demo_wp5_retain	Store retain data to a file. Variables: File_Path1 ('/System_Disk/Win-GRAF/retain_real.txt') Variables: File_Path2 ('/System_Disk/Win-GRAF/retain_other.txt')	<u>6.2</u>		
demo_binding	Exchange data between PACs (Data Binding) Binding: Exern (192.168.255.1/10.0.255.1)	<u>Z</u>		
Remote DCON I/O Mo I/O Boards: DCON (Po	<b>dule (8.2.1 ~ 8.2.8)</b> rt = 2, Baud rate = 9600)			
demo_d_7065	Connect the remote I-7065 I/O module. Function Block: D_7065 (Port = 2, Address = 2)	<u>8.2.1</u>		
demo_d_7018z	Connect the remote I-7018Z I/O module. Function Block: D_7018Z (Port = 2, Address = 3)			
demo_d_7083	Connect the remote I-7083 I/O module. Function Block: D_7083 (Port = 2, Address = 4)	<u>8.2.3</u>		
demo_d_87084_fr	Connect the remote I-87084W module to measure frequency. Function Block: D_7084_FREQ (Port = 2, Address = 5)	<u>8.2.4</u>		
demo_d_87084_c4	Connect the remote I-87084W module to measure the Counter.	<u>8.2.5</u>		
demo_d_87084_c8	<ul> <li>Function Block: D_7084_CNT4 (Port = 2, Address = 5)</li> <li>Function Block: D_7084_CNT8 (Port = 2, Address = 5)</li> </ul>	<u>8.2.6</u>		
demo_dl_100T485	Connect the remote DL-100T485 (temperature/humidity) module. Function Block: DL_100T485 (Port = 2, Address = 1)	<u>8.2.7</u>		
dmeo_gps721	Connect the remote GPS-721 GPS Receiver module Function Block: D_GPS721 (Port = 2, Address = 1)	<u>8.2.8</u>		
demo_my_ap	Protect your Win-GRAF program and avoid being embezzled.	11 12		
demo_passwd	Main (ST): file_name ('/System_Disk/Win-GRAF/my_product.pwd')	<u>11.12</u>		
demo_tmr1	Use the "tStart" and the "tStop" functions to operate the Timer.	<u>12.2.1</u>		
demo_tmr2	To do periodic operations for the Timer.	<u>12.2.2</u>		
demo_tmr3	To do periodic operations more accurately for the Timer.	<u>12.2.2</u>		

Follow the steps as noted above to restore the demo program to the Win-GRAF Workbench.

demo_com_port1	Send a string by the COM Port. Variables: Port_number (1: RS-232), Main (ST): '9600,N,8,1'	<u>12.3.1</u>				
demo_com_port2	Request/answer the device by the COM Port. Variables: Port_number (2: RS-485), Main (SFC): '9600,N,8,1'	<u>12.3.2</u>				
demo_com_port3	Wait for data coming from the remote device to the COM Port. Variables: Port_number (2: RS-485), Main (ST): '9600,N,8,1'	<u>12.3.3</u>				
demo_com_port4	Report data periodically to the remote device by the COM Port. Variables: Port_number (2: RS-485), Main (SFC): '19200,E,8,2'	<u>12.3.4</u>				
demo_file1	Write data to a file on the PAC. Main (ST): '/System_Disk/Real_data1.txt'	<u>12.4.1</u>				
demo_file2	Read data from a file on the PAC. Variables: File_path ('/System_Disk/Real_data2.txt')	<u>12.4.2</u>				
demo_datalog Data Logging Main (ST): /mnt/microSD/yyyy-mm/yyyy-mm-dd.csv		<u>12.4.3</u>				
demo_vb03 demo_vb04	Use the C program to read/write Win-GRAF variables.	<u>13</u>				
<ul> <li>Redundant System.</li> <li>i_redundancy:</li> <li>Fixed LAN2 IP:</li> <li>Custom security r</li> </ul>	Active_IP = 192.168.71.37, Mask=255.255.0.0 199.193.195.17 (Main)/ 199.193.195.9 (Backup) <b>nechanism</b> : Refer to the <b>RDN_control</b> program					
Function Block:						
demo_rdn_1	D_87064 \ D_87018Z \ D_7065 (Port = 2, Address = 2 \ 3 \ 4) I/O Boards:					
	DCON (Baud rate = 9600) ` i redundancy ` i redundancy rs485	<u>14</u>				
demo_rdn_2	I/O Boards: i redundancy	-				
demo_rdn_3	LAN3, ET-7050 I/O Drivers: Modbus Slave IP (10.0.255.1:502) I/O Boards: i_redundancy					
demo_rdn_4	LAN3, iDCS-8830 I/O Drivers: Modbus Slave IP (10.0.79.200/10.0.79.201) I/O Boards: i_redundancy					
demo_schedule	Schedule control. I/O Boards: Schedule (Password: 0) Binding: Control Variables (ID: 5001 to 5030)	<u>15</u>				
demo_sms	emo_sms Use the 2G/3G GSM Modem to send/receive the SMS message. Variables: Phone_Nb ('0900629879')					
Enable a serial port for connecting the Win-GRAF Workbench.demo_extra_portVariables: file1_name ('/System_Disk/Win-GRAF/Extra_Ports.txt')ST1 (ST): 'COM2:115200,N,8,1'		<u>C</u>				
	ST1 (ST): 'COM2:115200,N,8,1'					

## **12.2 Timer Operations**

### 12.2.1 Start, Stop and Reset the Timer

Open the project ("demo\_tmr1.zip") and view variables in the variable area. Refer to Chapter 12.

### ST Program:

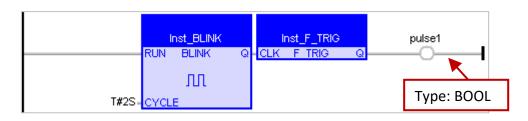
```
(* Declare "START_tmr", "STOP_tmr", "RESET_tmr", "LED1", "LED2" as BOOL
  Declare "TMR1" as TIME *)
(* Set START_tmr as TRUE to start ticking Timer TMR1 *)
IF START tmr THEN
 START tmr := FALSE;
 TSTART (TMR1);
END IF;
(* Set STOP_tmr as TRUE to stop ticking Timer TMR1 *)
IF STOP tmr THEN
   STOP 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

Open the project ("demo\_tmr2.zip") and view variables in the variable area. Refer to Chapter 12.

In this example, using both the "**BLINK**" function and the "**F\_TRIG**" function block generates a pulse TRUE at specific intervals. It can be applied to periodic operations.

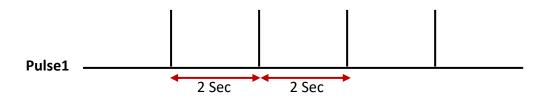
### LD Program:



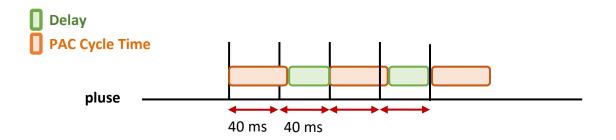
### ST Program:



The LD program above can be used to generate a pulse TRUE every two seconds. But, this method has a drawback:



When the time interval is shorter (e.g., less than 100 ms) or the PAC Cycle Time is larger (e.g., 20 to 50 ms, typically 3 to 15 ms), then the operating time will be inaccurate. For example, set it to execute the operation every 40 ms, as compared to 2 seconds or 250 ms, the interval time of 40 ms is very close to the PAC Cycle Time. Using this way is easy to accumulate the output delay time and the operating time finally becomes inaccurate.



To improve the situation, the following method of coding will be more accurate:

Open the project ("demo\_tmr3.zip") and view variables in the variable area. Refer to Chapter 12.

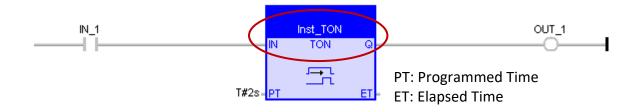
#### ST Program:

```
(* 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;
IF TMR1 >= TMR1_next THEN
  IF TMR1 > T#10h THEN
     TMR1 := T#0s;
     TMR1_next := T#0s;
  END_IF;
  TMR1 next := TMR1 next + T#50 ms;
  (* Do periodic operations here *)
  • • •
END_IF;
```

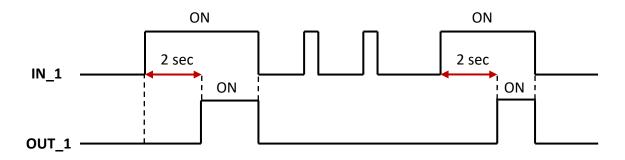
When the timer reach T#23h59m59s999ms, the value will overflow. Therefore, reset it automatically to "0" after 10 or 18 hours.

### 12.2.3 Detect the Steady ON or Steady OFF Signal

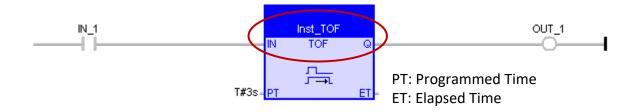
**"TON"** function block can detect a steady signal that remains "ON" for a specified time.



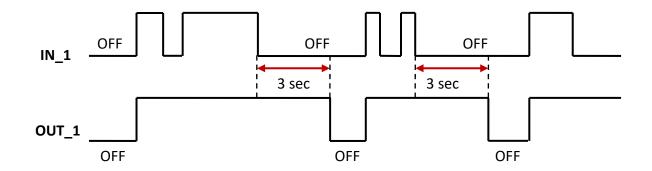
In the example, the function can detect a signal that remains "**ON**" for at least 2 seconds.



"TOF" function block can detect a steady signal that remains "OFF" for a specified time.

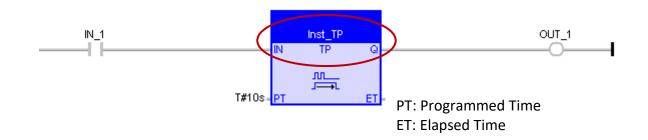


In the example, the function can detect a signal that remains "**OFF**" for at least 3 seconds.

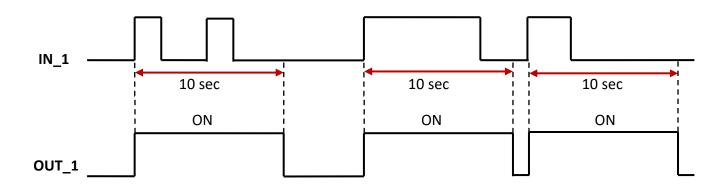


## **12.2.4** Keep Outputting ON for Some Time after Triggering

**"TP"** function block can keep outputting "**ON**" for a specified time after it is triggered (i.e., from OFF to ON).



In the example, when the IN\_1 is ON, it will keep outputting an "**ON**" signal in 10 seconds.



# **12.3** Operations of Serial Port Communication

Serial communication is often used either to control or to receive data, the user can use the following functions to directly perform specified operations to the serial port (e.g., RS-232, RS-485, or RS-422).

Functions	Description			
COM_open	Open a serial port.			
COM_close	Close a serial port.			
COM_clear	Clear the input buffer of a serial port.			
COM_test	Test if any Byte data received by a serial port?			
COM_send	Send Byte data to a serial port.			
COM_send_str	Send String data to a serial port.			
COM_recv	Receive Bytes data from the input buffer of a serial port and save them in a Byte Array.			
COM_status	Get the current status of a serial port.			

Refer to <u>Section 1.2.3</u> to open the Library Manager and look up the description of functions.

	📶 Library Manager - ICP DAS - XP-WP-YP	×
1000	<u>File Wizard H</u> elp	
Q	Function and FBs //Os Profiles AS-i Types	
	COM_clear (* Clear the input buffer of a serial COM port *)	^
	FCOM_close (* Close a serial COM port *)	-
	COM_open (* Open a serial COM port. For ex., TMP_BOO := COM_open(2, 9600,N,8,1'); *)	
	COM_recv (* Receive bytes form the input buffer of a serial COM port and save them in a byte array *)	
	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_test (* Test if any data received in the input buffer of a serial COMM. port *)	-
	Convert to For & Convert a Dississi input value to an Environmine value #	1.00
	Parameters Description	
N.L.N.	COM_clear : Clear the input buffer of a serial COM port .	^
30	*** Function	
100	*** Input parameters	
	Port : DINT : serial COMM. port number to open .	
	can be 1 to 37 (depends on PAC, means COM1 to COM37) .	
.07	*** Output parameters	
	Q : BOOL :	
	TRUE : clear successfully.	
	FALSE : something wrong (for example, port number is incorrect).	
	Tur 10 2014	
- 3	Jun. 18, 2014	
010	ICP DAS , Taiwan	

### 12.3.1 Send a String via a COM Port

Open the project ("demo\_com\_port1.zip") and view variables in the variable area. Refer to Chapter 12.

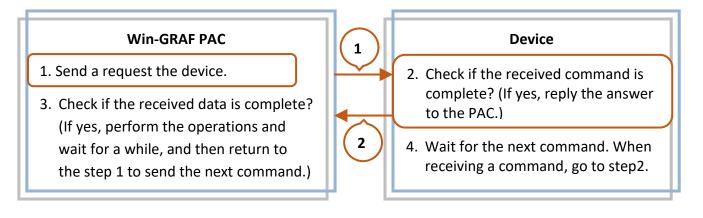
### ST Program:

Used to send a String (e.g., <CNT1=1>...<CNT1=100>) every 2 seconds through the COM1 (`9600,N,8,1') of the PAC.

```
(* Operations in the first PAC cycle *)
  if INIT then
                                              Declare "INIT" as BOOL and has initial value TRUE;
                                                      "Port_OK" as BOOL ;
   INIT := FALSE ; (* No more first cycle *)
                                                      "CNT1" as DINT ;
   CNT1 := 0;
                                                      "TMR1", "TMR1_next" as TIME
   TMR1 := T#0s ;
                                                      "Port number" as DINT and has initial value "1"
   TMR1 next := TMR1 + T#2s ;
   (* start ticking TMR1 *)
   tStart(TMR1);
  end if;
   (* 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;
```

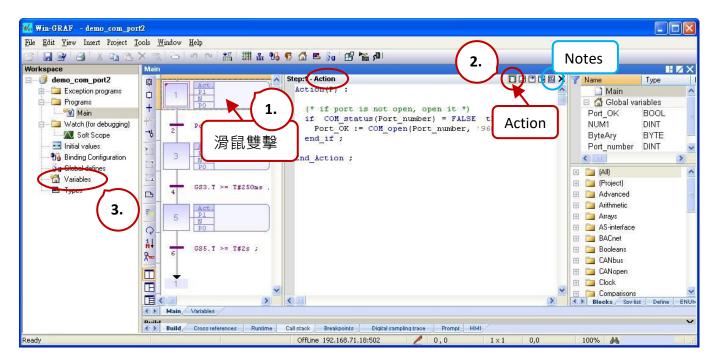
## 12.3.2 Request-Reply Communication via a COM Port

If it needs to get data from a device through an RS-232/RS-485/RS-422 port, using the Request-Reply communication as follows:



Open the project ("**demo\_com\_port2.zip**") and view variables in the variable area.

1. Double-click an **Action** to open the **Notes** window and change it to the "Action" window to view the code.

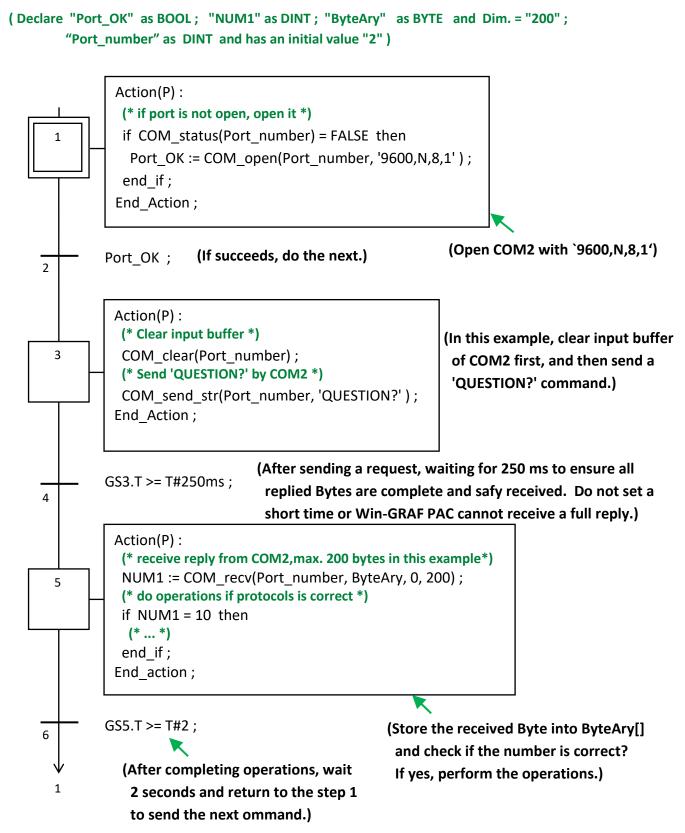


2. Also, click the "Variables" tag to open the Variables window.

Va	riables									ZΧ
T	Name 🗸 🗸	Туре	Dim.	Attrib.	Syb.	Init value	User Group	Tag	Description	
	🗉 🚮 Global vari	ables								^
	Port_OK	BOOL								
	Port_number	DINT				2			init as 2	
	NUM1	DINT								
	ByteAry	BYTE	[0199]						Byte Array, DIM=200	$\checkmark$
	<									>

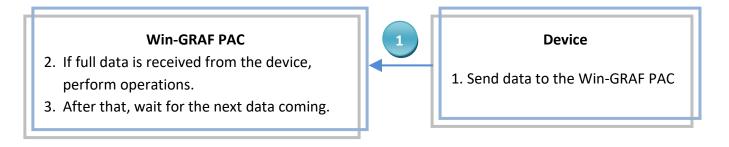
In this example, the Win-GRAF PAC will send a 'QUESTION?' string to the device through the COM2 and waits for a complete reply before processing. After completing operations, wait for 2 seconds to send the next command, and repeats the procedure.

### SFC Program:



## 12.3.3 Wait for a Remote Device to Send Data to a COM Port

This way is commonly used in convenience stores or supermarkets like the use of barcode reader. After reading the product barcode, it will send data to a COM Port (RS-232/485/422) of a Win-GRAF PAC, and no response is required.



Open the project ("demo\_com\_port3.zip") and view variables in the variable area. Refer to Chapter 12.

### 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 to wait and check if any data received by COM2?** If returns TRUE means there is some data. Then set STEP1 to "1" and T1 to "0" to start timing.

Declare "INIT" as BOOL and has initial value TRUE;

"ByteAry" as BYTE and Dim. = "200";

"Port number" as DINT and has initial value "2".

"Port\_OK" as BOOL ;

"T1" as TIME ;

"STEP1", "NUM1" as DINT;

```
(* wait 250 ms, then receive all bytes by the COM port *)
```

```
if T1 >= T#250ms then
Tstop(T1) ;
T1 := T#0s ;
STEP1 := 0 ;
```

1:

STEP = 1 means data is comeing, waits for 250 ms to receive full data and store them into an array. The waiting time concerns the specifications and Baud Rate of the device, set a short time may causes data lost. Remember to set STEP1 to "0" to wait for next data coming.

```
(* 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;

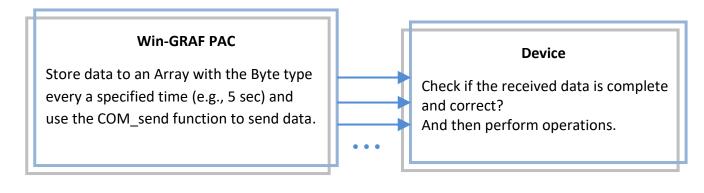
After receiveing full data, check if data is correct? If yes, perform the operations.

```
end_if;
```

END\_CASE ;

## 12.3.4 Reply Data to the Remote Device Periodically via a COM Port

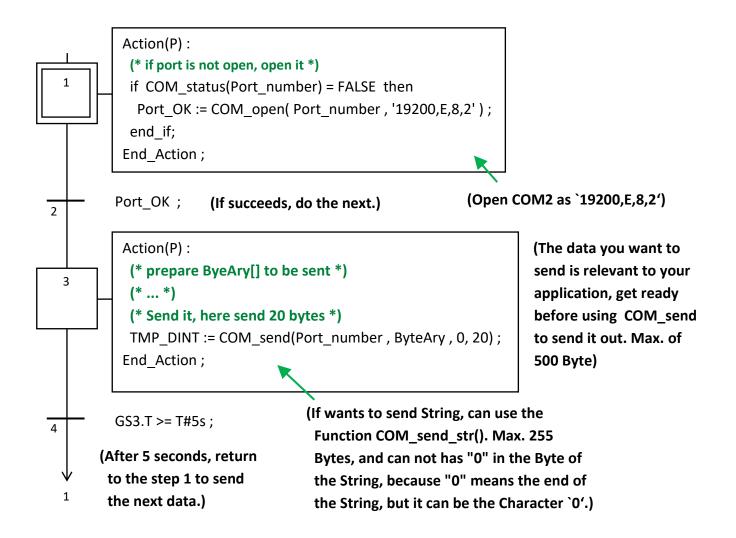
To periodically reply data to the remote device through an RS-232/RS-485/RS-422 port, the process of the method is as follows



Open the project ("demo\_com\_port4.zip") and view variables in the variable area. Refer to Chapter 12.

### SFC Program:

```
(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

Win-GRAF Workbench provides the following functions to enable read/write operations in a file on the PAC. For more information, open the "HTML Help" window and search the word "**File**", and then select the **File Management functions** topic.

Functions	Descriptions				
Help <u>Jopics</u> <u>Search</u> <u>Tu</u> torials <u>R</u> <u>A</u> bout Language	内容(C) 搜尋(S) 我的最愛(I)         輸入要搜尋的字(W):         File       ✓         列出主題(L)       顯示(D)         選取主題(T):       找到: 182         標題       位置       階 ▲         File Management functions       IEC 61131-3       1				
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 the file is reached in a file open for reading.				
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	ME Rename a file.				
Refer to Section	efer to Section 1.2.3 to look up the description of these functions.				
F_dir	Create a directory.				
F_cp_dir	dir Copy a directory and all its contents to another directory.				
F_del_dir	Delete a directory and all its contents.				

**Note:** Win-GRAF Linux PAC does not support F\_SAVERETAI and F\_LOADERETAIN functions.

### 12.4.1 Write Data to a File on the PAC

Open the project ("demo\_file1.zip") and view variables in the variable area. Refer to Chapter 12.

**<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	: BOOL					
Tmp_string	: String, len=255					
File_ID	: DINT					
REAL_val[09] : REAL						
ii	: DINT					
File_Status	: String, len=128	*)				

\*)

- (\* 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
```

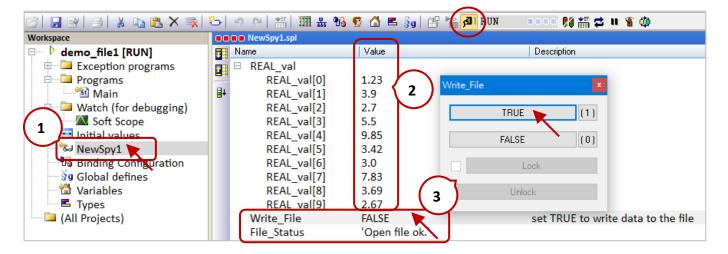
Because the capacity of /System\_Disk/ is small; you can change the file path to **/mnt/microSD/**. Also, the SD Card must be installed in the PAC.

```
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 ;
end_if ;
To write Integers by using the code below:
Tmp_string := Any_to_string( DINT_val[ii] ) ;
and declare the "DINT_val" variable as DINT
and the Dim. at least "10"
```

### Test Program:

In this example, when the "Write\_File" is set to "TRUE", it will write data to the **/System\_Disk/ Real\_data1.txt** on the PAC.

 After downloading the project to the PAC, click "NewSpy1" to open a Spy List and fill in the values to be written, and then set the "Write\_File" to "TRUE" to Write data. (If OK, "File\_Status" will show "Open file ok".)



2. Find out the "Real\_data1.txt" file on the PAC by using SSH (refer to Section 13.2).

SmarTTY - 192.168.79.11			—		×
File Edit View SCP Tools Help					
Welcome to 1 Browse File System on Remote Host		× <sup>f5</sup>	am	av71)	^
* Document; * Managemen	-				
* Support:	File Edit	View	SCP	Tools	Help
Last login:	1	1.23	3		
ubuntu@RPAC-	2	3.9			
	3	2.7			
File name: Real_data1.txt	4	5.5			
	5	9.85	5		
Open 🍗	6	3.42	2		
	7	3			
🛅 ubuntu@RPAC-2658M:~ 🗙 🕂 🗲 🎓	8	7.83	3		
	9	3.69	9		
SCP: No transfers	10	2.67	7		

# 12.4.2 Read Data from a File on the PAC

Open the project ("demo\_file2.zip") and view variables in the variable area. Refer to Chapter 12.

**<u>ST Program</u>**: This program can be used to read 10 "REAL" values from a file on the PAC.

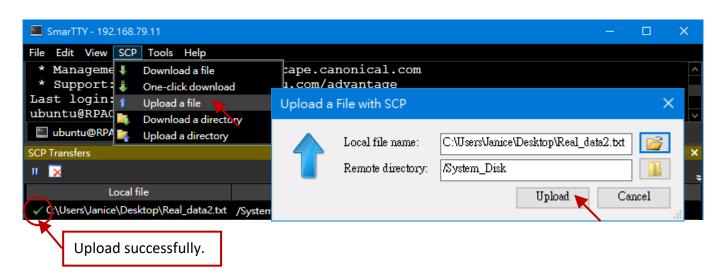
```
(* this "demo file2" project will read 10 REAL value from a file in the /System Disk/Real data2.txt.
   File format :
       Each row contains one REAL value and ends with <CR><LF> characters. Like:
       1.08
       2.786
       38.45
       41.5
       59.875
       60.76
       71.23
       80.5
       99.8
       100.7
*)
(*
 Variables Declaration:
 Write_File
              : BOOL
 Tmp_string : String, len=255
 File ID
               : DINT
 REAL_val[0..9] : REAL
 ii
                : DINT
                : String, len = 128, initial val = '/System Disk/Real data2.txt'
 File path
 File_Status : String, len=128
                                                             Because the capacity of /System_Disk/
*)
                                                             is small; you can change the file path
(* Set Read_File as TRUE to read data from the file *)
                                                             to /mnt/microSD/. Also, the SD Card
if Read_File then
                                                             must be installed in the PAC.
  Read File := FALSE ;
  (* Check if file exists *)
  if F exist(File path) = FALSE then
    (* file doesn't exist *)
    File_Status := 'File "' + File_path +'" does not exist !' ;
  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;
                                     To read Integers, use the code below:
                                     DINT_val[ii] := Any_to_DINT( Tmp_string ) ;
     (* close the file *)
     F_close( File_ID ) ;
                                     and declare the "DINT val" variable as DINT
                                     and the Dim. at least "10" for this example.
   end if;
 end if;
end if;
```

### **Test Program:**

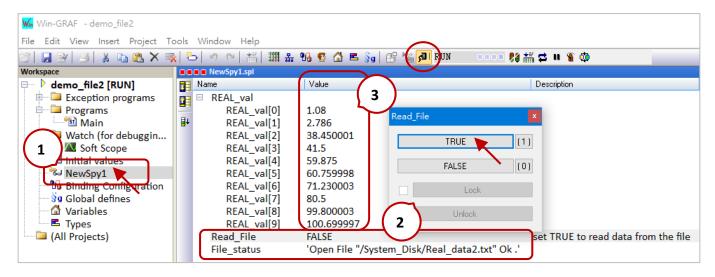
In this example, when the "Read\_File" is set to "TRUE", it will read data from the "/System\_Disk/ Real\_data2.txt" on the PAC.

 Upload the Real\_data2.txt file to the /System\_Disk/ directory by using the SSH tool. (Refer to Section 13.2)



2. After downloading the project ("**demo\_file2.zip**") to the PAC, click "NewSpy1" to open a Spy List and set the "Read\_File" to "TRUE" to read the data.

If OK, the read data will be displayed and "File\_Status" will show 'Open File "/System\_Disk/ Real\_data2.txt" Ok.'



### Note:

- When using the Linux PAC, change the symbol of the path to / (slash).
- Also, refer to Section 6.2 to write multiple data to a file by using retain variables.

# 12.4.3 Data Logging

Refer to Chapter12 to open this project ("**demo\_datalog.zip**"). The project provides a simple data logging function.

# **Description of the Project:**

The project creates a spy list that contains a String variable (write\_date), an Integer variable (int\_data), and a Real variable (float\_data). The variable data will be stored in a file per minute in the specified folder on the PAC.

The log file will be named according to the current date (e.g., Jan 02, 2021, the file name is "2021-1-2.csv"). Besides, the log file of the previous day will be moved to "/the specified folder/current month/" every day (e.g., Jan 01, 2021, the file will be moved to "/mnt/microSD/2021-1/").

<u>Note:</u> In the example, the log file is stored in **/mnt/microSD/** on the PAC. The user needs to install an SD card in the PAC.

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							

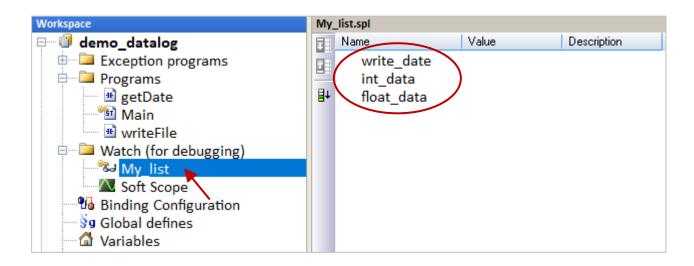
**Description of variables:** The user can view/set variables in the Win-GRAF "Variables" window.

Name	Data Type	Description
Year1		
Month1		
Day1		
WeekDay1		Used to get the PAC system time in the "PAC_Time" program.
Hour1	DINT	
Minute1		
Second1		
old_day		Used to know if the time is changed and then rename the log file
old_hour		in the "PAC_Time" program.
log_tmr1	ТІЛАГ	Timor
log_tmr2	TIME	Timer.
CSV_Path		The storage path of the CSV file.
CSV_Dir	STRING	The storage folder of the CSV file.
write_date		Used to record the time when writing data to a CSV file.
init	BOOL	Set it as TRUE to initialize. (Init value = TRUE)

Name	Data Type	Description
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	
tmp_msg	STRING	The temporary variable.

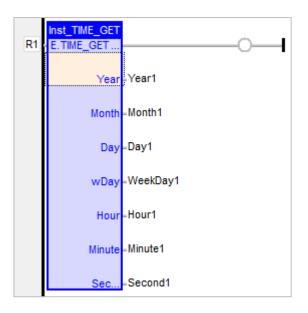
# Spy List – "My\_List":

This example creates a watch list called "My\_list". (Refer to Section 11.3 for more details)



# LD Program – "getDate":

To get/set the PAC system time.



**Note:** In the demo program, the file folder is **/mnt/microSD/**. The user needs to install an SD Card in the PAC.

### ST Program – "Main":

To generate random values as the data of the log file (.csv) and to change the file storage path.

```
(* Declare "init" as "BOOL" ; Declare "log_tmr1" and "log_tmr2" as "TIME" *)
(* Set "init" as TRUE to start ticking "TMR1" *)
IF init THEN
init := FALSE ;
  (* Enable the timer of the data logger *)
    TSTART (log_tmr1) ;
  (* Set the separator and decimal within the csv file *)
    SetCsvOpt(',','.');
END_IF;
IF old_day<> day1 THEN
    tmp_msg:='/mnt/microSD/'+CSV_Dir+'/'+ANY_TO_STRING(year1)+'-'+
        ANY_TO_STRING(month1)+'-'+ANY_TO_STRING(old_day)+'.csv';
  (* Copy the file to /mnt/microSD/ *)
    tmp_bval:=F_COPY (CSV_Path,tmp_msg);
```

```
(* Delete the original file stored at /run/ *)
tmp_bval:=F_DELETE(CSV_Path);
```

```
(* Create the CSV file and its field name *)
```

```
tmp_msg:='/run/'+any_to_string(year1)+'-'+any_to_string(month1)+'-'+
any_to_string(day1)+'.csv';
CSV_Path:=tmp_msg;
File_ID:= F_WOPEN(tmp_msg);
```

```
IF File_ID<>0 THEN
```

```
tmp_msg:='Time,Int_val,Float_val';
tmp_bval:=FM_WRITE(File_ID,tmp_msg);
tmp_bval:=F_CLOSE(File_ID);
```

END\_IF;

```
old_day:=day1;
```

```
END_IF;
```

## (\* 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:='/mnt/microSD/'+CSV_Dir;
    F_DIR(tmp_msg);
```

end\_if;

### (\* 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 \*)

### (\* Delete the previous file \*)

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

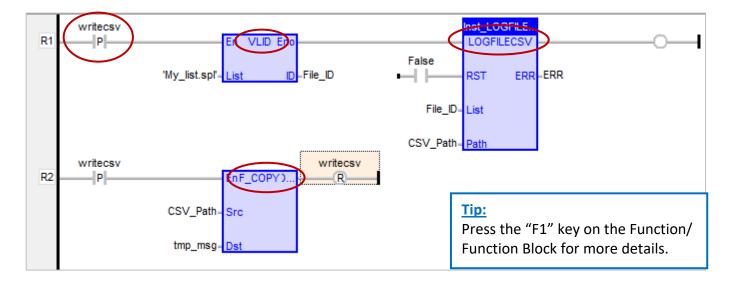
Because the capacity of /System\_Disk/ is small; you can change the file path to **/mnt/microSD/**. Also, the SD Card must be installed in the PAC.

```
(* Delete the file stored at /mnt/microSD/ *)
  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.



VLID: Get the identifier of an embedded list of variables (i.e., Spy List - My\_list.spl).

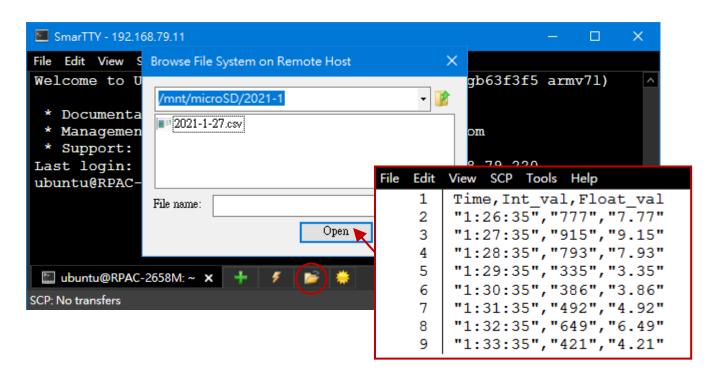
LogFileCSV: Generate a log file in CSV format for a list of variables.

**F\_COPY:** Copy a file. Using "F\_COPY" to copy a file from the "**/run/**" folder (RAM Disk, which can read/write data quickly) to "**/mnt/microSD/**". (Refer to ST program – "Main")

### **Test Program:**

The write\_date, int\_data, and float\_data variable data will be recorded and stored to **/mnt/microSD/** YYYY-MM/YYYY-MM-DD.csv (e.g., **/mnt/microSD/**2021-1/2021-1-27.csv).

Find out the .csv file on the PAC and check values by using SSH tools (refer to Section 13.2).



# Chapter 13 Using a C Program to Access Win-GRAF Variables

The Chapter describes how to create public variables in the Win-GRAF project for the C program to read/write data. The C demo program is developed by using the GCC tool as well as two Win-GRAF projects are provided.

### <u>C demo program</u> - Quicker\_Demo.tgz

Used to read/write Win-GRAF Char, Short, Long, ULong, Int64, Float, Double, or String variable.

### Win-GRAF demo programs - demo\_vb03.zip and demo\_vb04.zip

Download <u>the demo program</u> on the website and run the **File** menu commands **Add Existing Project** and then **From Zip...** in the Workbench to open the project.

# 13.1 Publishing the Win-GRAF Variable for C Program

In the Win-GRAF, except for the String variable, all the public variables and IDs must be set in the "Binding" window so that the C program can access data with the same address.

Refer to <u>Chapter 7</u> to create public variables. After completing the settings, the results are shown below.

Note: Up to 8192 public variables can be created and the range of ID is 1 to 8192.

File Edit View Insert Project To	ols	Window	Help	~		$\frown$						
27 🖬 🕑 🛃 🗴 🖬 🛍 🗙 🤿		5 9 9	<b>H</b>	III # %		1.	ig   65 '	ka 🖌 🖌				
Workspace	Bin	ding										ZΧ
🖙 🕡 demo_vb03 🌒	몲	⊡ <mark>¤¦</mark> ⊟ Ne	twork				Name	Value	7	Name	Туре	D
Exception programs	2	÷	PUBLI	IC (:9000)		2.	ame	PUBLIC		🗉 🚮 Globa	l variable	s_ ^
🖻 🛄 Programs	-		🕒 1:l	ong_val	$\mathcal{N}$	2.	ddress			long_val	DINT	
🖬 Main	∎∔		🔍 2:r	real_val			Port	9000		real_val	REAL	
🖻 📮 Watch (for debugging)			🕒 3:t	tmr_val						tmr val	TIME	
Soft Scope										INIT	BOOL	~
🔤 Initial values										<		>
Binding Configuration		Name		Identifier	P P	ositive	e hysteresis	Negative hys	N	ame	Value	
📲 🗿 Global defines		long_val		1	(							
🖓 Variables		real_val		2	- <b>X</b>	3.						
Types		tmr_val		3								
🛄 (All Projects)		<						>	<			>

### ST program (Main):

(\* Execute in the 1st scan cycle \*)
if INIT then
INIT := False; (\* Do not execute after the 1st scan cycle \*)
tstart(tmr\_val);
end\_if;

The following procedure will show you how to use the "pub\_string" function to publish the Win-GRAF String variable in the ST program.

Pub\_string (Address, String\_val) ;

Address: The public address number. It can be 1 to 1024 String\_val: The name of String variable.

Also, restore the demo program (demo\_vb04.zip) to view the codes.

Workspace	Main				ΖX
<ul> <li>demo_vb04</li> <li>Exception programs</li> <li>Programs</li> <li>Main</li> <li>Watch (for debugging)</li> <li>Soft Scope</li> <li>Initial values</li> <li>Binding Configuration</li> <li>Global defines</li> </ul>	<pre>if init then</pre>	~ 7	Name ■ ▲ Global msg1 init tmp_val NewVar msg2 msg3 <	Type variables STRING(128) BOOL BOOL WORD STRING(120) STRING(50)	Dim.

# The description of variables:

Name	Туре	Description
Init	BOOL	To initialize. The initial value is TRUE.
Tmp_val	BOOL	To determine if the setting is ok. TRUE: Binding succeeds. FALSE: Binding fails.
msg1	STRING, Length is 100	
msg2	STRING, Length is 32	The String variable to be policed. NOTE: The String length can be 1 to 255.
msg3	STRING, Length is 60	

# ST program (Main):

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;

# 13.2 Downloading and Compiling the C Program

Follow the steps below to download and compile the C program:

- 1. Download an SSH tool such as **SmarTTY** that supports SCP commands. <u>https://sysprogs.com/SmarTTY/download/</u>
- 2. Connect to the PAC by using SmartTTY.

Execute the SmartTTY and set up a new SSH connection.

Host name: Enter the IP address of PAC. (LAN1's default IP is '192.168.255.1')User name: Enter 'ubuntu'.Password: Enter 'temppwd'.

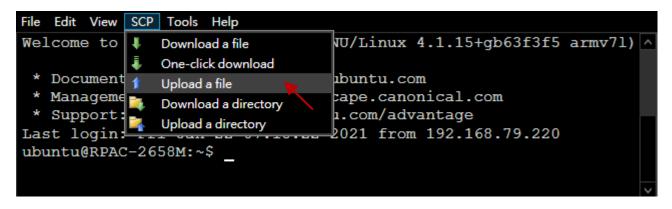
🔟 Smarl	TTY - New SSH	Connection				—		×
		Setup ne	w SSH	conn	ection	1		
	Host Name:	192.168.79.11						
	User Name:	ubuntu						
Conne	ection alias:							
	entication met							
	word: 💽 💽 🕻 etup public ke	•••• ey authentication	and don't ask	for passwo	ord again			
		ows key store (ass				rto RSA I	DSA	
Oper	nSSH key							
c	Override defau	It key location:						<b>1</b>
P	assphrase:							
Use H		T proxy:						
Enabl	e zlib compres	ssion (recommend	led for slow co	onnections	)			
Trans	fer folders usir	ng:	On-the-fly	TAR File-	by-file SCP (	slow)		
Enabl	e keep-alive p	ackets every:	0	sec	onds			
🗸 Save t	his connectior	n to connections l	ist					
						Connec	Ca	ncel
🔲 Virtual	Terminals							×
inside Sm	art Terminal ta suggestion p	/ introduces a nev abs, SmarTTY will popups and displa	let you use m	ouse and V	Vindows key	shortcut	s, show	
Please sel	ect the type o	f a terminal to cre	ate for the ne	w session:				
		🗳 St	art with a Sma	rt Termina	l			
		🛅 Sta	art with a regu	lar Termin	al 📐			
Reme	ember the cho	oice for all further	sessions					

If the connection is successful, the following message will be displayed.

SmarTTY - 192.168.79.11	-		×
File Edit View SCP Tools Help			
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.1.15+gb63f3)	£5	armv7	/1) ^
<pre>* Documentation: https://help.ubuntu.com * Management: https://landscape.canonical.com * Support: https://ubuntu.com/advantage Last login: Fri Jan 22 07:18:22 2021 from 192.168.79.22 ubuntu@RPAC-2658M:~\$</pre>	20		
<			>
🖭 ubuntu@RPAC-2658M:~ 🗙 🕂 👂 👹			
SCP: No transfers 2627B sent,	, 392	1B recei	ved 📑

**Note:** The first time you log in to the PAC, you will be asked to change the password.

- 3. Upload the Quicker\_Demo.tgz file to the PAC.
  - 1) Run the SCP menu command "Upload a file".



Note: LAN1, LAN2, and LAN3 must be set as different network segments, or the file cannot be uploaded due to a routing error

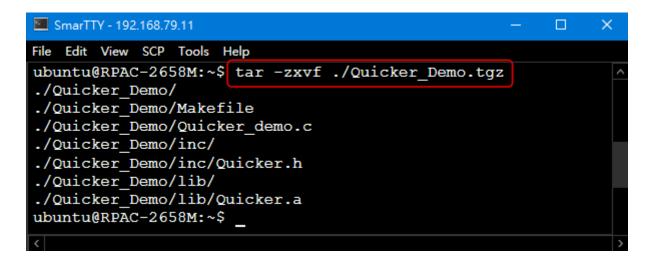
2) Download the **Quicker\_Demo.tgz** file to the **/home/Ubuntu** directory.

	Local file name:	C:\Users\Janice\Desktop\Quicker_Den	no.tgz	6
Т	Remote directory:	/home/ubuntu		
			Upload	Cancel

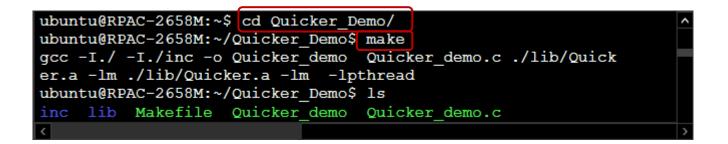
🗈 ubu	intu@RPAC-2658M: ~ 🗙	+ 🐔	📂 🕴	<b>)</b>			
SCP Tran	sfers						×
ш 🔀	Upload Successfully						*
	opicad caccessiany	]		Remote file	Size	Transfer rate	Progra
<b>√</b> C:\Us	ers\Janice\Desktop\Quicker	_Demo.tgz	/home/u	buntu/Quicker_Demo.tgz	4144	4046KB/s	

4. Extract the Quicker\_Demo.tgz archive file.

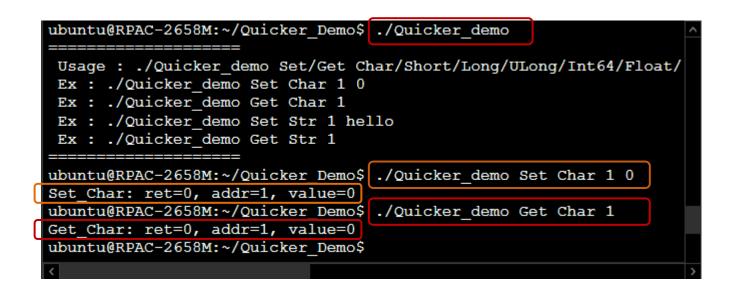
Using the 'tar –zxvf ./Quicker\_Demo.tgz' command to extract files to the Quicker\_Demo folder.



- 5. Run the make command to build the demo program.
  - 1) Type the 'cd Quicker\_Demo/' command to enter the directory.
  - 2) Type the 'make' command to build the program in the current folder.



6. Run the demo program to access Win-GRAF variables.



# 13.3 Accessing Win-GRAF Variables

There are three pubic variables - long\_val, real\_val, and tmr\_val in the Win-GRAF demo\_vb03 project.

🐱 Win-GRAF - demo_vb03				
File Edit View Insert Project Tool	s Window Help			
67 <b>. e</b> (3 . e (2 × 1×)	🍅 🖉 😢 🏭 🎟 🏪 😘 😨 🔂 🛋 💱	🛛 🗗 🕍 🚮 RUN	oooo 💔 🏭 🚅	u 🐒 🧔
Workspace	Binding]			
<ul> <li>demo_vb03 [RUN]</li> <li>Exception programs</li> <li>Programs</li> <li>Main</li> <li>Watch (for debugging)</li> <li>Soft Scope</li> <li>Initial values</li> <li>Binding Configuration</li> </ul>	B 5:tmr_vai = t#5m5255t	Name Name Identifier ng_val 5 5 5 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	Lock	

### long\_val:

The data type is **Long** and the ID is **1**.

- 1) Double-click long\_val in the Binding window to enter a value (e.g., 5).
- 2) Enter the **./Quicker\_Demo Get Long 1** command to read data.

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

real\_val:

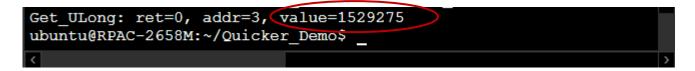
The data type is **Float** and the ID is **2**.

- 1) Enter the ./Quicker\_Demo Set Float 2 1.1 command to write data.
- 2) Enter the ./Quicker\_Demo Get Float 2 command to read data.

ubuntu@RPAC-2658M:~/Quicker_Demo\$	./Quicker_demo Set Float 2 1.1	
<pre>Set_Float: ret=0, addr=2, value=1.</pre>	100000	
ubuntu@RPAC-2658M:~/Quicker_Demo\$	./Quicker_demo Get Float 2	
Get_Float: ret=0, addr=2, value=1.	100000	
ubuntu@RPAC-2658M:~/Quicker_Demo\$	_	$\sim$

### tmr\_val:

The data type is **Ulong** and the ID is **3**. Enter the **./Quicker\_Demo Get ULong 3** command to read data. The unit of timer is microseconds, so the value 1529275 (ms) equal to 25 m 29 s 275 ms.



# 13.4 Descriptions of Functions in the "Quicker.a"

The Section describes functions in the "Quicker.a" static library.

"Quicker.a" provides many functions for users to read/write Win-GRAF variables data as follows:

- 1. Read/Write Boolean
- 2. Read/Write the 8-bit, 16-bit, 32-bit, or 64-bit Integer
- 3. Read/Write the 32-bit or 64-bit REAL

# 13.4.1 R/W a Boolean Variable

# Set\_BOOL

### **Description:**

Used to set the value of a Win-GRAF Boolean variable with the specific address no.

### Syntax:

unsigned char UserShare\_Set\_Coil(unsigned short iUserAddress, unsigned char iStatus)

### Parameter:

iUserAddress:The address no of the Win-GRAF variable (1 to 8192)iStatus:Set a Boolean value. (1: True; 0: False)

### Example:

'Set a Boolean variable value to True with address no.1 UserShare\_Set\_Coil(1, 1)

# Get\_BOOL

## Description:

Used to get the value of a Win-GRAF Boolean variable with the specific address no.

### Syntax:

unsigned char UserShare\_Get\_Coil(unsigned short iUserAddress, unsigned char \*iStatus)

### Parameter:

iUserAddress:The address no of the Win-GRAF variable no (1 to 8192)iStatus:Get a Boolean value. (1: True; 0: False)

### Example:

'Get a Boolean variable value with address no "1" unsigned char value; UserShare\_Get\_Coil(1, &value);

# 13.4.2 R/W an Integer Variable



### **Description:**

Used to set the value of a Win-GRAF 8-bit, 16-bit, 32-bit, or 64-bit Integer variable with the specific address no.

### Syntax:

unsigned char UserShare\_Set\_Char (unsigned short iUserAddress, char iStatus) unsigned char UserShare\_Set\_Short (unsigned short iUserAddress, short iStatus) unsigned char UserShare\_Set\_Long (unsigned short iUserAddress, long iStatus) unsigned char UserShare\_Set\_Int64 (unsigned short iUserAddress, long long iStatus) unsigned char UserShare\_Set\_ULong (unsigned short iUserAddress, unsigned long iStatus) unsigned char UserShare\_Set\_UChar (unsigned short iUserAddress, unsigned char iStatus) unsigned char UserShare\_Set\_UChar (unsigned short iUserAddress, unsigned char iStatus) unsigned char UserShare\_Set\_UShort (unsigned short iUserAddress, unsigned short iStatus) unsigned char UserShare\_Set\_UShort (unsigned short iUserAddress, unsigned short iStatus)

### Parameter:

iUserAddress:	The address no of the Win-GRAF variable (1 to 8192)
iStatus:	Set an 8-bit, 16-bit, 32-bit, or 64-bit Integer value

### Example:

'Set the value of a 32-bit integer variable with address no "1" to "1234567" UserShare\_Set\_Long(1, 1234567)

'Set the value of a 32-bit integer variable with address no "2" to "-1234" UserShare\_Set\_Long(2, -1234)

'Set the value of a 64-bit integer variable with address no "3" to "123456789012345" UserShare\_Set\_Int64(3, 123456789012345)

'Set the value of an 8-bit integer variable with address no "4" to "125" UserShare\_Set\_Char(4, 125)



### **Description:**

Used to get the value of a Win-GRAF 8-bit, 16-bit, 32-bit, or 64-bit Integer variable with the specific address no.

### Syntax:

unsigned char UserShare\_Get\_Char (unsigned short iUserAddress, char \*iStatus) unsigned char UserShare\_Get\_Short (unsigned short iUserAddress, short \*iStatus) unsigned char UserShare\_Get\_Long (unsigned short iUserAddress, long \*iStatus) unsigned char UserShare\_Get\_Int64 (unsigned short iUserAddress, long long \*iStatus) unsigned char UserShare\_Get\_UChar (unsigned short iUserAddress, unsigned char \*iStatus) unsigned char UserShare\_Get\_UChar (unsigned short iUserAddress, unsigned char \*iStatus) unsigned char UserShare\_Get\_UShort (unsigned short iUserAddress, unsigned short \*iStatus) unsigned char UserShare\_Get\_ULong (unsigned short iUserAddress, unsigned long \*iStatus) unsigned char UserShare\_Get\_ULong (unsigned short iUserAddress, unsigned long \*iStatus)

### Parameter:

iUserAddress:The address no of the Win-GRAF variable (1 to 8192)iStatus:Get an 8-bit, 16-bit, 32-bit, or 64-bit Integer value

### Example:

long long Dlong\_val; long long\_val; short short\_val; char sbyte\_val;

'Get the value of a 64-bit integer variable with address no "7" UserShare\_Get\_Int64(7, &Dlong\_val);

'Get the value of a 32-bit integer variable with address no "8" UserShare\_Get\_Long(8, &long\_val);

'Get the value of a 16-bit integer variable with address no "9" UserShare\_Get\_Short(9, &short\_val);

'Get the value of an 8-bit integer variable with address no "10" UserShare\_Get\_Char(10, &sbyte\_val);

# 13.4.3 R/W a REAL Variable

Set\_Float Set\_Double

#### **Description:**

Used to set the value of a Win-GRAF 32-bit or 64-bit REAL variable with the specific address no.

#### Syntax:

unsigned char UserShare\_Set\_Float(unsigned short iUserAddress, float iStatus) unsigned char UserShare\_Set\_Double(unsigned short iUserAddress, double iStatus)

#### Parameter:

iUserAddress:The address no of the Win-GRAF variable (1 to 8192)iStatus:Set a floating-point number

### Example:

'Set the value of a 64-bit REAL variable with address no "7" to "11234.234567" UserShare\_Set\_Double(7, 11234.234567)

'Set the value of a 32-bit REAL variable with address no "8" to "123.12" UserShare\_Set\_Float(8, 123.12);

# 📕 Get\_Float 📕 Get\_Double

#### **Description:**

Used to get the value of a Win-GRAF 32-bit or 64-bit REAL variable with the specific address no.

#### Syntax:

unsigned char UserShare\_Get\_Float(unsigned short iUserAddress, float \*iStatus) unsigned char UserShare\_Get\_Double(unsigned short iUserAddress, double \*iStatus)

#### Parameter:

iUserAddress:The address no of the Win-GRAF variable (1 to 8192)iStatus:Get a floating-point number

### Example:

float float\_val; double double\_val;

'Get the value of a 64-bit REAL variable with address no "7" UserShare\_Get\_Double(7, &double\_val);

'Get the value of a 32-bit REAL variable with address no "8" UserShare\_Get\_Float(8, &float\_val);

# 13.4.4 R/W a String Variable

# Set\_STRING

### **Description:**

Used to set Win-GRAF String data with the specific address no.

### Syntax:

int UserSetReg\_Str(unsigned short iUserAddress, char iStatus, unsigned short len)

### Parameter:

iUserAddress:	The address no of the Win-GRAF variable (1 to 1024)
iStatus:	Get/Set Win-GRAF String pointers
len:	Set the length of a Win-GRAF String

### Example:

char str\_value[256];
//Set String data with address no "7"
sprintf(str\_value,"hello world \n");
UserSetReg\_Str(7, str\_value, strlen(str\_value));

### Get\_STRING

#### **Description:**

Used to get Win-GRAF String data with the specific address no.

#### Syntax:

int UserGetReg\_Str(unsigned short iUserAddress, char \*iStatus, unsigned short len);

#### Parameter:

iUserAddress:The address no of the Win-GRAF variable (1 to 1024)iStatus:Get/Set Win-GRAF String pointerslen:Get the length of a Win-GRAF String

## Example:

char str\_value[256];
// Get String data with address no "7"
UserGetReg\_Str(7, str\_value, 256);

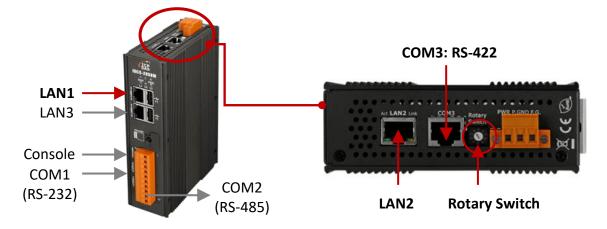
# **Chapter 14 Redundancy**

ICP DAS Win-GRAF Linux PAC - RPAC-2658M supports the redundant system:

One redundant system is composed of two Win-GRAF PACs. When the PAC that running programs is crashed or needs to release its control authority by user-defined event, the PAC control authority will automatically switch to the normal one.

Name Definition	Description
Main-PAC	The rotary switch is set to 7
Backup-PAC	The rotary switch is set to 9
Active-PAC	The PAC that is running the program
Passive PAC	The PAC that is synchronizing data

<u>Note:</u> The Main-PAC is Active-PAC at the beginning and its control authority will automatically be changed to the other PAC depends on conditions.



# Notice:

- 1. **Do not** use two Main-PACs or two Backup-PACs to form a redundant system.
- 2. Two PACs are communicating by using these communication ports.

Communication Port	Description
A. Public IP Port	LAN1: Used to communicate with Win-GRAF, SCADA, or HMI. <u>Note:</u> LAN3 is recommended if Ethernet I/O or devices are to be connected.
B. Replication Port	LAN2: Data Synchronization Port. Two PACs are connected with an Ethernet crossover cable for high-speed data synchronization. Do not connect an Ethernet switch or hub in-betweens.
C. Alive Port (or Heart-beat Port)	<b>COM3 (RS-422):</b> Used to detect if two PACs are running properly. Using the RJ-45 connector. Two PACs can be connected by using an Ethernet crossover cable

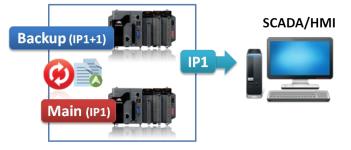
# Features of the Win-GRAF Redundancy

## 1. Higher safety:

The redundant system conducts communication through LAN1, LAN2, or COM3 (RS-422). When the Active-PAC detects hardware, software, or communication exceptions, the switching mechanism will be triggered which means the other PAC will take control of the system. So the program can work normally even if two cables are disconnected.

# 2. Unique Public IP:

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 address.



# 3. Easy maintenance:

If the redundant system is malfunctioning, the operator can power off and remove the broken one and replace a spare one without re-installing all files. The normal PAC will automatically send the Win-GRAF project and data to the new PAC.

# Notice:

- Do not shut down or dismounting the normal PAC, keep it running.
- Before powering on the PAC, adjust the rotary switch to the proper position, and connect all required communication cables or I/O modules.

# Exception:

Except for the Win-GRAF project, if there are other projects such as C or HMI running in the redundant system, these files need to pre-installed to the spare RPAC (or a repaired PAC) before re-installing this PAC to the redundant system.

# 4. Simplifying the programming process:

Users do not need to specify what files or data should be sent to spare PAC because the Win-GRAF redundant system will automatically handle these tasks.

# 5. Custom security mechanism:

Users can design a security mechanism in the program, for example, if Active-PAC is failed to connect to SCADA because LAN1 or RS-485 is disconnected or damaged, it will automatically reboot which means release its control authority to the other PAC.

# 6. I/O Redundancy:

Besides PAC Redundancy, users can choose the <u>iDCS-8000 system</u> to achieve I/O redundancy.

# 14.1 What Kinds of Data will Automatically Send to the Passive-PAC?

Win-GRAF redundant system will automatically send a part of data from Active-PAC to Passive-PAC.

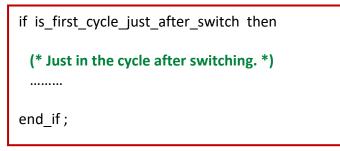
### What Kinds of Data can Backup Automatically:

- 1. The user's Win-GRAF applications.
- 2. The value of variables.
- 3. The private data of function block instance.
- 4. The PAC's RTC (Real Time Clock) time.
- 5. Retain memory.

### The most common data that cannot BackUp to the Passive-PAC Automatically?

- 1. The status of the Timer variable (Ticking or Sleeping).
- 2. Some files on the Active-PAC. For example, files stored in the "/system\_disk" or "/mnt/microSD" or non-Win-GRAF applications such as C or eLogger applications. These files should be pre-installed in the PAC before mounting them to the redundant system.
- 3. If using the COM\_OPEN() function to open the serial port, which will not automatically be opened on the Passive-PAC.
- 4. The PAC's data stored in FRAM memory cannot be backed up automatically.

For some data that unable to send to the Passive-PAC automatically, users can deal with them by programming as follows. Refer to the "Retain\_and\_timer" program in the "demo\_RDN\_2" project.



# 14.2 Redundant System (Rotary Switch: 7 & 9)

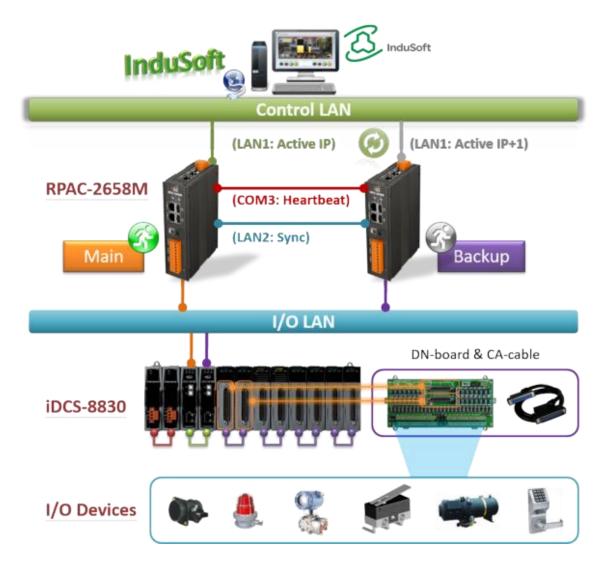
Set up the rotary switch and cables as follows:

Hardware				
		One is set to "7" (called Main-PAC)		
	Rotary Switch	One is set to "9" (called Backup-PAC)		
Two RPAC	LAN1 LAN2	2 x Ethernet cable		
TWURPAC		1 x Ethernet crossover cable		
	COM3 (RS-422)	1 x RS-422 crossover cable, also use Ethernet crossover cable.		

# 14.2.1 The Architecture of Win-GRAF Near-Field Redundant System

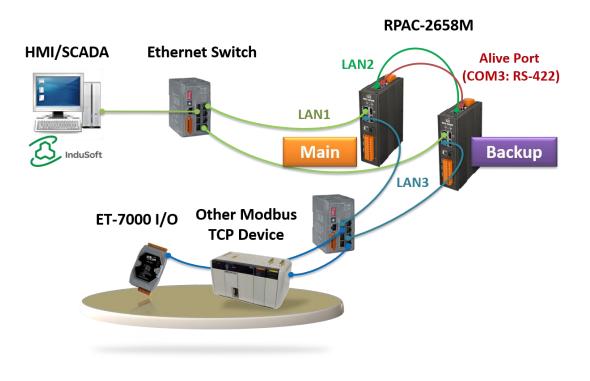
# 1. A redundant system with the use of iDCS-8830 I/O modules:

iDCS-8830 Ethernet expansion unit and redundant I/O modules.

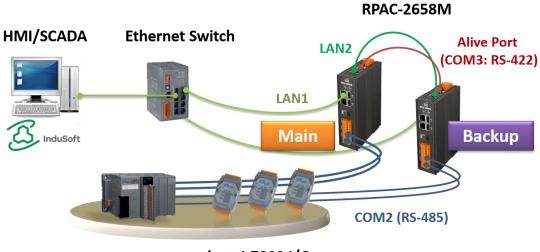


<u>Note:</u> Each pair of plugged-in redundant I/O modules with iDCS-8830 must be the same model numbers.

2. A redundant system with the use of Modbus TCP I/O modules:

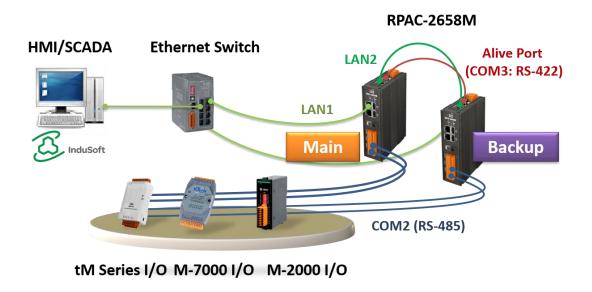


3. A redundant system with the use of DCON I/O modules:

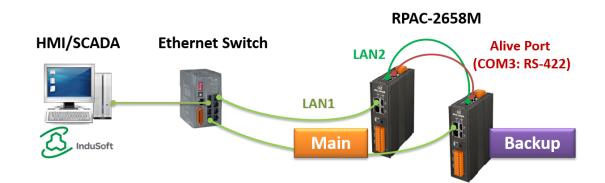


I-87K4-G + I-87KW I/O I-7000 I/O

4. Two PACs are equipped with Modbus RTU/ASCII I/O modules:

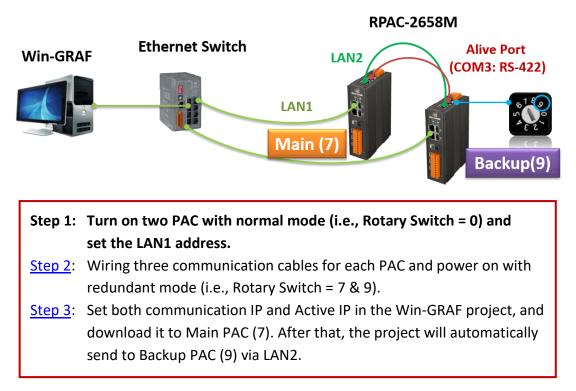


5. A redundant system without plugged-in I/O modules:



The above architecture of the I/O connection (2 to 5) can also be mixed with two or more.

# 14.2.2 Set up the Redundant System



### Step 1: Set the LAN1 Address

Conf

- A. Set the rotary switch to "0" for the redundant system and power on.
- B. Execute eSearch Utility. Click the Search Server button to search devices and click the Configuration button to set the LAN1 IP address. Refer to Section 1.3 and both the IP addresses of PC/Win-GRAF and PAC must on the same network segment to connect properly.

ø	eSearch Utilit	ty [ v1.2.6, Dec.09	, 2020 ]		- 🗆	×
File	Server To	ols				
Na	me	Alias	IP Address	Sub-net Mas	k Gateway	^
RP	AC-2658M	RPAC(7)_L1	192.168.79	.10 255.255.0.0	192.168.1.1	
	AC-2658M	RPAC[7]_L2	172.16.255		172.16.0.254	
	AC-2658M	RPAC(7)_L3	<u>10.0.255.1</u> 192.168.79	255.255.0.0	10.0.0.254	
L L	AC-2658M AC-2658M	RPAC(9)_L1 RPAC(9)_L2	172.168.75		192.168.1.1 172.16.0.254	
	AC-2658M	C[9]_L3	10.0.255	255.255.0.0	10.0.0.254	~
<		(1.)	2			>
State	Search Se	erver Configu	ration (UDP)	Web	Exit	
Configure Server (I	JDP)					
Server Name :	RPAC-265	8M				
DHCP:	0: OFF	▼ Su	b-net Mask : 🛛	55.255.0.0	Alias: RPAC(7)_	<u>_L1</u>
IP Address :	192.168.79	.10 Ga	teway: 1	92.168.1.1	MAC: 00:0d:e0	:c0:00:25
Warning!! Contact your N	etwork Admin	istrator to get cor	rect configuratio	on before any changing!	ОК	Ca

Note: To avoid IP address conflicts after rebooting PACs, LAN1 and Win-GRAF Active IP must be set to a different IP address.

# Step 2: Set up communication cables and redundant mode (Rotary switch: 7 & 9)

- A. Connect each LAN1 port of PACs to an Ethernet switch.
- B. In between, two LAN2 ports of PACs must connect to an Ethernet crossover cable.
- C. In between, two COM3 ports of PACs must connect to an RS-422 crossover cable.
- D. Set the rotary switch of PAC to "7" (called Main-PAC) and the other one to "9" (called Backup-PAC).
- E. Rebooting two PACs.

Note: In redundant mode, the PAC's LAN2 IP will automatically be set to 199.193.195.17/199.193.195.9.

## Step 3: Download the Redundant Project

### 1. Set the Win-GRAF communication IP address

For the first time to download the redundant project, e.g., demo\_rdn\_2, enter the LAN1 IP address of the Main-PAC (7). (If you are not familiar with the setting, refer to Section 2.3.4)

Communication Settings		×
T5 Runtime 192.168.79.10:502		OK Cancel
		Browse
192.168.71.17:502 192.168.79.10:502 192.168.79.11:502	The communication IP.	Help

# 2. Set the Active\_IP address

Set the Active\_IP and Mask address for the "i\_redundancy" function depends on the network environment. (If you are not familiar with the setting, refer to Chapter 4)

10: i_redundancy - Properties	
<b>Key</b> = 6	
<b>Ref</b> = 16#3	
<b>Active_IP</b> = 192.168.79.37	
<b>Passive</b> IP = auto	
<b>Mask</b> = 255.255.0.0	
Gateway_IF = disabled	
Reserved0 = 0	
Reserved1 = 0	
Reserved2 = 0	
Reserved3 = 0	
Reserved4 = 0	

# 3. Download the Win-GRAF project

Click the "On Line" button ( ) to connect and download the redundancy project to the Main-PAC. After completing the download, the PAC's LAN1 IP address will automatically be set to the Active\_IP address (e.g., 192.168.79.37) and the LAN1 IP address of the Backup-PAC(9) will automatically be set to Active IP + 1 (e.g., 192.168.79.38).

## 4. Change the Win-GRAF communication IP address to the Active\_IP address

Right now, Win-GRAF will show "Communication error" because the current IP address of the Active-PAC is automatically set to the Active\_IP address. Click the "On Line" button again to stop the connection between the Win-GARF and the PAC.

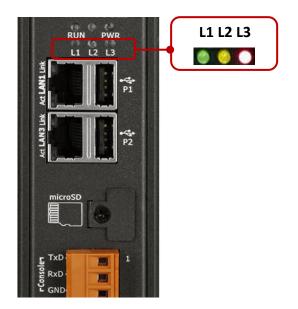
🕒 🕍 🥂 Communication error		
Stop the connection.		

Next, change the Win-GRAF communication IP to the Active\_IP address so that this project will always download to the Active-PAC whenever the user wants to update the program.

Communication Settings	×
T5 Runtime	OK Cancel
192.168.79.10:502 The communication IP.	Browse
192.168.79.37:502	Help

**Note:** Before updating the program of the Active-PAC, power off the Passive-PAC. After completing the update, power on the Passive-PAC to ensure that the redundant system is work properly.

Besides, the L3 LED is lit in red which means the PAC is Active-PAC.



# 14.2.3 Test the Redundant Project (demo\_rdn\_2)

Refer to Section 14.4 for the description of demo projects.

# 1. View the status of redundant PACs

After Win-GRAF connects to the PAC, click "NewSpy1" to open the spy list window and check the current status of the redundant system.

<u>F</u> ile <u>E</u> dit <u>V</u> iew Insert Project <u>T</u> ools <u>W</u> indow <u>H</u> elp						
注						
Workspace	Workspace NewSpy1.spl					
🖃 🗝 🕨 demo_rdn_2 [RUN]		H Name			Description	
🗄 🔤 Exception programs		Hour1	12			
🖨 🔤 Programs		Minute1	35	[		
PAC_Time	₽Ļ	Second1	21		The Main-PAC is Active-PAC now.	
BDN_control		is_Main_Active	TRUE	r ,		
🔤 📷 Retain_and_timer		is_Backup_Active	FALSE			
📥 🔤 Watch (for debugging)		is_Main_ready	TRUE		$\diamond$ Main-PAC and Backup-PAC are	
Soft Scope		is_Backup_ready	TRUE		ready and LAN1 IP addresses are	
📰 Initial values		is_first_cycle_just_after_switch	FALSE		,	
NewSpy1		is_Main_LAN1_ok	TRUE		function normally.	
		is_Backup_LAN1_ok	TRUE		$\diamond$ Alive Port (RS-422) are function	
🚽 🖇 😏 🖓 🔤 🖓 🦻		is_Alive_port_ok	TRUE		normally.	
🔤 🏠 Variables		is_Passive_ready	TRUE		,	
E Types		is_Active_LAN1_ok	TRUE		$\diamond$ Passive-PAC is ready.	
		is_Passive_LAN1_ok	TRUE		$\diamond$ Both LAN1 IP addresses of	
		DINT_1	0			
		DINT_2	0		Active-PAC and Passive-PAC are	
		REAL_1	0.0		function normally.	
		REAL_2	0.0		,	
		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	

# Note:

- 1) Before switching the control authority, make sure the Passive-PAC is ready (i.e., "is\_Passive\_ready" = TRUE).
- 2) Users can also click the "Redundancy" button ( 🍄 ) to manually switch the control authority.

🚰 RUN 0000 🎾 🁬	;≓ u ¥(‡)
Redundancy	×
O Redundancy is not enabled in the runtime	Refresh Close
<ul> <li>Running as active runtime</li> <li>Passive runtime is alive</li> </ul>	Set as passive
	C is Active-PAC right now.
O Running as passive runtime	

# 2. Set the value of variables

- 1) Assign a value to DINT\_1, DINT\_2, REAL\_1, and REAL \_2 variables.
- 2) Set the To\_tick\_TMR\_1 and To\_tick\_TMR\_2 to "TRUE" (it will automatically reset to FALSE) to start the "TMR\_1" and "TMR\_2" ticking.

	NewSpy1.spl					
	Name		Value	Description		
	is_Active_LAN1_ok		TRUE			*
	is_Passive_LAN1_ol	<	TRUE			
₽Ļ	DINT_1		9	Setup as Retain var	iable in the program "Retain_and_timer"	
	DINT_2		1234 🚽	Setup as Retain var	iable in the program "Retain_and_timer"	
	REAL_1		22.299999	Entera value	able in the program "Retain_and_timer"	
	REAL_2		33.5	Entera value	able in the program "Retain_and_timer"	
	TMR_1		t#1m10s26ms		-	
	TMR_1_last_state		TRUE	TRUE: ticking , FAL	SE: sleep	
	To_tick_TMR_1		FALSE	Set TRUE to start tid	cking timer1	
	To_stop_TMR_1	Calling	FALSE	Set TRUE to stop th	ne ticking of timer1	Ξ
	TMR_2	Set timer	t#36s996ms			
	TMR_2_last_state_ to "TRUE"		TRUE	TRUE: ticking , FAL	SE: sleep	
	To_tick_TMR_2		FALSE	Set TRUE to start tio	cking timer2	
	To_stop_TMR_2		FALSE	Set TRUE to stop th	ne ticking of timer2	Ŧ

- 3. Test the redundant system (refer to the "RDN\_control" program)
  - 1) Make sure the Passive-PAC is ready (i.e., is\_Passive\_Ready is TRUE).
  - Remove the LAN1 cable of the Active-PAC and the control will pass to the other PAC. In this demo
    program (RDN\_control), the PAC will automatically reboot after a specific time if the LAN1 cable is
    disconnected.
  - 3) After rebooting, the Active-PAC is Backup PAC, all values still exist and timers are still ticking.

	NewSpy1.spl			
	Name	Value	The Backup-PAC is A	ctive-PAC now.
	is_Main_Active	FALSE		
	is_Backup_Active	TRUE	NewSpy1.spl	
₽Ļ	is_Main_ready	TRUE		Value
	is_Backup_ready	TRUE		9
	is_first_cycle_just_after_switch	FALSE	DINT_1 DINT_2	1234
	is_Main_LAN1_ok	FALSE	H+ BEAL 1	22,299999
	is_Backup_LAN1_ok	TRUE	REAL_2	33.5
	is_Alive_port_ok	TRUE	TMB 1	t#29m38s510ms
	is_Passive_ready	TRUE	TMR_1_last_state	TRUE
	is_Active_LAN1_ok	TRUE	To_tick_TMR_1	FALSE
	is_Passive_LAN1_ok	FALSE	To_stop_TMR_1	FALSE
			TMR_2	t#29m5s480ms
			TMR_2_last_state	TRUE
			To_tick_TMR_2	FALSE
			To_stop_TMR_2	FALSE

### Note:

Plugin the LAN1 cable on the Main-PAC (Passive), the statuses of "is\_Main\_LAN1\_ok" and "is\_Passive\_LAN1\_ok" will become "TRUE".

# 14.3 Description of Win-GRAF Demo Projects

Download <u>the demo program</u> on the website (demo\_RDN\_1.zip, demo\_RDN\_2.zip, demo\_RDN\_3.zip, or and "demo\_RDN\_4.zip"). Click the menu commands **File - Add Existing Project - From Zip...** in the Workbench to open the project.

Project Name	Description		
demo_RDN_1	Two PACs are connected with three DCON I/O modules via COM2	<u>Test</u>	
demo_RDN_2	Using two PACs without connecting any I/O module.	<u>Test</u> , <u>Program</u>	
demo_RDN_3	Two PACs are connected with an ET-7050 (Modbus TCP I/O module) via LAN3 through an Ethernet switch.	<u>Test</u>	
demo_RDN_4	Two PACs are connected with an iDCS-8830 via LAN3 through an Ethernet switch. iDCS-8830 supports redundant I/O modules.	<u>Test</u> , <u>Program</u>	

# 14.3.1 [Important] "I/O Board" Settings (i\_redundancy and i\_redundancy\_rs485)

# Used for demo\_RDN\_1, demo\_RDN\_2, demo\_RDN\_3, and demo\_RDN\_4:

For the redundant system can work normally, the user must add the "i\_redundancy" function in the "I/O Boards" window (refer to Chapter4) and set the Active\_IP and Mask.

I/O Boards		×		
0 1 2 3 4 5 6 7 8 9 10 <u>i</u> redundancy 11 12 13 14 15 16	10: i_redundancy - Prope Key = 6 Ref = 16#3 Active_IP = 192.168.79.37 Passive_IP = auto Mask = 255.255.0.0 Gateway_IP = disabled Reserved0 = 0 Reserved1 = 0 Reserved2 = 0 Reserved3 = 0 Reserved4 = 0	Select rties <u>Note</u> : DO NOT set the	e last digit value of the "Active_IP 55. <b>It should be 1 to 253.</b>	х )"
	i_redundancy Enable Redundancy in the P	AC.		< >
	<		>	

# Description of "i\_redundancy" I/O board:

The function is used to display the current status of the Win-GRAF redundant system.

- Active\_IP: This IP address is public for the redundant system to communicate with HMI or SCADA.
   DO NOT set the last digit value of the "Active\_IP" IP address as 0 or 254 or 255.
   It should be from 1 to 253.
- Passive\_IP: "Auto", the LAN1 IP address of the Passive-PAC will automatically be set as Active\_IP +1. E.g., if the "Active\_IP" is set as "192.168.71.37", the "Passive\_IP" will be automatically set as "192.168.71.38".
- Mask: The common settings are either 255.255.255.0 or 255.255.0.0 depending on the network environment.
- <u>Note:</u> After adding 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 status of the redundant system.

Variables									∎⊿×
🍸 Name	Туре	e D	im. Attrib.	Syb.	Init value	User	Tag	Description	
🖃 👼 %IX10 - i redundancy									~
%IX10.0=is_Main_Active	BOC	)L	Input						
%IV10 1-te Backup Active	BOO	)L	Input						
is_Backup_Active	BOC	)L	Input						
%IX10.3=is_Backup_ready	BOC	1	Input				_		
%IX10.4=is_first_cycle_just_afte	Double-clie	ck it te	o modify	the	/ariable	name			
%IV10.5—io. Main I AN1. ak				the t		manne	, I		
%IX10.6=is_Backup_LAN1_ok	and press	the Er	iter key.						
%IX10.7=is_Alive_port_ok	ROC		Input						
%IX10.8=is_Passive_ready	BOC	DL	Input						
	- Doc	SI							
%IX10.9=is_Active_LAN1_ok	BOO	JL	Input						
%IX10.9=is_Active_LAN1_ok %IX10.10=is_Passive_LAN1_ok	BOG		Input Input						
		)L							~
%IX10.10=is_Passive_LAN1_ok	BOO	)L	Input						~

Channel	Variable Name	Description
Ch.0	is_Main_Active	Is the Main-PAC active now?
		TRUE: Active PAC.
		FALSE: Passive PAC.
Ch.1	is_Backup_Active	Is the Backup-PAC active now?
		TRUE: Active PAC.
		FALSE: Passive PAC.
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-PAC and
		Backup-PAC is broken.
		(2) The Main-PAC is dead or crashed.
		(3) The rotary switch of the Main-PAC is not set at 7 (or 6).

Channel	Variable Name	Description
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 the Main and
		Backup-PAC is broken.
		(2) The Main-PAC is dead or crashed.
		(3) The rotary switch of the Main-PAC is not set at 9
		(or 8).
Ch.4	is_first_cycle_just_after_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.
Ch.7	is_Alive_port_ok	Is the communication status of Alive Port ok?
		TRUE: OK.
		FALSE: Fail or the Passive-PAC is dead or crashed.
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 the Main and
		Backup-PAC is broken.
		(2) The Passive-PAC is dead or crashed.
		(3) The rotary switch setting of the Passive-PAC is incorrect.
		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.

# $\diamond$ Used for demo\_RDN\_1:

In this case, using the COM3 (RS-485) of PAC to connect DCON I/O modules. Besides "i\_redundancy", also "**DCON**" must be added in the I/O Boards window. Moreover, the "**i\_redundancy\_rs485**" function is used to check if the RS-485 port of Passive-PAC can receive data normally.

I/O Boards	×	
0 1 2 3 4 5 6 7 9 DCON 10 redundancy 11 i_redundancy_rs485 12 13 14	9: 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 Reserved2 = 0 Reserved3 = 0	x
15 16	DCON Enable one serial port (RS-485) to connect remote DCON I/Os (I-7000 series modules ,	^
		۲
	< >	

# Notice:

- 1. The "i\_redundancy\_rs485" must use with the "i\_redundancy" or it doesn't work.
- 2. The "i\_redundancy\_rs485" only enables RS-485 ports of Passive-PAC to receive data but not to send data.

IIII I/O Boards		$\times$	
0 1 2	11: i_redundancy_rs48 Key = 6 Paf = 16#20	35 - Properties	×
3 4 5	Ch00 Port No = 2 Ch00 Timeout = 30 Ch01 Port No = 0	<u>}</u>	
6 7 8	Ch01_Timeout = 30 Ch02_Port_No = 0 Ch02_Timeout = 30 Ch02_Timeout = 30	Chxx_Port_No: Using COM2 of Passive-PAC (0: Disable)	
9 DCON 10 i_redundancy 11 i_redundancy_rs485	Ch03_Port_No = 0 Ch03_Timeout = 30 Ch04_Port_No = 0 Ch04_Timeout = 30	Chxx_Timeout: Set timeout as 30 seconds.	
12 13 14	Ch05_Port_No = 0 Ch05_Timeout = 30 Ch06_Port_No = 0 Ch06_Timeout = 30	Note: If no data is received after the time-out, the	
15 16	i_redundancy_rs485	status of channel (0-15) will be displayed as "FALSE" in the "Variables" window.	
	<	>	

# 14.3.2 Declaring Variables (demo\_RDN\_2)

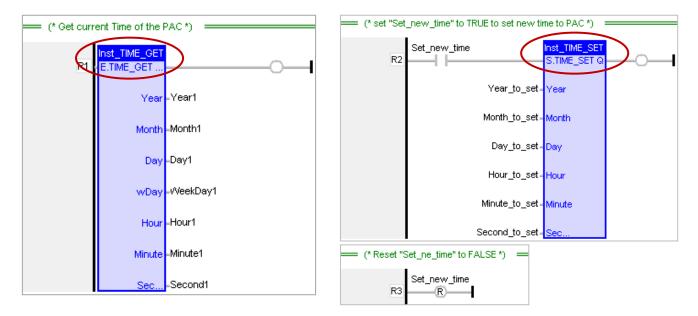
Users can view or add variables in the "Variable" window.

Name	Data Type	Description		
Used in the "PAC_Time"	program			
Year1				
Month1				
Day1				
WeekDay1	DINT	Used in the "PAC_Time" program: They are used to get the PAC's system time.		
Hour1		They are used to get the TAE's system time.		
Minute1				
Second1				
Set_new_time	BOOL	Set it as "TRUE" to set up a new system time.		
Year_to_set				
Month_to_set				
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				
Second_to_set				
Used in the "Retain_and	_timer" progra	am		
DINT_1	DINT			
DINT_2		Used in the "Retain_and_timer" program: Set them as retain variables.		
REAL_1	REAL			
REAL_2	REAL			
TMR_1	TIME	Timer		
TMR_2	TIVE			
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		It used to return the Retain status.		
TMR_1_last_state		TRUE: Ticking; FALSE: Sleeping.		
TMR_2_last_state		TRUE: Ticking; FALSE: Sleeping.		
To_tick_TMR_1	BOOL	Set it as TRUE to start TIMER1.		
To_tick_TMR_2		Set it as TRUE to start TIMER2.		
To_stop_TMR_1		Set it as TRUE to stop TIMER1.		
To_stop_TMR_2		Set it as TRUE to stop TIMER2.		

## 14.3.3 Introduction of the "demo\_RDN\_2" Project

### LD Program – "PAC\_Time"

It used to get/set the system time of PAC.



### LD Program – "RDN\_control"

If the Passive-PAC's communication and health status is better than Active-PAC when there is a problem with Active PAC, the control authority will be changed to the Passive-PAC immediately. In this example program, PAC will automatically reboot after detecting LAN1 disconnection for more than 1 second.

RDN	RDN_control										
-11-+											
+-H-	Switch to Passive PAC if Passive is	ready and its LAN1 is	ok however Active L	AN1 communication	has problem 🛛 🛁						
Ψ	is_Active_LAN1_ok	Inst_TON IN TON Q		is_Passive_ready	is_Passive_LAN1_ok	En PAC REBOOT Q					
<b>–</b> *	T#1s	PT ET	_								

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_1, 3); (* retain a REAL variable *)
```

```
(* 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;
```

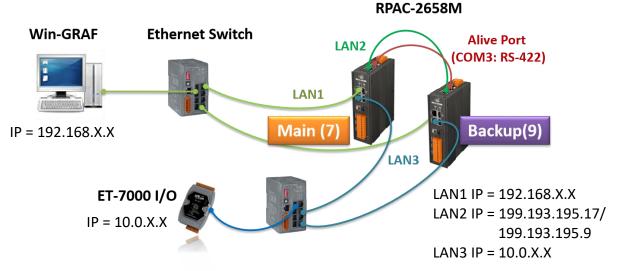
## 14.4 The Description for Other Redundant Projects

## 14.4.1 Test the Redundant Project (demo\_rdn\_3, demo\_rdn\_1)

## Description (demo\_rdn\_3):

Using both the **LAN3** of two PACs to connect an <u>ET-7050</u> (Modbus TCP I/O module, 12 DI, 6 DO) through one Ethernet switch.

### Hardware Wiring:

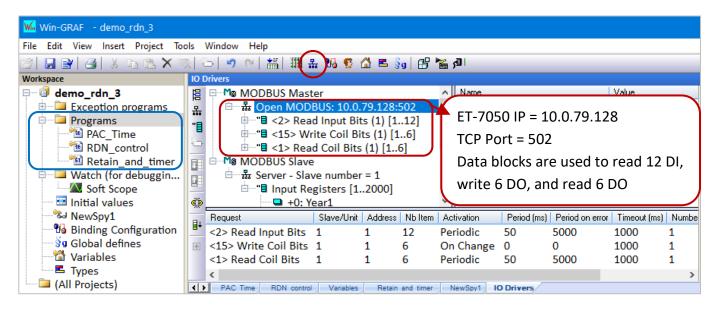


### 1. Configure the ET-7000 module

Refer to the ET-7000 manual to set the IP address and I/O settings (also refer to Section 5.2.1). Manual: https://www.icpdas.com/en/download/show.php?num=2217&model=ET-7050

### 2. View settings of the Win-GRAF project

In the "I/O Drivers" window, set the PAC as Modbus TCP Master to connect an ET-7050 module (Modbus TCP Slave, Addr. = 1) and create some data blocks to read/write DI and DO data (refer to Section 5.2 for details about the setting).



In the Workspace, double-click the program name to view the program.

## 3. Download the Win-GRAF project ("demo\_rdn\_3")

Configure the proper Win-GRAF communication IP, Active\_IP, Mask, and ET-7000 IP addresses, and then download the project to the PAC.

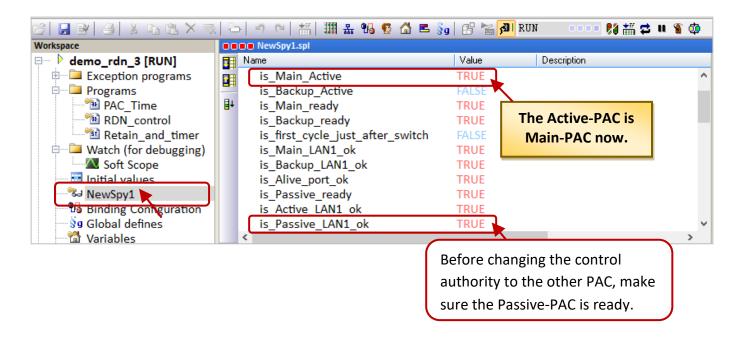
#### Note:

- Using eSearch Utility to check the IP address of devices.
- In the example, the IP address of LAN3 and PET-7050 must on the same network segment.
- In redundant mode, the LAN2 IP address of two PACs will automatically be set to 199.193.195.17 and 199.193.195.9.
- Power off the Passive-PAC before **updating the project** to the Active-PAC. After completing the update, power on the Passive-PAC to ensure that the redundant system is work properly.

	🥩 eSearch Utili	– 🗆 X			
	File Server Too				
	Name	Alias	IP Address	Sub-net Mask	Gateway 🔨
Γ	PET-7050	EtherIO	10.0.79.128	255.255.0.0	10.0.0.254
	WP8000	WP8000	192.168.1.38	255.255.0.0	192.168.1.1
1	RPAC-2658M	RPAC(7)_L1	192.168.79.37	255.255.0.0	192.168.0.254
	RPAC-2658M	RPAC(7)_L2	199.193.195.17	255.255.255.0	172.16.0.254
	RPAC-2658M	RPAC(7)_L3	10.0.79.10	255.255.0.0	10.0.0.254
	RPAC-2658M	RPAC(9)_L1	192.168.79.38	255.255.0.0	192.168.0.254
	RPAC-2658M	RPAC(9)_L2	199.193.195.9	255.255.255.0	172.16.0.254
	RPAC-2658M	RPAC(9)_L3	10.0.79.11	255.255.0.0	10.0.0.254 🌙
	<			W-F   [	Exit
	Search Se	rver configura	ation (UDP)	Web	
	Status				11

### 4. View the status of redundant PACs

Click "NewSpy1" to open the spy list, now the Main-PAC is the Active-PAC.



## 5. Set the value of variables

- 1) Assign values to DINT\_1, DINT\_2, REAL\_1, and REAL \_2 variables.
- 2) Set the To\_tick\_TMR\_1 to "TRUE" to start the "TMR\_1" ticking.
- 3) Set the ET-7050\_Do\_x to "TRUE" and the ET-7050\_DO\_x\_ReadBack will return "TRUE".
- If the Ethernet cable of ET-7050 is removed, the "ET-7050\_COM\_err" will return a non-zero value to indicate the communication error.

61	i 🖩 🏭 🏭 % 🥵 🏠 🛤 💡 🕯	😗 🕍 🚮 RUN	🎀 🛗 📫 🦉 🏟
	NewSpy1.spl		
DE N	ame	Value	Description
	is_Passive_LAN1_ok	TRUE	^
	DINT_1	1234	Setup as Retain variable in the program "Retain_and_timer"
₽	DINT_2	6006	Setup as Retain variable in the program "Retain_and_timer"
	REAL_1	2356.75	Setup as Retain variable in the program "Retain_and_timer"
	REAL_2	6592.5	Setup as Retain variable in the program "Retain_and_timer"
	TMR_1	t#12s378ms	
	To_tick_TMR_1	FALSE	Set TRUE to start ticking timer1
	To_stop_TMR_1	FALSE	Set TRUE to stop the ticking of timer1
	ET7050_COM_err	0	0: No error (comm. ok) , Communication error of the ET-7050
	ET7050_DO_0_ReadBack	TRUE	If ET-7050_DOx is set to TRUE, ET-7050_DOx_ReadBack will return TRUE.
	ET7050_DO_1_ReadBack	TRUE	
	ET7050_DO_2_ReadBack	TRUE	
	ET7050_DO_3_ReadBack	FALSE	
	ET7050_DO_4_ReadBack	FALSE	
	ET7050_DO_5_ReadBack	FALSE	
	ET7050_DO_0	TRUE	DOx of the ET-7050
	ET7050_DO_1	TRUE	
	ET7050_DO_2	TRUE	)
	ET7050_DO_3	FALSE	
	ET7050_DO_4	FALSE	
	ET7050_DO_5	FALSE FALSE	INx of the ET-7050
	ET7050_DI_00	FALSE	INX of the EI-7050
	ET7050_DI_01 ET7050 DI 02	FALSE	
	ET7050_DI_02 ET7050_DI_03	FALSE	
	ET7050_DI_03	FALSE	
	ET7050_DI_04	FALSE	
	ET7050_DI_06	FALSE	
	ET7050 DI 07	FALSE	
	ET7050_DI_08	FALSE	
	ET7050_DI_09	FALSE	
	ET7050_DI_10	FALSE	
	ET7050_DI_10	FALSE	
<	2		×
< >	PAC Time RDN control Variables	Retain and timer Ne	wSpy1/ 10 Drivers
الشاغي	A HER CONTRACT A VEHICLES A		

#### 6. Test the redundant system

- 1) Make sure the Passive-PAC is ready (i.e., is\_Passive\_Ready is TRUE).
- Remove the LAN1 cable of the Active-PAC and the control will pass to the other PAC. In this demo
  program (RDN\_control), the PAC will automatically reboot after a specific time if the LAN1 cable is
  disconnected.

	NewSpy1.spl			
	Name	Value	Description	
	is_Main_Active	FALSE		^
	is_Backup_Active	TRUE	The Active-PAC is	
∎+	is_Main_ready	FALSE	Backup-PAC now.	
	is_Backup_ready	TRUE	· · · · · · · · · · · · · · · · · · ·	
	is_first_cycle_just_after_switch	FALSE		

	NewSpy1.spl		
	Name	Value	Description
	is_Main_LAN1_ok	FALSE	
	is_Backup_LAN1_ok	TRUE	Plug in the LAN1 of Main-PAC (Passive)
∎+	is_Alive_port_ok	FALSE	and the value will become TRUE.
	is_Passive_ready	FALSE	
	is Active LAN1 ok	TRUE	
ļĻ	is_Passive_LAN1_ok	FALSE	
	DINT_1	1234	Setup as Retain variable in
	DINT_2	6006	After the control is changed, the variable
	REAL_1	2356.75	data still exist and the timer is ticking.
	REAL_2	6592.5	
	TMR_1	t#18m32s4	
	To_tick_TMR_1	FALSE	Set TRUE to start ticking tim
	To stop TMR 1	FALSE 0	Set TRUE to stop the ticking
	ET7050_COM_err ET7050 DO 0 ReadBack	TRUE	0: No error (comm. ok) , Cor If ET-7050 DOx is set to TR
	ET7050_DO_0_ReadBack ET7050_DO_1_ReadBack	TRUE	IT ET-7050_DOX IS SET TO TR
	ET7050_DO_1_ReadBack	TRUE	
	ET7050_DO_2_ReadBack	FALSE	<b>ET-7050_COM_err</b> : A non-zero value
	ET7050_DO_3_ReadBack	FALSE	indicates the communication error.
	ET7050_DO_5_ReadBack	FALSE	Set the <b>ET-7050_Dox</b> to TRUE and the
	ET7050 DO 0	TRUE	_
	ET7050 DO 1	TRUE	ET-7050_DOx_ReadBack will return TRUE
	ET7050 DO 2	TRUE	If the communication is OK
	ET7050 DO 3	FALSE	
	ET7050 DO 4	FALSE	
	ET7050 DO 5	FALSE	
	ET7050_DI_00	FALSE	INx of the ET-7050
	ET7050_DI_01	FALSE	
	ET7050_DI_02	FALSE	
	ET7050_DI_03	FALSE	
	ET7050_DI_04	FALSE	
	ET7050_DI_05	FALSE	
	ET7050_DI_06	FALSE	
	ET7050_DI_07	FALSE	
	ET7050_DI_08	FALSE	
	ET7050_DI_09	FALSE	
	ET7050_DI_10	FALSE	
	ET7050_DI_11	FALSE	✓
	<		>
<b>&lt; &gt;</b>	PAC Time RDN control Variables	Retain and timer	NewSpy1 IO Drivers

## Test the redundant project (demo rdn 1)

#### **Description:**

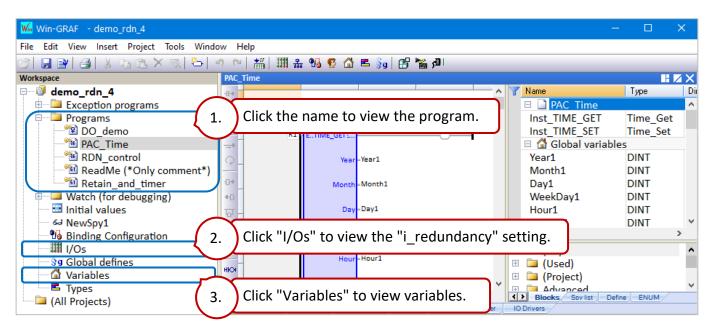
Using both the **COM2 (RS-485)** of two PAC to connect three DCON I/O modules (I-87064, I-87018ZW, I-7065).

- 1. Before connecting two PACs with remote DCON I/O modules, refer to <u>Chapter 8</u> to configure each module by using DCON Utility and set the "DCON" I/O board.
- 2. In the Win-GRAF, view programs and set "i\_redundancy" and "i\_redundancy\_rs485" I/O boards. (Refer to Section 14.3.1 I/O Boards Settings).
- 3. After downloading the Win-GRAF project ("demo\_rdn\_1"), refer to step5 and step6 of "demo\_RDN\_3" as noted before to test.

## 14.4.2 Introduction of the "demo\_RDN\_4" Project

### **Description:**

Two PACs are connected with an iDCS-8830 Redundancy I/O unit via LAN3 through an Ethernet switch.



iDCS-8830 equips redundant power inputs, redundant Ethernet communication modules, and eight I/O slots. In the example, a set of redundant DI modules (i.e., F-8040 in slots 0 and 1) and a set of redundant DO modules (i.e., F-8041 in slots 2 and 3) are used.

### The "Modbus Master" Setting

Click the "Open Fieldbus Configuration" button to open the "IO Drivers" window. In this example, the Modbus Master is enabled to connect two Modbus TCP Slaves (i.e., communication modules of iDCS-8830) to read/write data or the communication status.

CI												
	Drivers *	aim e	<b>,</b>				_					
眉	Model Model Master						~	Name			Value	
몳	e 뮮 Open MODBUS: 10	0.0.79.200	:502	$\rightarrow$	通訊樽	<b>≒</b> ⊀⊟ 1		Mode			Open MODE	BUS
*8	≝ <sup>™</sup> ∎ <2> Read Input	t Bits (1) [1	32] - F	Read D/				Addres	55		10.0.79.200	
217	🕮 📲 <15> Write Coi							Port			502	
-	⊞—"≣ <4> Read Input	t Registers	(1) [513	3521] -	IDCS-8830 s	tatus		Recon	nect after e	error		<ul><li>✓</li></ul>
	🗄 📲 <4> Read Input			7664] -	I/O status				diagnostic			
	🖹 🖻 🏪 Open MODBUS: 10	0.0.79.201	:502	+		±/=				equests (ms)	10	
	≝ <2> Read Input		-		通訊樽	Q組 2		Disabl	ed			
ġ,þ	🕮 📲 <15> Write Coi		-		$\square$							
∎∔	⊞*∎ <4> Read Input	_		-		tatus						
	⊞*≣ <4> Read Input	t Registers	(1) [577	7664] -	I/O status							
•	MoDBUS Slave											
	🖮 🚠 Server - Slave num											
	⊡*∎ Input Registers	5 [12000]					~					
	Request	Slave/Unit	Address	Nb Item	Activation	Period (ms)	Period	on error	Timeout (ms)	Number of trials	Description	
	<2> Read Input Bits	1	1	32	Periodic	0	3000		1000	1	Read D/I	
	<15> Write Coil Bits	1	65	32	Periodic	0	3000		1000	1	Control D/O	
	<4> Read Input Registers	1	513	9	Periodic	0	3000	:	1000	1	iDCS-8830 st	tatus
	<4> Read Input Registers	1	577	88	Periodic	0	3000		1000	1	I/O status	
	<											>

**Note:** the IP addresses of LAN3 and communication modules must on the same network segment.

Read DI (F-8040 in slot 0 and 1):

To read **32** DI data start from address **1** and send the request continuously. It will be sent after **3** seconds if an exception occurs. No response for **1** second can be regarded as abnormal.

10 0	rivers *	
目	□Mo MODBUS Master	^
品	🖨 🚠 Open MODBUS: 10.0.79.	200:502
	🕀 🐨 🗄 <2> Read Input Bits (	1) [132] - Read D/I
*8	🗉 📲 <15> Write Coil Bits (	1) [6596] - Control D/O
8	⊞ ••••∎ <4> Read Input Regis	ters (1) [513521] - iDCS-8830 status Double-click
	⊞—"∎ <4> Read Input Regis	ters (1) [577664] - I/O status
	🖻 🚠 Open MODBUS: 10.0.79.	201:502 🗸
_	Symbol	Operation Offset Mask Storage
¢.	iDCS8830_slot01_F8040_DI[0]	Data exchange 0 FFFF Default
≣∔	iDCS8830_slot01_F8040_DI[1]	Data exchange 1 FFFF Default
_	iDC\$8830_slot01_F8040_DI[2]	Data exchange 2 FFFF Defoult
÷	iDC\$8830_slot01_F8040_DI[3]	Data excha MODBUS Master Request
	iDC\$8830_slot01_F8040_DI[4]	Data excha
	iDC\$8830_slot01_F8040_DI[5]	Data excha
	iDC\$8830_slot01_F8040_DI[6]	Data excha Description: Read D/I
	iDC\$8830_slot01_F8040_DI[7]	Data excha Cancel Cancel
	iDCS8830_slot01_F8040_DI[8]	Data excha
	iDC\$8830_slot01_F8040_DI[9]	Data excha
	iDC\$8830_slot01_F8040_DI[10]	Data excha
	iDCS8830_slot01_F8040_DI[11]	Data excha
	iDCS8830_slot01_F8040_DI[12]	Data excha
	iDCS8830_slot01_F8040_DI[13]	Data excha
	iDCS8830_slot01_F8040_DI[14]	Data excha
	iDC\$8830_slot01_F8040_DI[15]	Data excha
	iDC\$8830_slot01_F8040_DI[16]	Data excha Base address: 1
	iDCS8830_slot01_F8040_DI[17]	Data excha Nb items: 32
	iDCS8830_slot01_F8040_DI[18]	Data excha
	iDCS8830_slot01_F8040_DI[19]	Data excha Activation
	iDCS8830_slot01_F8040_DI[20]	Data excha
	iDCS8830_slot01_F8040_DI[21]	
	iDCS8830_slot01_F8040_DI[22]	Data excha On call (on error)
	iDCS8830_slot01_F8040_DI[23]	Data excha O On change
	iDCS8830_slot01_F8040_DI[24]	Data excha
	iDCS8830_slot01_F8040_DI[25]	Data excha
	iDCS8830_slot01_F8040_DI[26]	
	iDCS8830_slot01_F8040_DI[27]	Data excha No trials: 1
	iDCS8830_slot01_F8040_DI[28]	
	iDCS8830_slot01_F8040_DI[29]	Data exchange 29 FFFF Default
	iDCS8830_slot01_F8040_DI[30]	Data exchange 30 FFFF Default
	iDC\$8830_slot01_F8040_DI[31]	Data exchange 31 FFFF Default
	<	

<u>Note:</u> iDCS-8830 Ethernet redundant I/O unit is used in this example, refer to Section 4.1.1 of the <u>FCM-MTCP software manual</u> to set the "Base address".

**<u>Read DO</u>** (F-8041 in slot 2 and 3):

To read **32** DI data start from address **65** and send the request continuously. It will be sent after **3** seconds if an exception occurs. No response for **1** second can be regarded as abnormal.

10 0	rivers *								
眉	🗉 Modbus Master				~				
	品 Open MODBUS: 10.0.79.200:502								
	** * - * Read Input Pite (1) [1, 22] Boad D/I								
	B								
ъ		ers (1) [513521] -	iDCS-8	830 status		_			
					Double-clic	k			
	□ 🖶 🖧 Open MODBUS: 10.0.79.2	01:502							
_	Symbol	Operation	Offset	Mask	Storage				
ġ,	iDCS8830 slot23 F8041 DO[0]	Data exchange	0	FFFF	Detault				
∎∔	iDCS8830_slot23_F8041_DO[1]	Data exchange	1	FFFF	Default				
	iDCS8830_slot23_F8041_DO[2]	Data exchange	2	FFFF	Default				
÷	iDCS8830_slot23_F8041_DO[3]	Data exchange	3	FFFF	Default				
	iDCS8830_slot23_F8041_DO[4]	Data exchange	4	VODBUG	laster Request				
	iDCS8830_slot23_F8041_DO[5]	Data exchange	5	MODBUS I	nasier Kequesi				
	iDCS8830_slot23_F8041_DO[6]	Data exchange	6	_Request -					
	iDCS8830_slot23_F8041_DO[7]	Data exchange	7	Descripti	ion: Control D	0			
	iDCS8830_slot23_F8041_DO[8]	Data exchange	8			_			
	iDCS8830_slot23_F8041_DO[9]	Data exchange	9	<u>S</u> lave/Un	nit: 1				
	iDC\$8830_slot23_F8041_DO[10]	Data exchange	10						
	iDC\$8830_slot23_F8041_DO[11]	Data exchange	11	-MODBUS F	Request				
	iDC\$8830_slot23_F8041_DO[12]	Data exchange	12		rite single coil bit		~		
	iDC\$8830_slot23_F8041_DO[13]	Data exchange	13		rite single holding r	egister			
	iDC\$8830_slot23_F8041_DO[14]	Data exchange	14	<15> W	Vrite Coil Bits		~		
	iDCS8830_slot23_F8041_DO[15]	Data exchange	15						
	iDCS8830_slot23_F8041_DO[16]	Data exchange	16	Data block	<				
	iDCS8830_slot23_F8041_DO[17]	Data exchange	17	Base <u>a</u> do	dress: 65				
	iDCS8830_slot23_F8041_DO[18]	Data exchange	18	Mile Street		=			
	iDCS8830_slot23_F8041_DO[19]	Data exchange	19	<u>N</u> b items	32		J		
	iDCS8830_slot23_F8041_DO[20]	Data exchange	20	Activation					
	iDCS8830_slot23_F8041_DO[21]	Data exchange	21						
	iDCS8830_slot23_F8041_DO[22]	Data exchange	22	• Perior		ms	3000		
	iDCS8830_slot23_F8041_DO[23]	Data exchange	23	🔘 On ca	-		(on error)		
	iDCS8830_slot23_F8041_DO[24]	Data exchange	24	🔘 On ch	nange				
	iDCS8830_slot23_F8041_DO[25]	Data exchange	25	Misc.					
	iDCS8830_slot23_F8041_DO[26]	Data exchange	26		1000	ms			
	iDCS8830_slot23_F8041_DO[27]	Data exchange	27	<u>T</u> imeout:	1000	ins			
	iDCS8830_slot23_F8041_DO[28]	Data exchange	28	Nb trials	. 1				
	iDCS8830_slot23_F8041_DO[29]	Data exchange	29		·				
	iDCS8830_slot23_F8041_DO[30]	Data exchange	30	FFFF	Default				
	iDCS8830_slot23_F8041_DO[31]	Data exchange	31	FFFF	Default				
	<								

**<u>Note:</u>** Refer to Section 4.1.2 of the <u>FCM-MTCP software manual</u> to set the "Base address".

#### Read the status of iDCS-8830:

- Set the Operation field of "iDCS8830\_LAN1\_error\_status" to "Error report" which means to return an error code when fails to read data and be reset to 0 when a successful read. Also, its "Offset" field must be set as "0".
- The data type of "iDCS8830\_System\_minor\_fault\_status" and "iDCS8830\_System\_major\_fault\_status" is "DWORD" (32-bit), so the "Offset" requires two Modbus addresses and the "Storage" must be set as "DWORD (Low-High)".

10 [	)rivers *								
冒	🖻 Modbus Master				^				
뮮									
*	🗄 🐨 📲 <2> Read Input Bits (1) [1	· · · · · · · · · · · · · · · · · · ·							
	$\blacksquare < 15 > $ write Coll Bits (1) [6:								
8	□*8 <4> Read Input Registers (			status					
	🗄 📲 <4> Read Input Registers (1) [577664] - I/O status 💦 Double-click								
	📄 🚠 Open MODBUS: 10.0.79.201:5	02							
_	Symbol	Operation	Offset	Mask	Storage				
ŝ	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				
	<								

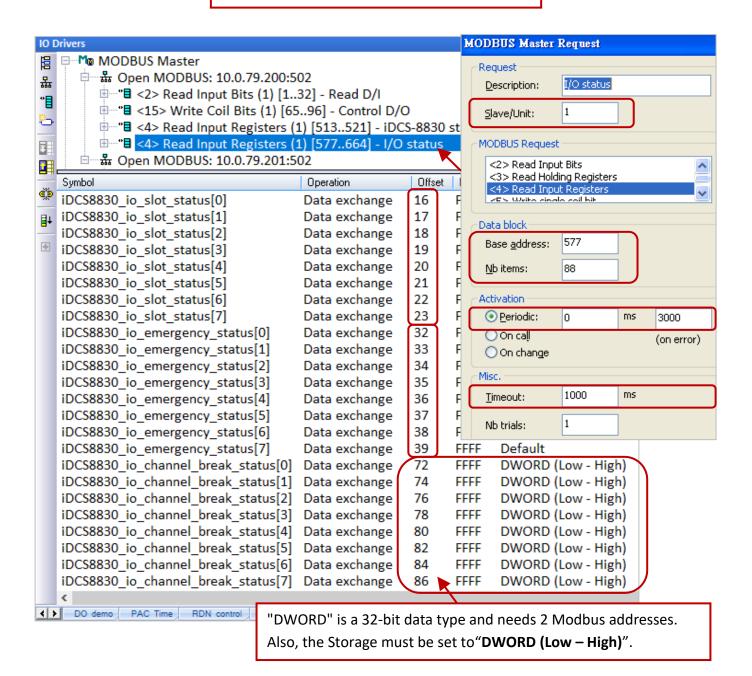
**Note:** Refer to Section 4.2.1 of the <u>FCM-MTCP software manual</u> to set the "Base address".

MODBUS Master Request		X	
Request Description: DC5-8830 :	status	ОК	
Slave/Unit: 1			
MODBUS Request		) ]	
<2> Read Input Bits <3> Read Holding Registers <4> Read Input Registers <5> Write single cell bit			
Data block			
Base address:         513 <u>Nb</u> items:         9		Read 9 ad	ddresses starts from address 513.
Activation			
O Periodic: 0	ms 3000	Send the	request continuously. It will be
○ On caḷ ○ On change	(on error)	sent afte	r <b>3</b> seconds if an exception occurs.
Misc.			
Timeout: 1000	ms	No respo	nse for <b>1</b> second indicates an exception
Nb trials: 1	]		

## Read the I/O status of the iDCS-8830

To read the system information start from address **577** and send the request continuously. It will be sent after **3** seconds if an exception occurs. No response for **1** second can be regarded as abnormal. Refer to Section 4.2.2 of the <u>FCM-MTCP software manual</u> to set the "Base address" and "Offset".

Start Address:	
Module ID:	00577
I/O slot status:	00593
Emergency:	00609
Channel break status:	00649



Refer to <u>Chapter 5</u> for more details on Modbus Master settings and <u>Chapter 3</u> for Modbus Slave settings.

## 14.4.3 Test the Redundant Project (demo\_rdn\_4)

### **Description**:

Two PACs are connected with an iDCS-8830 Redundancy I/O unit via LAN3 through an Ethernet switch.

The following table lists all used devices in this project:

Products	Quantity	Products	Quantity
RPAC-2658M	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	
F-8041	2	CA-37 <b>20</b> AM/ <b>30</b> AM/ <b>50</b> AM/ <b>100</b> AM	4

The following describes how to configure an iDCS-8830. Visit the web page to download the software and user manual.

- ♦ iDCS-8000 Website: https://www.icpdas.com/en/product/iDCS-8830
- Download Center: https://www.icpdas.com/en/download/index.php?model=iDCS-8830
- ♦ Software Manual: https://www.icpdas.com/en/download/show.php?num=1677&model=iDCS-8830

**FPM-D2440 \* 2:** Power module1, Power module2 (for power input, 24V).

FCM-MTCP \* 2: MCU1 (set the SW2 to "C", the SW1 to "8", and the IP address to 10.0.79.200). MCU2 (set the SW2 to "C", the SW1 to "9", and the IP address to 10.0.79.201). SW2/SW1 means the fourth octet of IP address of the MCU. (C8<sub>16</sub> = 200; C9<sub>16</sub> = 201)

### Configure IP addresses by using eSearch Utility:

Open "eSearch Utility" and click "Search Server" to search devices, then click "Configuration" to set the IP addresses. In this example, set two IP addresses to 10.0.79.**200** and 10.0.79.**201** and set the Mask address to "255.255**.0.0**".

🥩 eSearch	n Utility [ v1.2.6, D	ec.09, 2020 ]		_	- 🗆	×
File Server	Tools					
Name	Alias		IP Address	Sub-net Mas	sk Gateway	~
FCM-MTCP	iDCS-8830		10.0.79.200	255.255.0.0	10.0.0.254	
FCM-MTCP	iDCS-8830		10.0.79.201	255.255.0.0	10.0.0.254	$\sim$
<	$\bigcirc$		$\bigcirc$			>
Searc	h Server Cont	figuration (UDP)	2 v	Web	Exit	
Status						1.
onfigure Server	(UDP)					>
erver Name :	FCM-MTCP					
HCP:	0: OFF 🗨	Sub-net Mask :	255.255.0.0	Alias:	iDCS-8830	
P Address :	10.0.79.200	Gateway :	10.0.0.254	MAC:	00:0d:e0:90:02:5	7
Varning!! Contact your Net	work Administrator to g	get correct configur	ation before any ch	anging!	ок	Cancel

### F-8040 \* 2: 32-channel DI modules which plugged into slot 0 and 1.

**F-8041 \* 2:** 32-channel **DO** modules which plugged into **slots 2 and 3**.

### Configure I/O modules by using iDCS-8000

1) Open iDCS-8000 Utility and enter the IP address of the iDCS-8830 to connect.

	iDCS-8000 Utility ver 3.2.3	
MBus	File Online Setting Help	
iDCS-8000 Utility	Connect Q 2 Upload E	Download Monitor
N	Controller	
	IP 10.0.79.200	Note: The PC and the iDCS-8830 must on the same network segment (e.g., <b>10.0.x.x</b> ) to connect successfully.
	·	

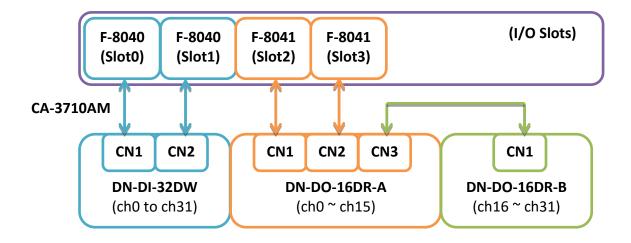
2) Click "Upload" to upload the current I/O settings used in the iDCS-8830.

File Online Setting Help		
Connect Disconnect Upload Download Monitor		Status 下午 02:33:22 Successfully connect with iDCS-8000.
iDCS-8830 3		Successfully connect.
IP 10.0.79.200 Modbus ID 1 Set	Don't change it.	

3) Click the 1st FCM-MTCP (MCU1) and set "IOM\_0 ~ 3" (F-8040/F-8041) as "Duplex" Mode.

iDCS-8000 Utility ver 3.2.3 (Online Mode)	
File Online Setting Help	
	read Load Save Status
iDCS-8830 7	Common IOM Configuration Successfully
IP 10.0.79.200 Modbus ID 1 Set	IOM_0 Duplex I ON I Upload.
D D D D	IOM_1 Duplex VON V
1 1 0 0	IOM_2 Duplex  ON  5
FCM-MTCP	IOM_3 Duplex  ON
	IOM_4 Single V OFF V
	IOM_5 Single V OFF V
	IOM_6 Single V OFF V
	IOM_7 Single V OFF V

3) Make sure that F-8040 and F-8041 modules have connected to the termination boards and set the "Break Line" (i.e., Open Wire Detection) as "ON".



5) Click "Set" and click "Download" to download settings to iDCS-8830 and close the iDCS-8830 utility.

### Start testing:

Make sure that all devices are connected properly and modify the Win-GRAF communication IP address, Active\_IP address, and Mask address. Then download the project ("demo\_RDN\_4") to the PAC. Both the IP address of PAC and PAC must on the same network segment.

#### Note:

Power off the Passive-PAC before **updating the project** to the Active-PAC. After completing the update, power on the Passive-PAC to ensure that the redundant system is work properly.

Click "New Spy1" to open the spy list.

🖆   🛃 🕑   🎒 👗 🗈 🛍 🗙 🛼 🏷	-17	♥   #   # & % 😨 🖒 = §g   🗗 🌘	AU RUN	oooo 👂 🛗 🚅 🛯 🖉 🏟
Workspace		NewSpy1.spl		
🖃 🕨 demo_rdn_4 [RUN]		Name	Value	Description
Exception programs		is_Main_Active	TRUE	<u>^</u>
🖻 📮 Programs		is_Backup_Active	FALSE	
🗝 🔁 DO_demo	∎+	is_Main_ready	TRUE	The Active-PAC is
PAC_Time		is_Backup_ready	TRUE	Main-PAC now.
RDN_control		is_first_cycle_just_after_switch	FALSE	Main-PAC now.
ReadMe (*Only commen		is_Main_LAN1_ok	TRUE	
🔤 🔛 Retain_and_timer		is_Backup_LAN1_ok	TRUE	
🖶 🚞 Watch (for debugging)		is_Alive_port_ok	TRUE	
Initial volues		is_Passive_ready	TRUE	
😪 NewSpy1 🌪		is_Active_LAN1_ok	TRUE	
Do Binding Configuration		is_Passive_LAN1_ok	TRUE	
I/Os		iDCS8830_LAN1_error_status	0	Va O: OK error. LAN
👌 Global defines		iDCS8830_LAN2_error_status	0	Va 32: Master
🖓 Variables		iDCS8830_FCM1_mode	32 🚽	Va , 16#002
🖾 🖻 Types		iDCS8830_FCM2_mode	33	va 33: Slave , 16#002
🦾 🗀 (All Projects)		iDCS8830_FCM1_status	16	<sup>Va</sup> 16: Normal <sup>16#0001</sup>
		iDCS8830_FCM2_status	16	Va 16#0001
		iDCS8830_FPM_status	3	Value: 0x0000: No FPM plugged , C
		iDCS8830_System_minor_fault_status	0	Value: 0: normal , non-zero: somet
		iDCS8830_System_major_fault_status	0	Value: 0: normal , non-zero: somet
		iDCS8830_io_slot_status		Value: 16#0001: Empty , 16#0002:
		iDCS8830_io_slot_status[0]	32	Value: 16#0001: Empty , 16#0002: 🗸
		<		>
	< >	DO demo PAC Time RDN control ReadMe	Retain and timer	IO Drivers Variables NewSpy1

### iDCS-8830's Communication Status:

Name	iDCS8830_LAN1	l_error_status	iDCS8830_LAN2_error_status	
Description	The communication st	atus of LAN1	The communication status of LAN2	
	0: OK Non-zero: Error	Name		Value
Value	Non-zero. Error	iDCS8830	_LAN1_error_status _LAN2_error_status	0 0

## FCM-MTCP's Redundancy Mode and Status:

Name	iDCS8830_FCM:	1_mode	iDCS8830_FCM2_mode	
Description	The redundancy mode of	the FCM1	The redundancy mode of the FCM2	
Description	The start Modbus address			
	<b>32</b> (16#0020): Master	NewSpy1.spl		Value
Value	/alue 33 (16#0021): Slave		Name         Name           iDCS8830_FCM1_mode         3           iDCS8830_FCM2_mode         3	

Name	iDCS8830_FCM1	_status	iDCS8830_FCM2_	iDCS8830_FCM2_status	
Description	The status of the FCM1		The status of the FCM2		
Description	The start Modbus address	: 514			
Value	<ul> <li>0 (16#0000): Empty</li> <li>1 (16#0001): Timeout</li> <li>2 (16#0002) Undefinded</li> <li>16 (16#0010) Normal;</li> </ul>		_FCM1_status _FCM2_status	Value 16 16	

	1.	1. Unplug the LAN1 cable and the FCM2 will change to Master (32) and take over.						
	iDCS8830_LAN1_error_status=130 iDCS8830_LAN2_error_status = 0							
		iDCS8830_FCM1_mode = 33 iDCS8830_FCM2_mode = 32						
		iDCS8830_FCM1_status = 1	iDCS8830_FCM2_status = 16					
Test Steps								
		iDCS8830_LAN1_error_status=0	iDCS8830_LAN1_error_status=0 iDCS8830_LAN2_error_status = 0					
		iDCS8830_FCM1_mode = 33	DCS8830_FCM1_mode = 33 iDCS8830_FCM2_mode = 32					
		iDCS8830_FCM1_status = 16	iDCS8830_FCM2_status = 16					
			·					

#### iDCS-8830's Power Status:

Name	iDCS8830_FPM_status					
Description	The status of two power modules					
Description	The start Modbus address: 516					
Value	0: No FPM is plugged         1: FPM1 Off / FPM2 Good         2: FPM1 Good / FPM2 Off         3: Two FPM are Good.					

#### iDCS-8830's Failuure Status:

Name	iDCS8830_System_minor_fault_status	iDCS8830_System_major_fault_status			
Description	The status of a glitch or major failure of the	system			
Description	The start Modbus address: 517 / 518 The start Modbus address: 519 / 520				
Value	0: Empty Non-zero: Something wrong, refer to Section NewSpy1.spl	n 4.2.1 of the <u>FCM-MTCP software manual</u>			
	Name	Value			
	iDCS8830_System_min				
	iDCS8830_System_ma	jor_fault_status 0			

## The Status of I/O Slots

Name	iDCS8830_FPM_status				
Deceriation	The status of I/O slots 0 to 7				
Description	The start Modbus address: 59	93			
	<b>1</b> (16#0001): Empty		NewSpy1.spl		
	<b>2</b> (16#0002): Halt		Name	Value	
			■ iDCS8830_io_slot_status		
	<b>4</b> (16#0004): Bootup		iDCS8830_io_slot_status[0]	32	
	<b>8</b> (16#0008): Bootloade	∎+	iDCS8830_io_slot_status[1]	32	
Value			iDCS8830_io_slot_status[2]	32	
	<b>16</b> (16#0010): Pre-operation		iDCS8830_io_slot_status[3]	32	
			iDCS8830_io_slot_status[4]	1	
	<b>32</b> (16#0020): Operation		iDCS8830_io_slot_status[5]	1	
	, , , , , , , , , , , , , , , , , , , ,		iDCS8830 io slot status[6]	1	
	<b>64</b> (16#0040): Stop		iDCS8830 io slot status[7]	1	

## The Status of I/O Emergency:

Name	iDCS8830_io_emergency_status						
Description	The Emergency status of two power modules <u>Note</u> : "iDCS8830_io_emergency_status[x]" only works when the "IOM_ The start Modbus address: <b>609</b>	_x" is set to " <b>Duple</b>					
Value	0: OK, 32 (16#0020): CJC Error 256 (16#0100): Cable Brea	k-off Value 0 0 0 0 0 0 0 0 0					

## The Status of I/O Channels:

Name		iDCS8830_io_channel_break_status						
Description	Open wire detect	Open wire detection for I/O channels 0 to 7						
Description	The start Modbus address: 649							
	<b>0</b> : Normal,							
	Non-zero: The co	rresponding bit is 1, which means the wire is disc	onnected					
		NewSpy1.spl						
		Name	Value					
		iDCS8830_io_channel_break_status						
		<ul> <li>iDCS8830_io_channel_break_status[0]</li> </ul>	0					
Value	8	iDCS8830_io_channel_break_status[1]	0					
		iDCS8830_io_channel_break_status[2]	0					
		iDCS8830_io_channel_break_status[3]	0					
		iDCS8830_io_channel_break_status[4]	0					
		iDCS8830_io_channel_break_status[5]	0					
		iDCS8830_io_channel_break_status[6]	0					
		iDCS8830 io channel break status[7]	0					

#### The Status of F8040 (DI) and F8041 (DO) Modules

Name	iDCS8830_slot01_F8040_DI	iDCS8830_slot23_F8041_DO
Description	The status of F8040 (32 DI) modules on I/O slots 0 and 1	The status of F8041 (32 DO) modules on I/O slots 2 and 3
	The start Modbus address: 1	The start Modbus address: 65
Tost Stons		

#### Test Steps

At first, the LEDs (DO0 – 7) of the F-8041 module on slot2 will light up sequentially.

- 1. Unplug the F-8041 module on slot2 and, the F-8041 module on slot3 will take over and light up LEDs in turns.
- 2. At this time, the status value of the iDCS8830\_io\_slot\_status[2] is changed to "64" which means the module on slot2 stops working. Plug the module into the I/O slot again.

Name	Value		Name	Value
iDCS8830_io_slot_status				
iDCS8830_io_slot_status[0]	32		iDCS8830_io_slot_status[0]	32
iDCS8830_io_slot_status[1]	32	<b>a</b> t	iDCS8830 io slot status[1]	32
iDCS8830_io_slot_status[2]	32		iDCS8830_io_slot_status[2]	64
iDCS8830_io_slot_status[3]	32		iDCS8830_io_slot_status[3]	32
iDCS8830_io_slot_status[4]	1		iDCS8830_io_slot_status[4]	1
iDCS8830_io_slot_status[5]	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		iDCS8830_io_slot_status[7]	1
iDCS8830_io_emergency_status			iDCS8830_io_emergency_status	
			iDCS8830_io_channel_break_status	
iDCS8830_slot01_F8040_DI			iDCS8830_slot01_F8040_DI	
iDCS8830_slot23_F8041_DO			iDCS8830_slot23_F8041_DO	
iDC\$8830_slot23_F8041_DO[0]	TRUE		iDC\$8830_slot23_F8041_DO[0]	TRUE
iDC\$8830_slot23_F8041_DO[1]	TRUE		iDC\$8830_slot23_F8041_DO[1]	TRUE
iDC\$8830_slot23_F8041_DO[2]	TRUE		iDCS8830_slot23_F8041_DO[2]	TRUE
iDC\$8830_slot23_F8041_DO[3]	TRUE		iDC\$8830_slot23_F8041_DO[3]	TRUE
iDCS8830_slot23_F8041_DO[4]	TRUE		iDCS8830_slot23_F8041_DO[4]	TRUE
iDCS8830_slot23_F8041_DO[5]	TRUE		iDCS8830_slot23_F8041_DO[5]	TRUE
iDCS8830_slot23_F8041_DO[6]	TRUE		iDCS8830_slot23_F8041_DO[6]	TRUE
iDCS8830 slot23 F8041 DO[7]	TRUE		iDC\$8830_slot23_F8041_DO[7]	TRUE

The above is the description of testing the iDCS-8830 redundant I/O, also refer to <u>Section 14.4.1</u> to test the RPAC redundant system.

# **Chapter 15 Schedule Control**



#### Introduction:

All Win-GRAF PACs support the Schedule-Control function. One PAC can control the scheduling for up to 10 Targets. Each Target can control one boolean (BOOL), integer (DINT), and real (REAL) variables and up to 5 weekly schedules can be set for each year or four seasons. Also, up to 15 periods of time can be set for one schedule.

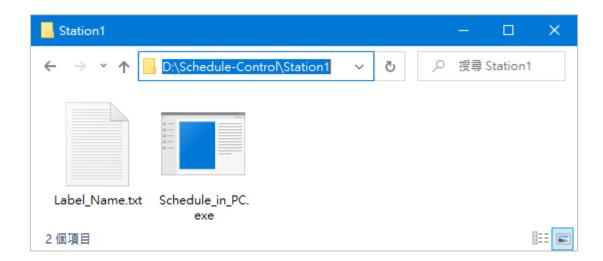
ICP DAS provides free software – "Schedule-Control Utility" for users to simply and quickly complete the scheduling configurations.

## **15.1** Install the Schedule-Control Utility and Restore the Win-GRAF Project

The user can download the Schedule-Control Utility ("Schedule\_in\_PC.zip") on the website.

#### Install the Schedule-Control Utility:

Extract the zip file and copy Schedule\_in\_PC.exe and label\_name.txt to your PC. It is recommended to store files in the project directory (e.g., D:\Schedule-Control\**Station1**\).

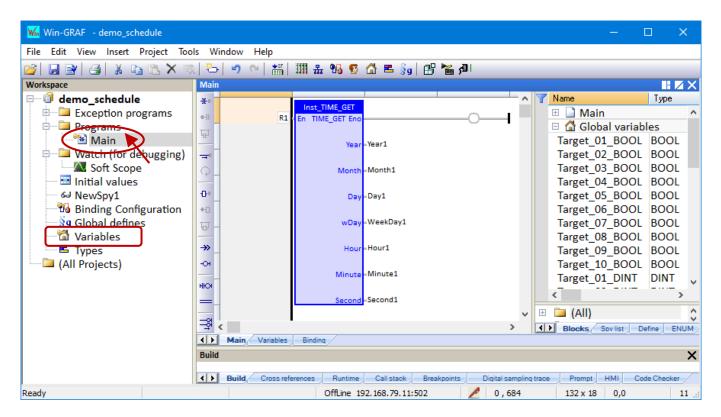


#### **Restore a Win-GRAF Demo Program:**

Download <u>the demo program</u> on the website (**demo\_schedule.zip**). Click the menu commands **File** - **Add Existing Project** - **From Zip...** in the Workbench to open the project. (Refer to Section 11.4)

## 15.2 Introduction of the "demo\_schedule" Project

This project includes one LD program (Main) that used to get the system time.



Double-click "Variable" can view or set variables.

Name	Data Type	Description
Target_01_BOOL to Target_10_BOOL	BOOL	For ten targets to control Boolean variable
Target_01_DINT to Target_10_DINT	DINT	For ten targets to control DINT variable
Target_01_REAL to Target_10_REAL	REAL	For ten targets to control REAL variable
Year1		
Month1		
Day1		
WeekDay1	DINT	Used for "TIME_GET" function block
Hour1		
Minute1		
Second1		

## 15.2.1 "I/O Boards" Setting (Schedule)

To enable the Schedule-control on the PAC, the user must add the "Schedule" function in the "I/O Boards" window (refer to Chapter4)

i 👬 🎹 🏭 😘 😨	🖆 🖻 💡 📴 🅍 🖓
IIII I/O Boards	×
0 1 2 3 4 5 6 7 8 9 10 Schedule 11 12 13 14	10: Schedule - Properties     ×       Key = 6 Ref = 16#10 Password = 0 Reserved0 = 0 Reserved1 = 0 Reserved2 = 0 Reserved3 = 0     Password
15 16	Parameters: Password : DINT : 0 : No password protection. others : a password number for the "Schedule Utility" to

#### Parameters:

#### Password: DINT:

0: No password protection.

Others: A password number for the "Schedule Utility" to access the Win-GRAF PAC.

### 10-Ch Boolean Inputs: (TRUE: Enabled; FALSE: Disabled)

To indicate the related target of the schedule-control is enabled or disabled. For example, if Ch.O and Ch.5 return TRUE, it means the Target 1 and Target 6 of the schedule control are enabled.

The password is "0" in this example. If a password is set, the user will be asked for entering the password before downloading the settings to the PAC by using "Schedule-Control Utility". After adding the "Schedule" I/O board, there are 10 Boolean variables that are used to display the status of the schedule-control will automatically be added.

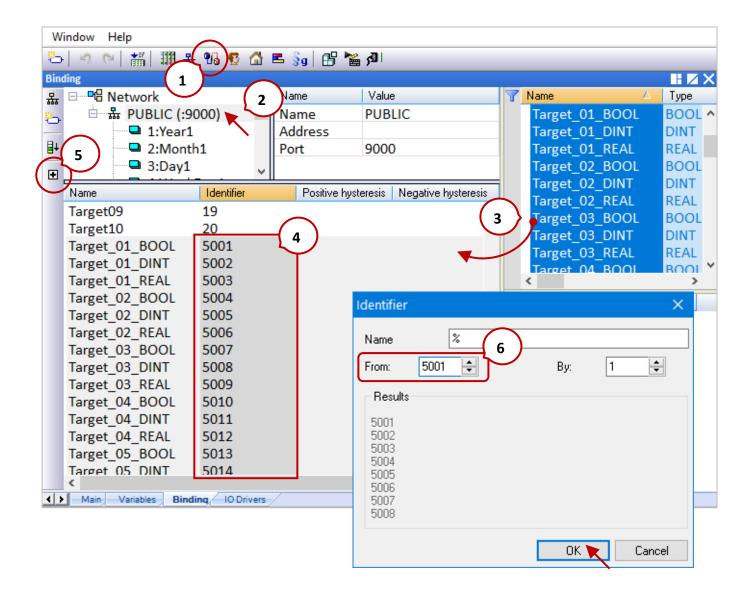
Variables								HZ	1X
🝸 Name	Туре	Dim.	Attrib.	Syb.	Init value	User Group	Tag	Description	
🗆 📫 %IX10 - Schedule									~
%IX10.0=Target01	BOOL		Input						
%IX10.1=Target02	BOOL		Input						
%IX10.2=Target03	BOOL		Input						
%IX10.3=Target04	BOOL		Input						
%IX10.4=Target05	BOOL		Input						
%IX10.5=Target06	BOOL		Input						
%IX10.6=Target07	BOOL		Input						
%IX10.7=Target08	BOOL		Input						
%IX10.8=Target09	BOOL		Input						
%IX10.9=Target10	BOOL		Input						~

## 15.2.2 Public the Variable Data (Data Binding)

To public Win-GRAF variables for other PACs to read/write data, add the needed PUBLIC variables in the "Binding" window and specify the identifier, refer to Chapter 7. And, the ID of control variables must be set to **5001 to 5030**.

## Follow the steps:

- 1. Click the **Open Binding Configuration** button to display the **Binding** window.
- 2. Click on the **PUBLIC (:9000)** setting.
- 3. Drag variables to be published to the ID mapping area.
- 4. Select multiple Identifier fields that you want to set a continuous number.
- 5. Click the Iterate Property button.
- 6. Enter the starting ID number in the **From** field.



## **15.3 Description of the Schedule-Control Utility Example**

This example briefly describes how to set Target 1, also refer to Section 15.5 to learn more about the settings.

 Execute the Schedule-Control Utility (Schedule\_in\_PC.exe) and click the New button and then click Target 1 to display the settings screen.

	🖳 Schedule-Co	ontrol Utility				_	×	
🖳 Schedule-Control Utility	Save to PC	Send to Controller	Controller time synchronization	Open from PC	Get from Controller		Help-about	- 🗆 🗙
New 🔪	Title : Title							Help-about
		Target 1	Target 2		Target	3		
		Target 4	Target 5		Target (	5		
		Target 7	Target 8		Target 9	)		
			Target 10					

- 2. Tick the Target 1, Season Always, and Normal day to enable the setting.
- 3. Select the scheduling options you want to use (ex: Schedule 1) and click the **Schedule 1** button to open the window.

🖳 Schedule-Control Utility	Target 1 > Season Always	> Normal day				_		$\times$
Back Save to PC	a Farget 1	Default Value	Boolean OFF v	Integer O	Res 0.0	1		
Season Always *	Beason Always Season 1	Sea	son 2	Alway Season 3	78	Season 4		
	*號表示已設定	定,若選	擇其他的	」 項目・也必	《需勾選》	才可設;	定。	
<ul> <li>Normal day (Schedule 1 !!!)</li> <li>O Holiday 1</li> </ul>	Jormal day           Apply         Sched           Sunday         Monday	dule 1 🗸 🗸			Schedule 1 Schedule 2			
○ Holiday 2 ○ Special day	⊻ Honnay ⊻ Tuesday ⊻ Wednesday ⊻ Thursday				Schedule 3 Schedule 4			
	🗹 Friday 🗌 Saturday				Schedule 5			

RPAC-2658M User Manual, V1.0, Mar. 2021 by ICP DAS 239

4. Each schedule can set up to 15 periods of time, check the box to set the desired time and the value of control variables. After completing the setting, click the "Save and exit" button to save and exit the window.

									Copy from					
	He	our	Min	ute	То	Ho	or	Min	ute	в	oole	an	Integer	Real
01:	8	~	30	~		12	~	0	~	0	N	$\sim$	10	12.35
02:	13	~	0	$\sim$		17	~	30	~	0	N	$\sim$	20	27.5
03:	0	$\sim$	U	$\sim$		0	$\sim$	0	~	0	기가	$\sim$	0	0
04:	0	$\sim$	0	$\sim$		0	$\sim$	0	$\sim$	0	FF	$\sim$	0	0
05:	0	$\sim$	0	$\sim$		0	$\sim$	0	$\sim$	0	FF	$\sim$	0	0
06:	0	$\sim$	0	$\sim$		0	$\sim$	0	~	0	FF	$\sim$	0	0
07:	0	$\sim$	0	$\sim$		0	$\sim$	0	~	0	FF	$\sim$	0	0
08:	0	$\sim$	0	$\sim$		0	$\sim$	0	~	0	FF	$\sim$	0	0
09:	0	$\sim$	0	$\sim$		0	$\sim$	0	~	0	FF	$\sim$	0	0
10:	0	$\sim$	0	$\sim$		0	$\sim$	0	$\sim$	0	FF	$\sim$	0	0
11:	0	$\sim$	0	$\sim$		0	$\sim$	0	~	0	FF	$\sim$	0	0
12:	0	$\sim$	0	$\sim$		0	$\sim$	0	~	0	FF	$\sim$	0	0
13:	0	~	0	$\sim$		0	$\sim$	0	~	0	FF	$\sim$	0	0
14:	0	~	0	$\sim$		0	$\sim$	0	~	0	FF	$\sim$	0	0
15:	0	~	0	$\sim$		0	$\sim$	0	~	0	FF	$\sim$	0	0

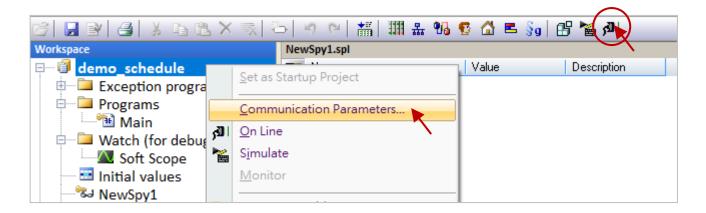
5. The "Default Value" settings will be applied if the time is not in the scheduled period. Click the "Save to PC" button to save the configuration file (e.g., test1.txt).

🖳 Schedule-Control Utility	Target 1 > Season Always	> Normal day				—		×
Back Save to PC	☑ Target 1		Boolean DFF ~	Integer 0	Real		)	
	☑ Season Always			Always				
Season Always *	Season 1	Seaso	n 2	Season 3		Season 4		
		* indica	tes the it	em is set.				
	☑ Normal day			[	Schedule 1 *			
• Normal day (Schedule 1 *)	Apply Sche Sunday	dule 1 * 🗸 🗸		[	Schedule 2			
<ul> <li>Holiday 1</li> <li>Holiday 2</li> </ul>	☑ Monday ☑ Tuesday			[	Schedule 3			
O Special day	☑ Wednesday ☑ Thursday ☑ Friday			[	Schedule 4			
	Saturday			[	Schedule 5			

# 15.4 Testing the Schedule-Control on the PAC

Before testing, download the scheduling settings and the Win-GRAF project to the Win-GRAF PAC.

First, download the Win-GRAF project (demo\_schedule) to the PAC.
 Modify the communication IP address and click the **On Line** button to download the project.



 Download the scheduling settings to the PAC by using Schedule-Control Utility. Click the Send to Controller button, enter the IP address of the PAC and the password for the scheduling function (in this case, 0), and click OK. After sending files, the "Send file success" dialog will be displayed and the utility will be automatically closed.

🖳 Schedule-Control Utility	- D X
Save to PC Send to Controller	Controller time synchronization Open from PC Get from Controller Help-about
Title : Title	
Target 1 *	Target 2 Controller info – □ ×
Target 4	IP :       192.168.79.11         Refer to Section 15.2.1 to set the password in the "Schedule" I/O board.
Target 7	OK Cancel arget 9
Sending file	
	Send file success.
	100 % OK

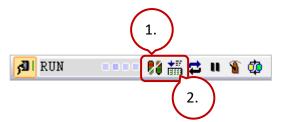
3. Test the scheduling. Open the Win-GRAF "NewSpy1" window and you can see "Target\_01 = TRUE" which means **Target 1** is enabled. In the example, we set two periods of time.

01: 08:30 ~ 12:00	Boolean=ON,	Integer=10,	Real=12.35
02: 13:00 ~ 17:30	Boolean=ON,	Integer=20,	Real=27.5

₩ Win-GRAF - demo_schedule							
File Edit View Insert Project Tools	Win	dow Help					
🚰 🛃 🕑 🖪 🐰 ங 🗈 🗙 🐟 🗁 🕐 🛗 🏭 🏭 🕵 🕵 🖆 🖷 윩 🚰 RUN 🛛 >>>>> 💔 🏭 😅 🖤 🍸							
Workspace NewSpy1.spl							
💷 🕨 demo_schedule [RUN]		Name	Value	Description			
Exception programs		Year1	2021				
🖻 📮 Programs		Month1	2				
📲 Main	≣∔	Day1	8				
🖃 📮 Watch (for debugging)		WeekDay1	1				
Soft Scope		Hour1	16				
Initial values		Minute1	31				
NewSpy1		Second1	39				
Binding Configuration		Target01	TRUE				
💮 🕄 Global defines		Target02	FALSE				
🖓 Variables		Target03	FALSE				
Types		Target_01_BOOL	TRUE				
🦾 🗀 (All Projects)		Target_01_DINT	20				
		Target_01_REAL	27.5				
		Target_02_BOOL	FALSE				
		Target_02_DINT	0				
		Target_02_REAL	0.0				
		Target_03_BOOL	FALSE				
		Target_03_DINT	0				
		Target_03_REAL	0.0				
	<b>&lt; &gt;</b>	NewSpy1 Main Bindin	q Variables				

#### Note:

If the scheduling setting is changed and downloaded to the PAC, you can reboot the PAC or click the "Stop application" button and click the "Download" button in Win-GRAF to apply the settings.



## 15.5 How do I Use the Schedule-Control Utility

## 15.5.1 Address for Control Variables (BOOL, DINT, and REAL)

Up to 10 targets can be configured in the Schedule-Control Utility. Each Target can control one BOOL variable, one DINT variable, and one REAL variable.

To publish the following variables for other devices to access data, all variables must be declared in the Variables window (refer to Section 15.2) and add them into the Win-GRAF **Binding** widow also specify the identifier that must be **5001 to 5030** (refer to Section 15.2.2).

Address	Туре	Description	Address	Туре	Description
5001	BOOL	BOOL, DINT, and REAL	5016	BOOL	BOOL, DINT, and REAL
5002	DINT	variable controlled	5017	DINT	variable controlled
5003	REAL	by Target 1	5018	REAL	by Target 6
5004	BOOL	BOOL, DINT, and REAL	5019	BOOL	BOOL, DINT, and REAL
5005	DINT	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	<ul> <li>BOOL, DINT, and REAL</li> <li>variable controlled</li> </ul>	5029	DINT	variable controlled
5015	REAL	by Target 5	5030	REAL	by <b>Target 10</b>

## 15.5.2 Configure the Target



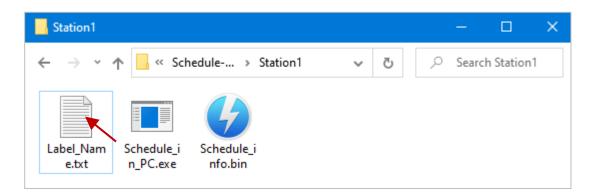
Execute the Schedule-Control Utility and click "New" to display the settings screen. Each Win-GRAF PAC can control a maximum of 10 Targets and the default name is Target 1 to Target 10.

- 1) **Open from PC**: Open an existing configuration file.
- 2) Get from Controller: Load the current settings from the PAC.

🖳 Schedule-Control Utility		1.	2.
New		Open from PC G	et from Controller Help-about
Schedule Save to PC	-Control Utility Send to Controller	Controller time synchronization Open fr	om PC Get from Controller Help-about
Title : T	itle		
	Target 1	Target 2	Target 3
	*: The target ha	as been set.	
	Target 4	Target 5	Target 6
	Target 7	Target 8	Target 9
		Target 10	

#### Change the Target Name:

There is a "Label\_Name.txt" file in the project folder (e.g., D:\Schedule-Control\Station1\). The name of Target, Season, and so on in the file can be changed according to the needs of the application. The default name will be displayed if the file does not exist.



### Edit the "Label\_Name.txt" File:

- 1. Enter the desired name after the colon (": ") symbol. E.g., Device-01 and the prefix/suffix of spaces will be erased.
- 2. After completing the edit, saving the file in a Unicode format.
- 3. Both the "Label\_Name.txt" and "Schedule\_in\_PAC.exe" must be stored in the same folder.

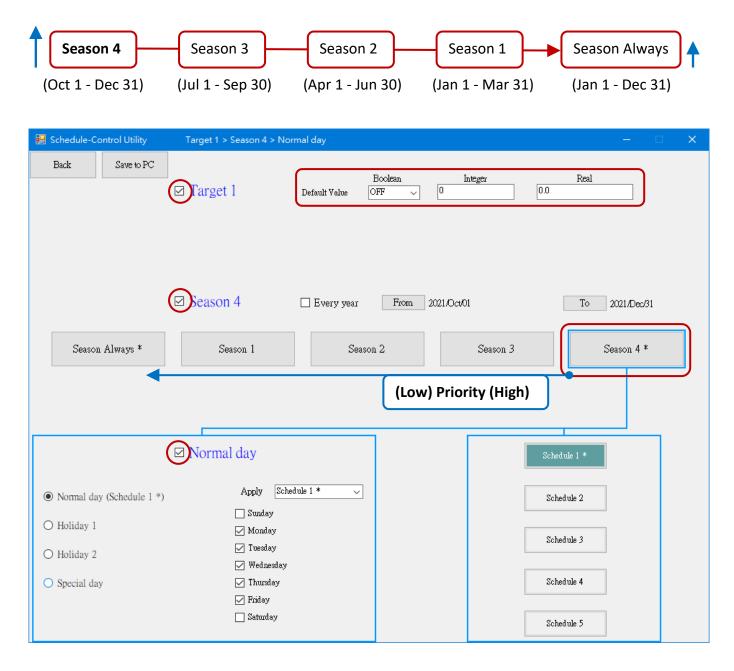
	1.		
//////////////////////////////////////			
File Edit Format Vie			
Target1 : Device-01			
larget2 : larget 2 Target3 : Target 3	Save As		×
Target4 : Target 4			
Target5 : Target 5	← → × ↑ 📙 D:\Schedule	-Control∖Station1 ∨ Ō	,○ Search Station1
Target6 : Target 6			
Target7 : Target 7	File name: Label_Name.	txt	~
Target8 : Target 8	Save as type: Text Docume	ents (*.txt) 2.	~
Target9 : Target 9			
Target10 : Target 10	✓ Browse Folders En	coding: UTF-16 LE 🗸 🗸	Save 💽 Cancel
Season Always : Sea			
Season 1 : Season	🔛 Schedule-Control Utility		– 🗆 X
Season 2 : Season	Save to PC Send to Controller	Controller time synchronization Open from	
Season 3 : Season		Controller time synchronization Open from	rec Gernom contoller neth-about
Season 4 : Season	Title : Title		
Normal day : Norm			
Holiday 1 : Holiday			
Holiday 2 : Holiday	Device-01	Target 2	Target 3
Special day : Specia			
Schedule1 : Schedu		)	
Schedule2 : Schedu			
Schedule3 : Schedu	Target 4	Target 5	Target 6
Schedule4 : Schedu	Taiget 4	Taiget J	Target 0
Schedule5 : Schedu			
1009 Windows (CRLF)			
	Target 7	Target 8	Target 9
		Target 10	

## 15.5.3 Configure the Season

Each Target provides the Season Always, Season 1, Season 2, Season 3, and Season 4 settings that can be used to configure all year-round or four seasons schedules. Note that check the checkbox before setting.

## The Searching Priority of Seasons:

It is recommended to enable the **Season Always** setting. If multiple Season settings are enabled, the PAC will apply the Boolean, Integer, and Real data in the following orders. The default setting (OFF, 0, 0.0) will be applied if no available setting is found.



### **The Season Setting:**

To enable Season (1 to 4), the date range must be set. Also, check the "Every year" box to apply the setting every year.

### Notice:

- **1.** The specified date range of Season 1 to 4 cannot overlap and the start date (From) must earlier than the end date (To).
- **2.** Uncheck the "Every year" box before changing the date.

#### For example,

The date range of Season 4 is **Oct 01** - Dec 31. If the date range of Season 3 is set to Jul 01 - **Oct 24**, an error message will be displayed when clicking the **Save to PC** button to save the settings.

🔛 Schedule-Control Utility	Target 1 > Season 3 > Norma	l day					—	×
Back Save to PC	⊠ Target 1 D		Boolean DFF v	0 0	nteger	4 0.0	Real	
Season Always *	⊠ Season 3	] Every year Seasoi		ับ1/01	Season 3 *		To Oct/24 Season 4 3	
Season Aiways *	Jeason 1	Seaso	12		Season 5 *		Jeason 4	
	☑ Normal day Apply Schedule	1 *				Schedule 1	*	
<ul> <li>Normal day (Schedule 1 *)</li> <li>Holiday 1</li> </ul>	Apply Schedule	1* ~				Schedule (	2	
O Holiday 2	☑ Fuesday ☑ Tuesday ☑ Wednesday					Schedule :	3	
○ Special day	🗹 Thursday 🖵 Friday					Schedule 4	4	
	🗌 Saturday					Schedule !	5	

## Error\_List Target 1: Season 3: [Season 3] overlaps the period of [Season 4]. Season 4: [Season 4] overlaps the period of [Season 3].

## 15.5.4 Configure the Normal Day / Holiday / Special Day

Each Season provides the Normal day, Holiday 1, Holiday 2, and Special day setting that can be used to configure the schedule for a weekday, holiday, or specific day. Note that check the checkbox before setting.

## The Searching Priority of Days:

If multiple Day settings are enabled, the PAC will apply the Boolean, Integer, and Real data in the following orders. The default setting (OFF, 0, 0.0) will be applied if no available setting is found.

Special day	Holida	ay 2 Holiday	y 1 → Normal d	ay
Schedule-Control Utility Target Back Save to PC	:1 > Season Always > H			– 🗆 X
⊡Tar	get 1	Bcolean Default Value OFF 🗸	Integer 23.5	Real
Sea	son Always		Always	
Season Always *	Season 1	Season 2	Season 3	Season 4
Hol	iday 1		Schedu	le 1 *
O Normal day (Schedule 1 *)	Apply Schedule	e 2 * 🗸	Schedu	le 2 *
<ul> <li>Holiday 1 (Schedule 2 *)</li> <li>Holiday 2</li> </ul>	🗹 Sunday 🔲 Monday 🔲 Tuesday		Sched	ule 3
O Special day *	🗌 Wednesday 🔲 Thursday 🗍 Friday		Sched	ule 4
	🖌 Thirday		Sched	ule 5

Normal day	Working days, usually Monday to Friday.	
Holiday 1	Normally set to Saturday and Sunday.	If the Day setting is enabled,
Holiday 2	It can be set from Monday to Sunday.	at least one Schedule (1-5)
Special Day	Used to specify a date, e.g., Oct 10, Jul 4, etc. Up to 50 days can be set for each Season.	must be set.

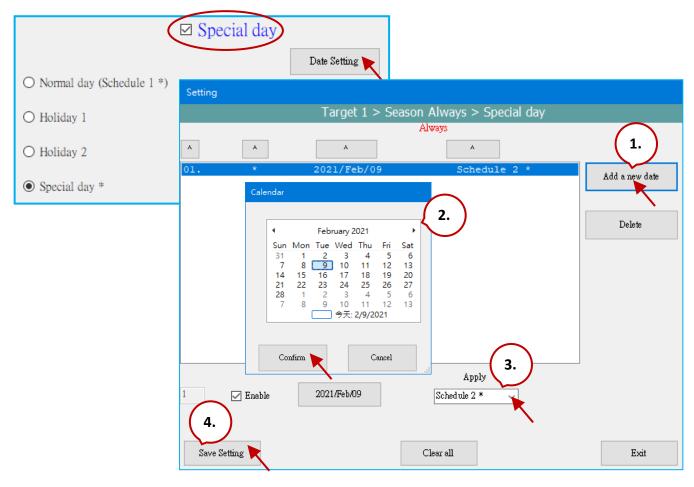
## Notice:

**The specified day of the Normal day, Holiday 1, and Holiday 2 cannot overlap.** E.g., Both the Wednesday of the Normal day and Holiday 1 are checked, an error message will be displayed when saving the settings.

⊠ No.	rmal day			
• Normal day (Schedule 1 *)	Apply Sched	ule 1 * 🗸	🖂 Hol	iden 1
O Holiday 1 (Schedule 2 *)	🖂 Monday			luay I
🔿 Holiday 2	🔽 Tuesday 🔽 Wednesday	🔿 Normal day (Sch	edule 1 *)	Apply Schedule 1 * 🗸
O Special day *	🗹 Thursday	Holiday 1 (Sched	tule 1 *)	🗹 Sunday
	🗹 Friday	O Honday I (Senec	auto i )	Monday —
	📃 Saturday	○ Holiday 2		Tuesday
				🗹 Wednesday
Error_List				Thursday
				🔲 Friday
Target 1: Season Always:				🗹 Saturday
Both [Normal day] and [Holiday 1] enable Wednesday. Both [Holiday 1] and [Normal day] enable Wednesday.				

## Specify the date of the Special Day:

"Special Day" can be used to specify a maximum of 50 dates for each "Season" setting.



RPAC-2658M User Manual, V1.0, Mar. 2021 by ICP DAS 249

## 15.5.5 Configure the Schedule Time

Up to five Schedules (1 to 5) can be set in each Season setting and up to fifty periods of time can be set in each Schedule setting. The minimum unit of time is minutes, which must be within the range of 00:00 to 24:00. Note that check the checkbox before setting.

## The Searching Priority of Schedule Time:

If multiple schedule times are enabled, the PAC will apply the Boolean, Integer, and Real values according to the settings with the larger number. The default values (OFF, 0, 0.0) will be applied if no available setting is found.

Season	] Every year From Ja	an/01 To Mar/31	
Season Always * Seas	son 1 * Season 2	Season 3 Season 4	
Normal	day	Schedule 1 *	
	Apply Schedule 1 *		
Normal day (Schedule 1 *)	Sunday	Schedule 2	
⊖ Holiday 1	☑ Monday	Schedule 3	
O Holiday 2	☑ Tuesday ☑ Wednesday		
O Special day	🗹 Thursday 🗹 Friday	Schedule 4	
	Saturday	Schedule 5	
Schedule 1			
	Copy from		
Hour Minute	fo Hour Minute	Boolean Integer Real	
	8 🗸 0 🗸	OFF V 100 30	
✓ 02:     8     ∨     0     ∨       ✓ 03:     12     ∨     0     ∨	$12 \sim 0 \sim$ $13 \sim 0 \sim$	ON v 150 25.5 OFF v 120 27	
✓ 04: 13 ✓ 0 ✓		ON V 150 25.5	
05: 17 0	24 ~ 0 ~	OFF ~ 100 30	
□ 07: □ 08: Setting 05 will be ap	plied at 17:00	OFF v 0 0 OFF v 0 0	
		OFF V 0 0	
10: 0 ~ 0 ~	0 ~ 0 ~	OFF V 0 0	
□ 11: 0 ~ 0 ~	0 ~ 0 ~	OFF V 0 0 (Low)	)
	0 ~ 0 ~	OFF 0 0 (Low)	
		OFF 0 0 Priority	
14:     0     ✓     0     ✓       15:     0     ✓     0     ✓		OFF V 0 0 (High)	
Save and exit		Cancel	

## 15.5.6 Save and Send the Configuration File

After completing the setting, save it as a file on PC and then send configurations to the PAC.

🔜 Schedule-Control Utility			- 🗆 🗙
Save to PC Send to Controller	Controller time synchronization	Open from PC Get from Controller	Help-about
Target 1 *	Target 2	Target 3	

1. Save the file on PC

Click the **Save to PC** button to save the configuration file (\*.txt).

🔜 Save As						×
← → · ↑ 📙 D:\Sc	hedule-Control\Station1	ٽ ×	Q	Search Station	1	
Organize 🔻 New folde	er					?
Label_Name.txt test1.txt test2.txt						
File name: test2.t Save as type: log fil						>
<ul> <li>Hide Folders</li> </ul>				Save	Cancel	

2. <u>Send the Configuration to the PAC</u>

Click the **Send to Controller** button to send configurations to the PAC. Enter the IP address of the PAC and the password (defaults: 0).

🔛 Controll	er info	—		×	
IP :	192.168.79.	11			
Password :	•				
	🗹 Remember password				
OK			Cancel	:	

## 15.5.7 Time Synchronization

If the system time of the PAC is different from the current time, for conducting the scheduling on time, run the time calibration in the Schedule-Control Utility to correct the PAC time. Make sure that both the PC and Win-GRAF PAC are online.

1. Click the **Controller Time Synchronization** button and enter the IP address of the PAC and the password (defaults: 0). Then, click OK.

🔜 Schedule-Control Utility	
Save to PC Send to Controller	Controller time synchronization Open from PC Get from Controller
Title : Title	E Controller info – 🗆 X
	IP : 192.168.79.11 Password : ● ☑ Remember password OK Cancel

2. Confirm the date, hour, minute, and second, then click the **Set new Controller time** button to do the time synchronization.

📙 Controller time setting				×
		Hour	Minute	Second
Controller time :	Tuesday , February $~9 \lor$	16	27	37
Set new Controller time :	Tuesday , February 9 🗸	16 🗸	27 🗸	18 🗸
			Se	t new Controller time

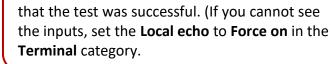
# Chapter 16 Win-GRAF SMS Function

This chapter shows the way to send/receive an SMS message by using the RPAC-2568M that comes with the <u>GTM-204M-4GE</u> 3G/4G wireless module.

#### Note:

- 1. Due to the product certification issue, these modules can sale in certain areas. Please contact our agents for more information.
- 2. Visit the <u>download page</u> of the GTM-204M-4GE and refer to the quick start guide to install the USB driver and test the module.

🖁 裝置管理員		– 🗆 X
檔案(F) 動作(A) 檢視(V) 說明(H)		
<ul> <li>◆ 를 連接堤 (COM 和 LPT)</li> <li>● GTM-204M-4G USB AT Port (COM7)</li> <li>● GTM-204M-4G USB DM Port (COM6)</li> <li>● GTM-204M-4G USB NMEA Port (COM9)</li> <li>● 通訊連接埠 (COM1)</li> </ul>	To view or set the E right click – Proper	
<ul> <li>&gt; ■</li></ul>	R PuTTY Configuration	If there is no hyper terminal on your PC, using the PuTTY to test.
VMware Virtual Ethernet Adapter for VMnet1		
Image: System Structure       Image: System Structure         Image: System Structure       Image: System Structure     <	Category: Session Logging Terminal Keyboard Bell Features Window Appearance Behaviour Translation Selection Colours Connection Proxy Telnet Rlogin SSH	Basic options for your PuTTY session         Specify the destination you want to connect to         Serial line         Specify the destination you want to connect to         COM7         115200         Connection type:         Raw       Telnet         Raw       Telnet         Raw       Telnet         Saved Sessions         Default Settings         Load         Save         Delete
	About Help	Close window on exit: Always Never Only on clean exit Open Cancel
🛃 COM7 - PuTTY — 🗖	Type AT	and press Enter, returns "OK" indicates test was successful. (If you cannot see



# 16.1 "GSM\_Open", "Send\_SMS" and "Read\_SMS" Functions

The following three functions can be used to send or receive an SMS message in the Win-GRAF PAC.

# The "GSM\_OPEN" function:

To open/close the GSM module.



Inputs	Data Type	Description
gEnable	BOOL	<ul><li>TRUE: Open the specified COM port to connect the GSM module and initialize it.</li><li>FALSE: Disconnect the GSM module and close the COM Port.</li></ul>
gCOMPort	DINT	The COM port number that connected to the GSM module.
gPinCode	STRING	Enter the PIN code if it is set in a SIM card. (E.g., '0000').

Outputs	Data Type	Description
gIsInitialized	BOOL	<ul><li>TRUE: The COM Port has been opened and the GSM module has been initialized.</li><li>FALSE: Failed to open the COM Port or initialize the GSM module.</li></ul>
gSignalQuality	SINT	<ul><li>0 ~ 31: The higher the value, the better the signal quality.</li><li>99: The signal is unknown or not detected.</li></ul>
gErrorCode	INT	<ul> <li>0: No error.</li> <li>-1: The GSM module is not connected.</li> <li>-2: The SIM card is not inserted properly.</li> <li>-3: The PIN code of the SIM card is wrong.</li> <li>-4: SIM configuration error.</li> <li>-5: Cannot open the specified COM port.</li> </ul>

# The "GSM\_SEND\_SMS" Function:

To send the SMS message via the GSM module.

	Inst_GSM_SEND_SMS	
gExecute	GSM_SEND_SMS	Eno
gPhoneNb		gErrorCode
gSMSText		
gCodePage		

### Notice:

Before using this "GSM\_SEND\_SMS" function, using the "GSM\_OPEN" function block to open the COM Port that is connected to the GSM module, or this function will not work.

Inputs	Data Type	Description			
gExecute	BOOL	Pulse TRUE: to	Pulse TRUE: to send an SMS message.		
gPhoneNb	STRING	Destination-Ad	ddress (i.e., the phone number).		
gSMSText	STRING	The SMS mess	age.		
gCodePage	UDINT	The SMS message.         The code page of the text.         0:       English         866:       Russian         932:       Japanese         936:       Simplified Chinese         950:       Traditional Chinese         950:       Traditional Chinese         Notice:       Only the maximum number of characters will be sent if exceeds         Value       The maximum length of a text message         0       160 characters         non-zero       70 characters			

Outputs	Data Type	Description			
		4: Send SMS succeeds	-1: No GSM module detected.		
	gErrorCode INT	3: Sending SMS	-2: SIM card is not inserted.		
gErrorCode		2: Send SMS is pending	-4: SIM card configuration error.		
genorcode	1: Prepare to send SMS -5: Cannot open the PAC's CON				
		0: No operation.	-6: No recipient's phone number.		
			-7: Send SMS message failed.		

# The "GSM\_READ\_SMS" Function:

To read the SMS message via the GSM module.

	Inst_GSM_READ_SMS	3
gExecute	GSM_READ_SMS	glsNewSMS
= gCodePa	ge gSer	nderPhoneNb
	gRe	ceiveSMSText

### Notice:

Before using this "GSM\_READ\_SMS" function, using the "GSM\_OPEN" function block to open the COM Port that is connected to the GSM module, or this function will not work.

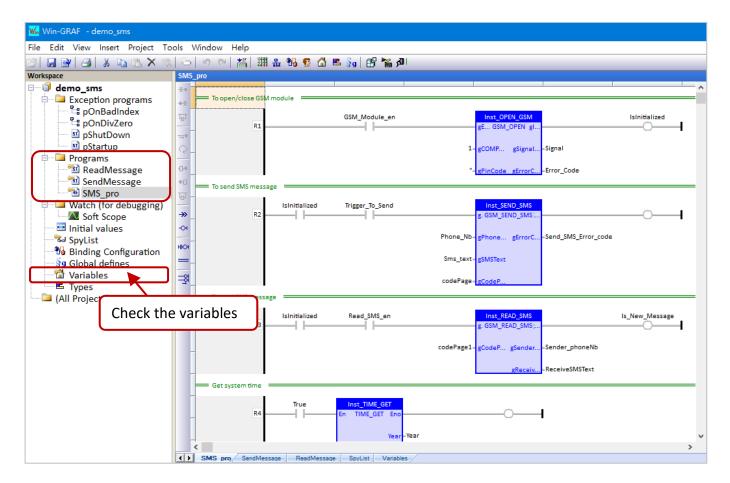
Inputs	Data Type	Description		
gExecute	BOOL	TRUE: Enable the GSM module to read an SMS message. FALSE: Disable the GSM module to read an SMS message.		
gCodePage	UDINT	The code page of the text.0:English866:Russian932:Japanese936:Simplified Chinese950:Traditional Chinese		

Outputs	Data Type	Description
glsNewSMS	BOOL	Pulse TRUE: A new message is coming.
gSenderPhoneNb	STRING	Originating-Address (i.e., the phone number).
gReceiveSMSText	STRING	The SMS message.

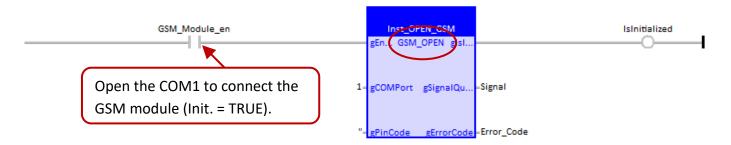
# 16.2 Introduction of the "Demo\_SMS" Project

Download <u>the demo program</u> on the website (**demo\_sms.zip**). Click the menu commands **File - Add Existing Project - From Zip...** in the Workbench to open the project. (Also refer to Section 11.4)

The project includes two ST programs (SendMessage and ReadMessage) and an LD program (SMS\_pro).

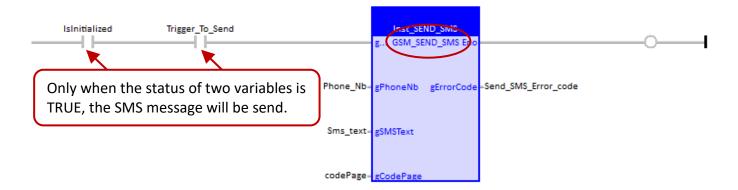


LD Program (SMS\_pro): The "GSM\_OPEN" function (refer to Section 16.1 for more details)



Name	Data Type	Description
GSM_Module_en	BOOL	Set it to TRUE to open the specified COM port to connect and initialize the GSM module (Init. = TRUE).
Signal	SINT	The signal quality of the GSM module.
Error_Code	INT	The error code of the GSM module.
IsInitialized	BOOL	The initialization results of the GSM module.

### The "GSM\_SEAD\_SMS" function (refer to Section 16.1 for more details)



Name	Data Type	Description
Trigger_To_Send	BOOL	Set it to "TRUE" to send an SMS message
Phone_Nb	STRING(255)	The phone number of the recipient. (Init. value: '0900629879')
Sms_text	STRING(255)	The content of the SMS message. (Init. value: 'This message is sent from Win-GRAF PAC')
codePage	UDINT	The code page of the text.
Send_SMS_Error_code	INT	The error code of the SMS message.

## The "GSM\_READ\_SMS" function (refer to Section 16.1 for more details)

IsInitialized Read_SMS_en		Inst_RE	AD_SMS EAD_SMS gl.		ls_New_Message
Only when the status of two variables is TRUE, the SMS message will be read.	codePage1	gCodePage	gSenderP	-Sender_phoneNb	
Set the "Read_SMS_en" to "FALSE" to stop receiving the SMS message.			gReceive	-ReceiveSMSText	

Name	Data Type	Description
Read_SMS_en	BOOL	Sec it to "TRUE" to read an SMS message. (Init value: TRUE)
codePage1	UDINT	The code page of the text. (Init value: UDINT#950)
Sender_phoneNb	STRING(255)	The sender's phone number.
ReceiveSMSText	STRING(255)	The content of the SMS message.
Is_New_Message	BOOL	To check if there is any new SMS message.

## 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;</pre>
```

## (\* 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; *)
```

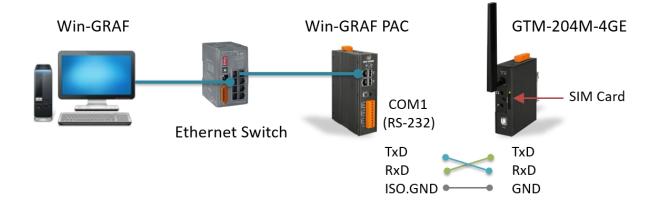
```
Got_New_Message := ReceiveSMSText;
Got_Message_from_who := Sender_phoneNb;
```

```
(* do more operating *)
```

```
end_if;
```

# 16.3 Test for SMS Messaging

In this example, the SMS message is sent or received by using the RPAC-2568M PAC and the <u>GTM-204M-4GE</u> module. First, insert the SIM card into the module and connect to the PAC via the RS-232 port, and then power on.



### Download the project

- 1. Modify the Win-GRAF communication IP to the current IP address of the PAC.
- 2. Enter the number of PAC's COM port that is connected to the GSM module.
- 3. Click the **On Line** button to download the "dmeo\_sms" project to the Win-GRAF PAC. (Also refer to Section 2.3.4 if you are not familiar with the setting.)

22   B BY   3   3 B B X 🕏	.  🏷   ୬ (×   ╫  ╫  ╫ ╫ ╫ ¶ 🥵 🟠 K 😼 🔐 (ff 🌾 📶 )™ 💿 οο οο 🗱 ╫ 🛱 🛚 🐒 🏟
Workspace	SMS_pro]
🖃 🕨 🕨 demo_sms [RUN]	1.8
Exception programs	+3E To open/close GSM module
🖻 🖻 Programs	
📲 🖬 ReadMessage	
📲 SendMessage	GSM_Module_en = TRUE Inst_OPEN_GSM IsInitialized = TRUE
SMS_pro	gEnaGSM_OPEN gIsin
🖃 📮 Watch (for debugging)	
Soft Scope	
🔤 Initial values	+0 Signal Qu Signal = 23
SpyList	
Binding Configuration	
🕺 Global defines	->> - "= <u>gPinCode</u> g <u>ErrorCode</u> = Error_Code = 0

Name	Description
GSM_Module_en	Open the COM1 to connect and initialize the GSM module. (Init. = TRUE).
Signal	The signal quality of the GSM module is "23".
Error_Code	The error code of the GSM module is "0" (no error).
IsInitialized	Successful to open COM1 and initialize the GSM module. Note that if the initialization is failed, reboot the PAC.

### Send an SMS Message

1. Double-click on the parameter to enter the phone number, the text, or the code page. For example,

Name	Description
Phone_Nb	The initial value is set to '0900629879' (or '886900629879').
Sms_text	The initial value is set to 'This message is sent from Win-GRAF PAC'.
codePage	0: to send an SMS message in English.

2. Double-click the "Trigger\_To\_Send" to set it to "TRUE" to send an SMS message.

	[SMS_pro]						
-16++		$\sim$					^
+-]E	To send SMS message	2.					-
Ъ₽ –			/				
	IsInitialized = TRUE	Trigger_To_Send = FALSE		Inst_SEND_SMS		<u> </u>	
Q -				gE GSM_SEND_SMS_Eno			
0+							
+0 -			Phone_Nb = '0900629879'	gPhoneNb gErrorCode	-Send_SMS_Error_code = 0		
Q						<u>,                                     </u>	
			Sms_text = This messag	gSMSText	0: No opera		
-01					3: Sending		
			codePage = 0-	gCodePage	4: Send SM	S succeeds	
ню			_				

### **Receive an SMS Message**

By default, the GTM-204M series send an SMS notification by using the USB port, and you can use the command **AT+QURCCFG?** to check the current setting.

- Using an USB port to receive an SMS message: AT+QURCCFG="urcport", "usbat"
- Using an UART port to receive an SMS message: AT+QURCCFG="urcport","uart1"

To test with PuTTY (also refer to Chapter 16), set both the **Local echo** and **Local line editing** in **Terminal** to **Force on**, fill in the port settings in **Session**, and click the **Save** button.

Category:	Specify the destination you want to connect to <b>2.</b>
Session Logging	Serial line Speed COM7 115200
i⊟- <mark>Teminal</mark> Keyboard Bell	Connection type:
Line discipline options Local echo:	Load, save or delete a stored session Saved Sessions
Local line editing: Auto Force on Force off	Default Settings Load 3.
Connection	Delete

- 1. Enter the command AT+QURCCFG? to check the current setting.
- 2. Enter the command **AT+QURCCFG="urcport","uart1"** to change to use the UART port to receive SMS notifications.
- 3. Enter the command **AT+QURCCFG?** to check the setting again.

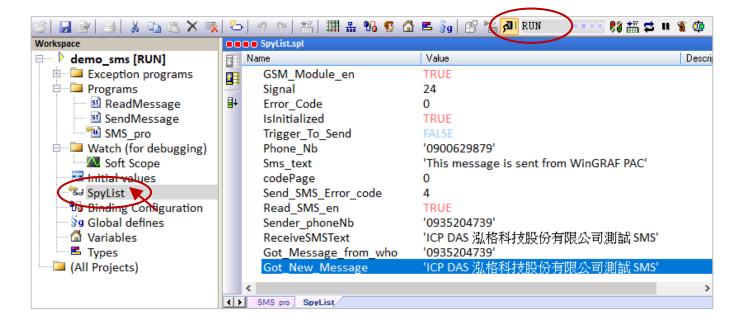
B COM7 - PuTTY	
AT+QURCCFG?	
AT+QURCCFG?	
+QURCCFG: "urcport", "usbat	
OK	
0: Echo mode OF	F
ATE 0 1: Echo mode ON	
ATE 0	
OK	
AT+QURCCFG="urcport","uart	1"
OK	
AT+QURCCFG?	
+QURCCFG: "urcport", "uartl	"
ok	

Please send an SMS message in Chinese from your cell phone (e.g., ICP DAS 泓格科技股份有限公司 測試 SMS) to the phone number of the SIM card.

	SMS_pro]				
-16-0					
+H	To read SMS message				
पुरु	_				
	Interviewed TRUE Read SMC on TRUE				In New Manager - FALOE
	IsInitialized = TRUE Read_SMS_en = TRUE		Inst_READ_SMS		Is_New_Message = FALSE
Q.			g GSM_READ_SMS gl		
		_			
-0+			· · · · · ·		
+0-	Set it to "FALSE" to sto	p codePage1=950	gCodePage gSenderP	-Sender_phoneNb = '09	
	receving the message.				
묘					
<i>→</i> >	_		gReceive	=ReceiveSMSText = 'ICP	J

Name	Description
codePage1	950: Chinese ; 0: English
Sender_phoneNb	The phone number of the sender.
ReceiveSMSText	E.g., ICP DAS 泓格科技股份有限公司測試 SMS
Is_New_Message	TRUE: An new SMS message is coming.

Moreover, double-click the **SpyList** to open the variables list for the test of sending or receiving an SMS message.



Visit the web pages for more related FAQ:

https://www.icpdas.com/en/faq/index.php?page=2&model=GTM-204M-4GE#399

# Appendix A Data types and Ranges

Users can specify the data type of variables in the Variables Area (refer to <u>Section 2.2.1</u>) or in the Variables window (refer to <u>Section 2.2.2</u>).

😑 🚮 Global variables	
PAC_Year	DINIT
PAC_Month	D BOOL 🔨
PAC_Day	D BYTE
PAC_WeekDay	
PAC_Hour	
PAC_Minute	D
PAC_Second	D LREAL
<	LWORD

Below are the available basic data types and ranges:

Data types	Size in Bits		Range of Values	
BOOL (*)	Boolean		TRUE or FALSE	
STRING (*)	Charact	er string	A max. of 255 characters	
SINT		Signed small integer	-128 to +127	
USINT	8-bit		0 to 1255	
BYTE		Unsigned small integer	0 to +255	
INT		Signed integer	-32768 to +32767	
UINT	16-bit			
WORD		Unsigned integer	0 to +65535	
DINT (*)		Signed double integer	-2147483648 to +2147483647	
UDINT		Unsigned double integer	0 to +4294967295	
DWORD	32-bit			
REAL (*)		Floating point	±3.4×10 <sup>-38</sup> to ±3.4×10 <sup>38</sup>	
TIME (*)		Time of day	T#0ms to T#23h59m59s999ms	
LINT		Signed long integer	-2 <sup>63</sup> to +(2 <sup>63</sup> -1)	
LREAL	- 64-bit	Floating point	±1.7×10 <sup>-308</sup> to ±1.7×10 <sup>308</sup>	
ULINT			0.4	
LWORD		Unsigned long integer	0 to +(2 <sup>64</sup> -1)	
Note: "ULINT" and "LWORD" are not supported for Win-GRAF PAC.				

(\*) stands for the commonly used data type.

# Appendix B Troubleshooting while On-Line the PAC

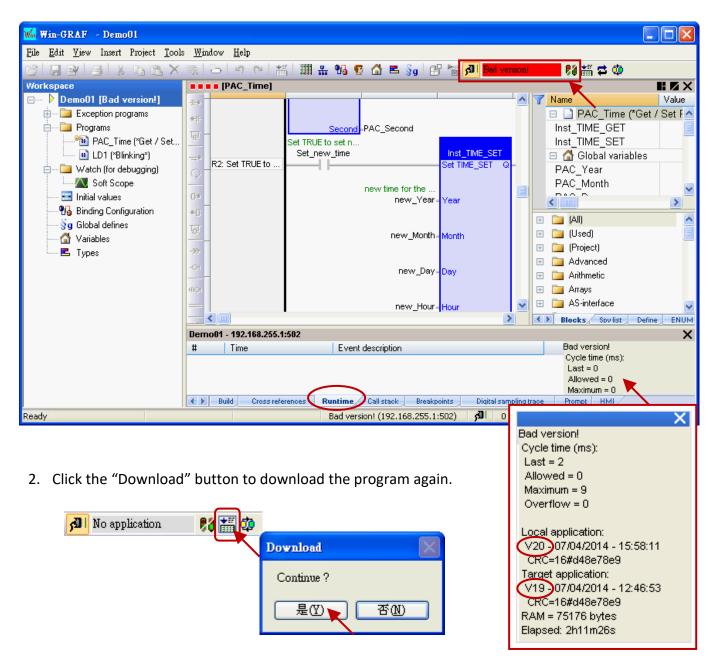
As the screenshot below, if the error message is showing up after connecting to the Win-GRAF PAC, follow the instruction to solve the problem.

#### • 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.

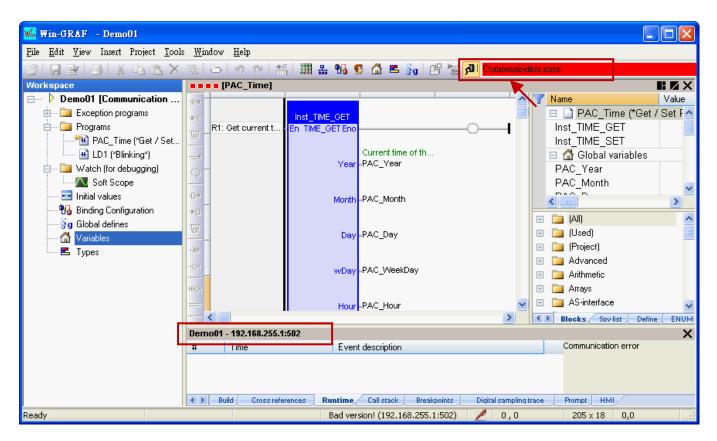


3. The "RUN" message means that the program is working properly.

📶 RUN 🛛 💿 🗆 🐉 🏭 😅 💷 🐒 🥸

### • The "Communication error" error message:

A communication failure occurs between the PC and the PAC.



### 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.
- Make sure the Win-GRAF communication IP address is set to the current IP address of the PAC. (refer to <u>Section 2.3.4</u>, the IP address is "192.168.255.1:502" in this example).
- 3. Make sure both IP addresses of the PC and PAC are on the same network segment.

# Appendix C Allows the Win-GRAF to Connect via PAC's Serial Port

Generally, the Win-GRAG can connect to the PAC to download projects or perform debugging via the Ethernet port. If the user needs to enable a PAC's serial port (normally RS-232 or RS-485) to allow Win-GRAF to connect with it, follow the instructions below:

### Method 1:

Once the Win-GRAF PAC is booted, it will try to read the "Extra\_Ports.txt" file from the "/System\_Disk/ Win-GRAF/" directory. If you want to enable the COM2 of the PAC, enter the text as follows:

COM2:115200,N,8,1

It means to enable the COM2 and the Baud Rate is 115200 bps

To upload the "Extra\_Ports.txt" file to the "/System\_Disk/ Win-GRAF/" directory by using the SSH tool (also refer to Section 13.2) and then reboot the PAC.

SmarTTY - 192.168.79.10
File Edit View SCP Tools Help
<ul> <li>Download a file</li> <li>Manageme</li> <li>Manageme</li> <li>Upload a file</li> <li>Download a directory</li> <li>Last login</li> <li>Upload a directory</li> <li>Upload a directory</li> <li>Upload a directory</li> <li>Upload a directory</li> <li>Support</li> </ul>
🖆 ubuntu@RPAC-2658M:~ 🗙 🕂 🖋 🎓
Upload a File with SCP  Local file name: Remote directory: C:\Users\Janice\Desktop\Extra_Ports.txt.txt  Upload Cancel
Browse File System on Remote Host /System_Disk/Win-GRAF Also check the content of the file.
Extra_Ports.txt         SmarTTY - 192.168.79.10           Image: t5.upd         File Edit View SCP Tools Help           1         COM2:115200, N, 8, 1
Open Cancel

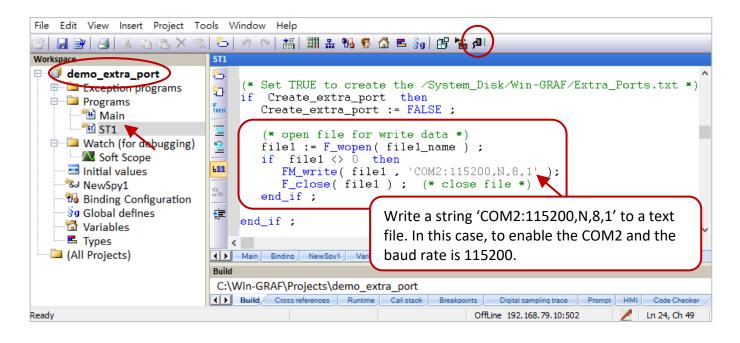
# Disable the COM Port:

If you want to disable the COM port, simply remove the "Extra\_Ports.txt" file in the "/System\_Disk/ RPAC-2658M User Manual, V1.0, Mar. 2021 by ICP DAS 267 Win-GRAF/ " directory and then reboot the PAC.

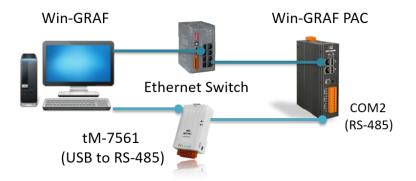
SmarTTY - 192.168.79.10	—		×
File Edit View SCP Tools Help			
Last login: Fri Feb 19 15:41:42 2021 from 192.168. ubuntu@RPAC-2658M:~\$ cd ubuntu@RPAC-2658M:/home\$ cd ubuntu@RPAC-2658M:/\$ cd System Disk	79.22	0	^
ubuntu@RPAC-2658M:/System_Disk\$ cd Win-GRAF ubuntu@RPAC-2658M:/System_Disk/Win-GRAF\$ ls Extra_Ports.txt t5.cod t5.upd			
ubuntu@RPAC-2658M:/System_Disk/Win-GRAF\$ rm Extra_			
<pre>rm: remove write-protected regular file 'Extra_Por ubuntu@RPAC-2658M:/System_Disk/Win-GRAF\$ ls t5.cod t5.upd</pre>	ts.tx	t'? y	
ubuntu@RPAC-2658M:/System_Disk/Win-GRAF\$			~
🖆 ubuntu@RPAC-2658M:/System_I 🗙 🕂 🥖 📂 🌞			

### Method 2:

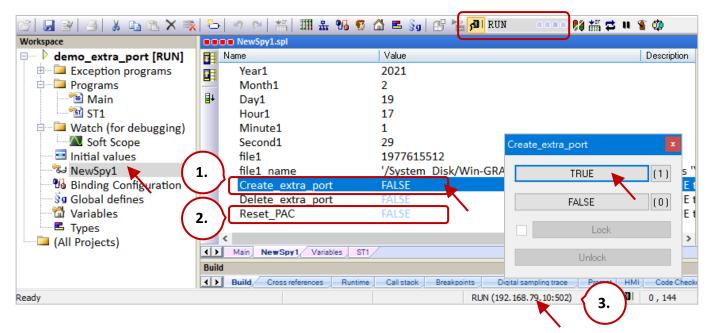
- Download <u>the demo program</u> on the website (demo\_extra\_port.zip). Click the menu commands File - Add Existing Project - From Zip... in the Workbench to open the project.
- 2. Double-click the ST1 program to modify the COM port and baud rate (e.g., 'COM2:115200,N,8,1') of the PAC.
- 3. Click the On Line button to download the project to the PAC via Ethernet.



## Test the Project



- After connecting to the Win-GRAF PAC, set the "Create\_extra\_port" to "TRUE" in the variables list. In this example, the "Extra\_Ports.txt" file will be added into the /System\_Disk/Win-GRAF/ directory and write a string 'COM2:115200,N,8,1' to the file.
- 2. Set the "Reset\_PAC" to "TRUE" to reboot the PAC automatically and apply the setting.



3. Disconnect the PAC and modify the Win-GRAF communication settings. Double-click the IP address on the status bar and then click the ... button. Select the **Serial link** radio button and choose the available COM port of the PC.

Communication Settings	×	
T5 Runtime	4. OK Communication Settings	×
192.168.79.10:502	Ethernet TCP/IP	ОК
192.168.79.11:502 192.168.79.37:502 192.168.79.38:502	5. Port number: The PC (Device Manager) no the same Baud Rate as the	
	Serial link     PC port:     Baudrate:     115200	
	Parity: None ~ Stop bits: 1 ~	

## **Disable the COM Port:**

- 1. Set the "Delete\_extra\_port" to "TRUE" to remove the Extra\_Ports.txt file in the "/System\_Disk/ Win-GRAF/" directory.
- 2. Set the "Reset\_PAC" to "TRUE" to reboot the PAC.

🖌 Win-GRAF - demo_extra_port					
File Edit View Insert Project Tools Window Help					
🖂 🔒 🖹 🦂 🕹 🖡 🛍 🗙 🔻	8-	🔊 (e   🏭   🎹 🏪 😘	😨 🗳 🛋 🛐 🗄 🕍	📶 RUN 🛛 🗆 🕬 👬 🕯	3 U 🐒 🧔
Workspace		NewSpy1.spl			
<ul> <li>demo_extra_port [RUN]</li> <li>Exception programs</li> <li>Programs</li> <li>Main</li> <li>ST1</li> <li>Watch (for debugging)</li> <li>Soft Scope</li> <li>Initial values</li> <li>NewSpy1</li> <li>Binding Configuration</li> <li>Global defines</li> <li>Variables</li> <li>Types</li> <li>(All Projects)</li> </ul>	∎ ∎+ ∎+ Build		14 1977615512 '/System_Disk/V FALSE FALSE FALSE	FALSE	Description (1) ited as " t TRUE t t TRUE t t TRUE t
Ready	<b>&lt; &gt;</b>	Build Cross references Rur	ntime Call stack Breakpoi	RUN (COM5: 115200,N,8,1)	t HMI Code Checks

### Note:

When using method2 to download the "Extra\_Ports.txt" file to the PAC, the file cannot be overwritten by using method1.

# Appendix D Pin Assignment of PAC's Serial Ports

## <u>RPAC-2658M</u>:

Pin assignment of COM1, COM2, and COM3.

	Pin	Signal	Description
	1	TxD	
	2	RxD	Console (RS-232)
	3	GND	(10 202)
	4	TxD	
	5	RxD	<b>COM1</b> (RS-232)
	6	ISO.GND	(10 202)
	7	D+	
	8	D-	<b>COM2</b> (RS-485)
	9	ISO.GND	
	10	F.GND	Frame Ground

