Win-GRAF Workbench Software

User Manual

Ver. 3.0

This User Manual applied to Win-GRAF Workbench 10.x and after version software.

ICP DAS WinCE-based Win-GRAF PAC includes:

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

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

TECHNICAL SERVICE
If you have any questions, please feel free to contact us via email at: service@icpdas.com

REVISION

Version	Date	Description	Author
3.0	06.23.2022	OPC UA Server	Lynn Tang
	01.03.2022	Version Revision for 10.x and after Workbench	

Contents

1 Software Installation & Hardware Setting

1.1 Installing the Win-GRAF Workbench

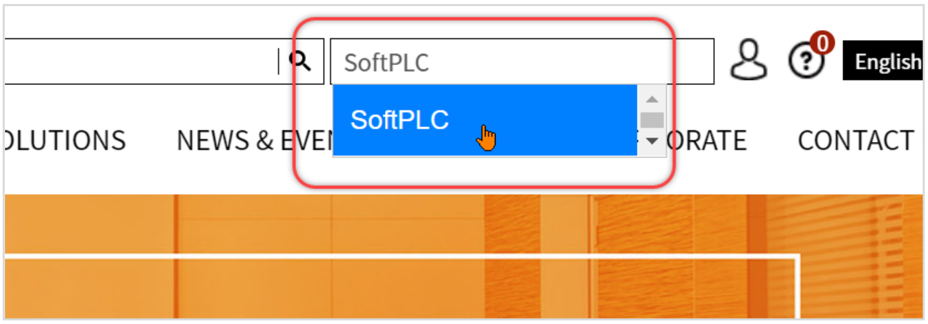
Before installing the Win-GRAF Workbench, check the installation environment of your PC.

System requirements:

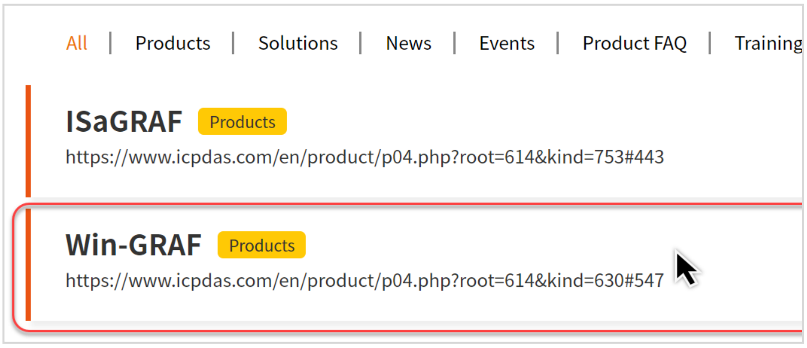
- **O.S.**: Windows 7, Windows 8, Windows 10 (32-bits or 64-bits)
- **Microsoft .Net Framework 3.5** (Download it from the Microsoft website: http://www.microsoft.com/en-US/download/details.aspx?id=22)
- **RAM:** 1 GB minimum (Recommended: 2 GB or more)
- **Available hard-disk space:** 200 MB minimum

Installation Steps:

Download the Win-GRAF Workbench Installation file from ICP DAS website (https://www.icpdas.com). Enter *'SoftPLC'* into the Tag search box, and click it.



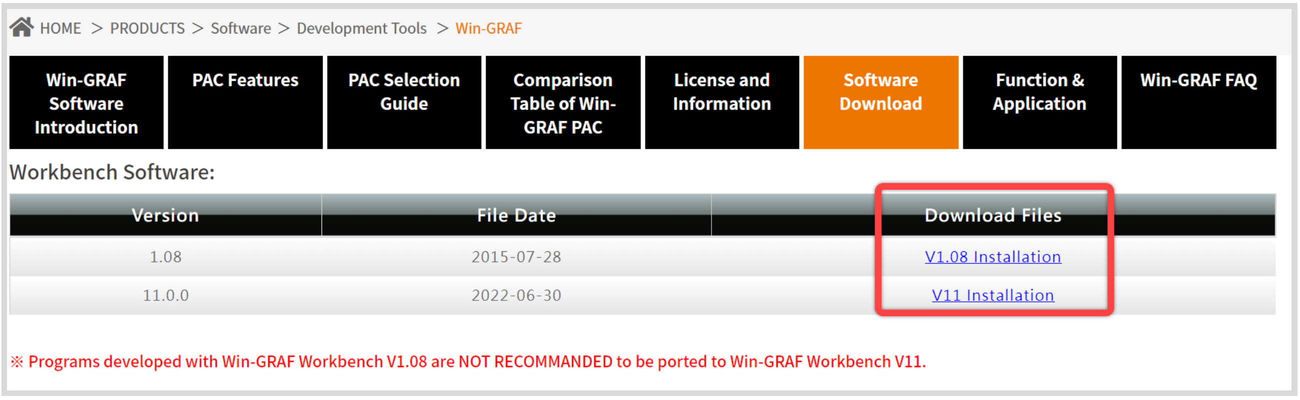
Click "Win-GRAF" and later opens a new Tab.


All | Products | Solutions | News | Events | Product FAQ | Training

ISaGRAF Products
https://www.icpdas.com/en/product/p04.php?root=614&kind=753#443

Win-GRAF Products
https://www.icpdas.com/en/product/p04.php?root=614&kind=630#547

This figure shows the Titles of Win-GRAF. Click "Software Download" and select the version of "Win-GRAF Workbench" to install.

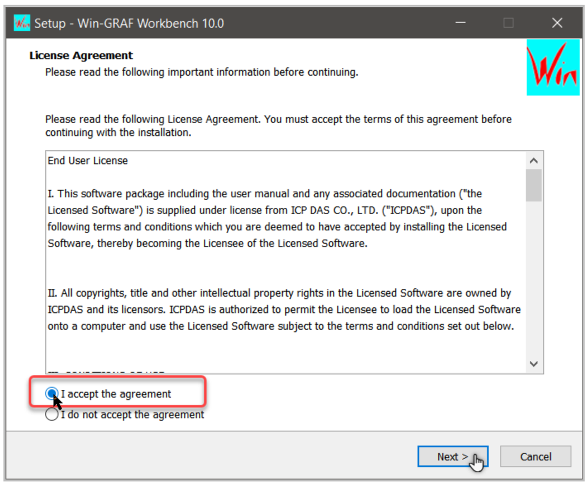

HOME > PRODUCTS > Software > Development Tools > Win-GRAF
Win-GRAF Software Introduction
PAC Features
PAC Selection Guide
Comparison Table of Win-GRAF PAC
License and Information
Software Download
Function & Application
Win-GRAF FAQ

Workbench Software:

Version	File Date		Download Files
1.08	2015-07-28		V1.08 Installation
11.0.0	2022-06-30		V11 Installation

※ Programs developed with Win-GRAF Workbench V1.08 are NOT RECOMMANDED to be ported to Win-GRAF Workbench V11.

Accept the License Agreement and click "Next". There shows an information dialog, please read and click “Next”.


Setup - Win-GRAF Workbench 10.0

License Agreement
Please read the following important information before continuing.

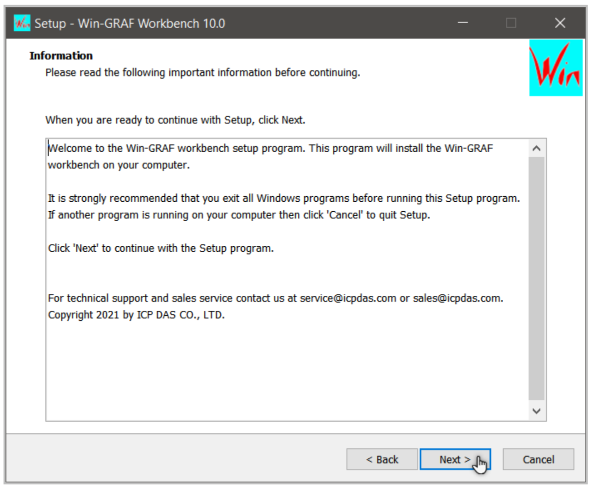
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

End User License

I. This software package including the user manual and any associated documentation ("the Licensed Software") is supplied under license from ICP DAS CO., LTD. ("ICPDAS"), upon the following terms and conditions which you are deemed to have accepted by installing the Licensed Software, thereby becoming the Licensee of the Licensed Software.

II. All copyrights, title and other intellectual property rights in the Licensed Software are owned by ICPDAS and its licensors. ICPDAS is authorized to permit the Licensee to load the Licensed Software onto a computer and use the Licensed Software subject to the terms and conditions set out below.

[✓ I accept the agreement]
I do not accept the agreement


Setup - Win-GRAF Workbench 10.0

Information
Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

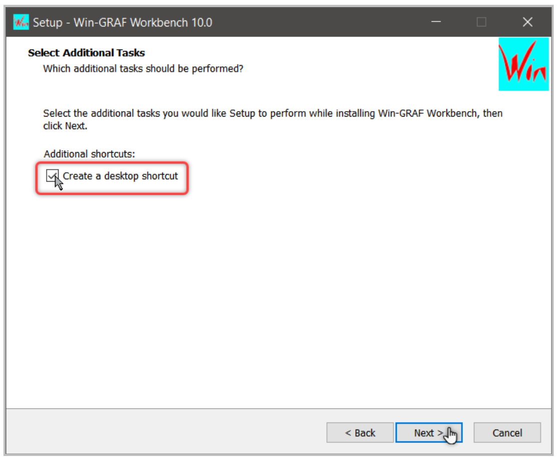
Welcome to the Win-GRAF workbench setup program. This program will install the Win-GRAF workbench on your computer.

It is strongly recommended that you exit all Windows programs before running this Setup program. If another program is running on your computer then click 'Cancel' to quit Setup.

Click 'Next' to continue with the Setup program.

For technical support and sales service contact us at service@icpdas.com or sales@icpdas.com.
Copyright 2021 by ICP DAS CO., LTD.

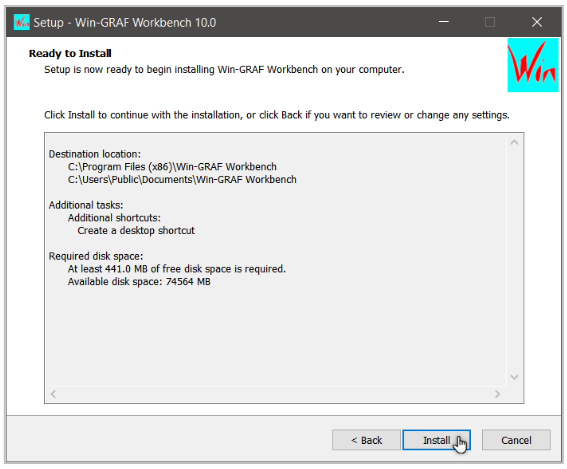
Check "Create a desktop shortcut" to add a shortcut icon on your desktop, then click "Next" to continue. Click "Install" to begin installing the Win-GRAF Workbench.


Setup - Win-GRAF Workbench 10.0

Select Additional Tasks
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing Win-GRAF Workbench, then click Next.

Additional shortcuts:
☑ Create a desktop shortcut


Setup - Win-GRAF Workbench 10.0

Ready to Install
Setup is now ready to begin installing Win-GRAF Workbench on your computer.

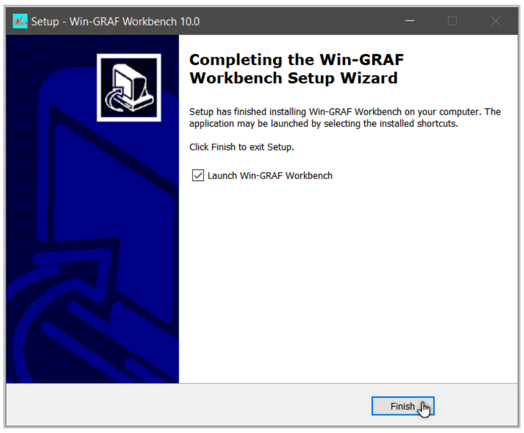
Click Install to continue with the installation, or click Back if you want to review or change any settings.

```
Destination location:
    C:\Program Files (x86)\Win-GRAF Workbench
    C:\Users\Public\Documents\Win-GRAF Workbench

Additional tasks:
    Additional shortcuts:
        Create a desktop shortcut

Required disk space:
    At least 441.0 MB of free disk space is required.
    Available disk space: 74564 MB
```

Finish the installation and Launch the Workbench.


Setup - Win-GRAF Workbench 10.0

Completing the Win-GRAF Workbench Setup Wizard

Setup has finished installing Win-GRAF Workbench on your computer. The application may be launched by selecting the installed shortcuts.

Click Finish to exit Setup.

☑ Launch Win-GRAF Workbench

1.2 Run the Win-GRAF Workbench

After the installation process has successfully completed, the workbench is ready to get started by clicking the *"Win-GRAF Workbench ##.##"* in the start menu.

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:

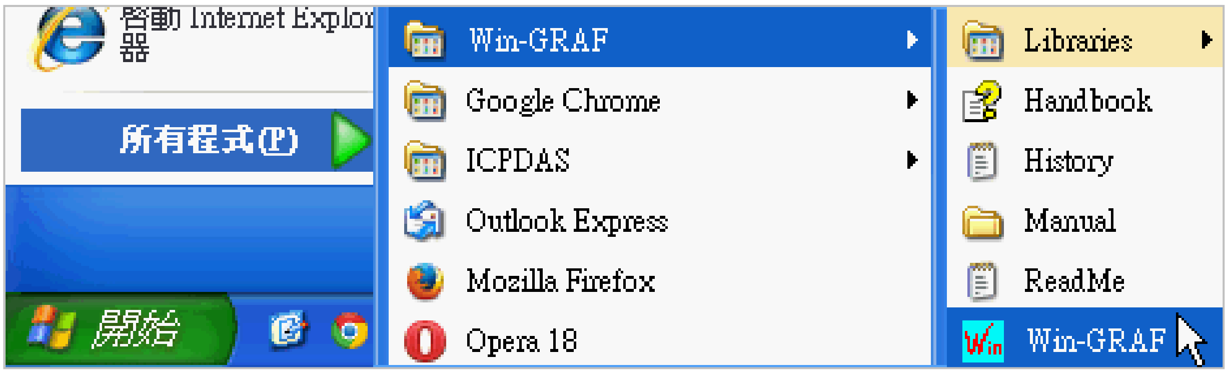
1. Applications are limited to 40 I/O tags.

2. The code generated by the compiler stops after 15 minutes.

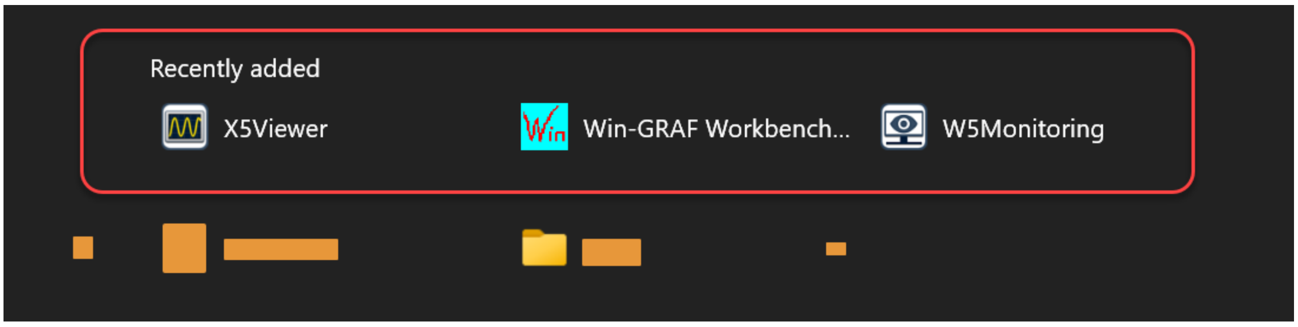
3. The simulation stops after 15 minutes.

Open the Windows Start menu, click on "Win-GRAF" folder and "Win-GRAF" to open this software.

Capture 1: Windows 7

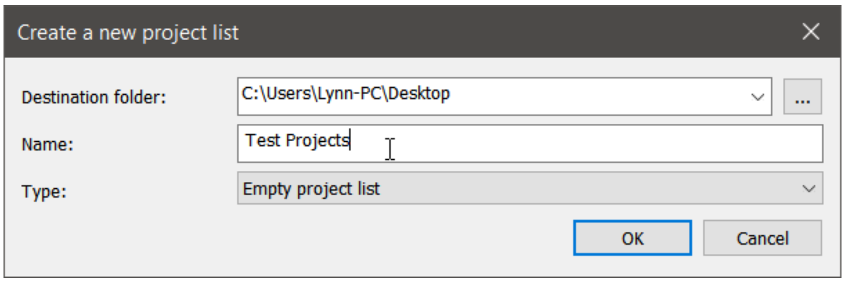


Capture 2: Windows 8 & 10



1.2.1 Win-GRAF Operating Environment

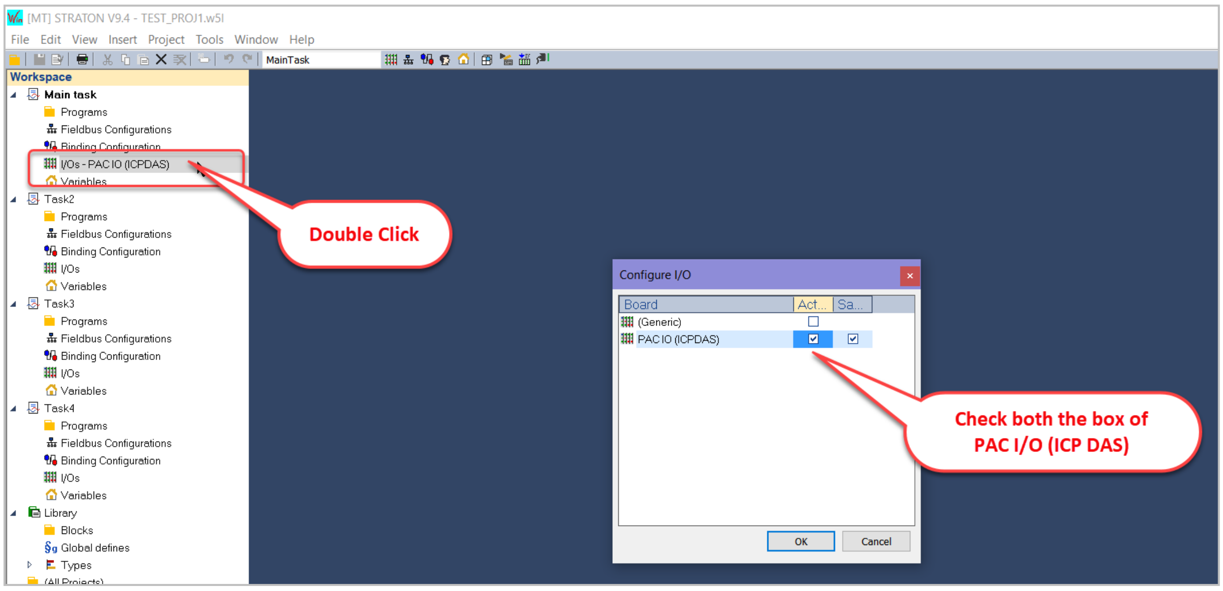
Find the execution route from previous section. Run Win-GRAF Workbench and follow the steps below to understand this development tool.


Create a new project list

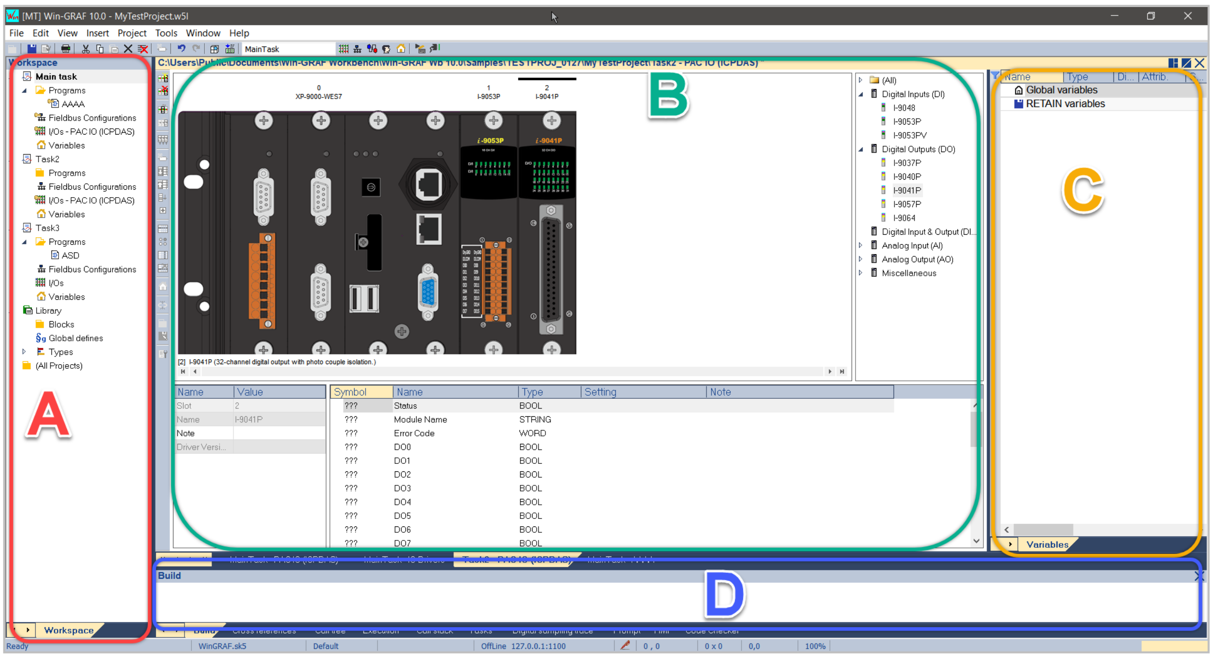
Destination folder: C:\Users\Lynn-PC\Desktop

Name: Test Projects

Type: Empty project list


[MT] STRATON V9.4 - TEST_PROJ1.w5l

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



[2] I-9041P (32-channel digital output with photo couple isolation.)

Name	Value
Slot	2
Name	I-9041P
Note	
Driver Versi...	

Symbol	Name	Type	Setting	Note
???	Status	BOOL		
???	Module Name	STRING		
???	Error Code	WORD		
???	DO0	BOOL		
???	DO1	BOOL		
???	DO2	BOOL		
???	DO3	BOOL		
???	DO4	BOOL		
???	DO5	BOOL		
???	DO6	BOOL		
???	DO7	BOOL		

A. Workspace: the tasks name and it associated programs, fieldbus configuration, Spylist and global variables are shown in the workspace in the left-hand window of the Workbench. The content of the items listed in the workspace are shown in the main window by double clicking the item. The project workspace is stored in a file with the format ".W5L". It basically stores the list of task folders and some configuration data. The workbench creates for each task a separated folder for storing the program source code, Fieldbus configuration, IO settings, etc.

Hint: It is possible, to copy items from task to another task within the workspace. This can be done by either selecting the copy command from the menu bar entries, using the shortcuts CTRL+C and CTRL+V or via drag & drop.

B. Editor Area: double-click the item listed in the "A. Workspace" to display the editing area. Users can call out the Main PAC (the default is XP-9000-WES7) with the relative PAC I/Os in this part to configure all by adding or deleting the product pictures. For those who is used to edit in one of the five IEC-61131 defined PLC languages can also add and start to edit program in the workspace. Beneath window shows PAC I/O information include data type that are preloaded into the library of the Win-GRAF workbench.

C. Variable Tab: the declaration of global and local defined variables (include bind or unbind) are displayed in the top-right area of the Workbench main window. The variable editor is a grid tool that enables you to declare all variables of the application. Variables in the editor are sorted by groups:
- Global variables.
- "Retain" non volatile global variables.
- I/O variables (each I/O device is a group).
- variables local to a program (including in and out parameters in case of a UDFB).

D. Output Message: shows the compiler messages and if connected to the runtime, all the messages generated by the runtime and the status of each task.

Note: To change the language, click the "Help" in the Toolbar and select "Language". After the Workbench automatically restart again, the language will be switch to the one you choose.

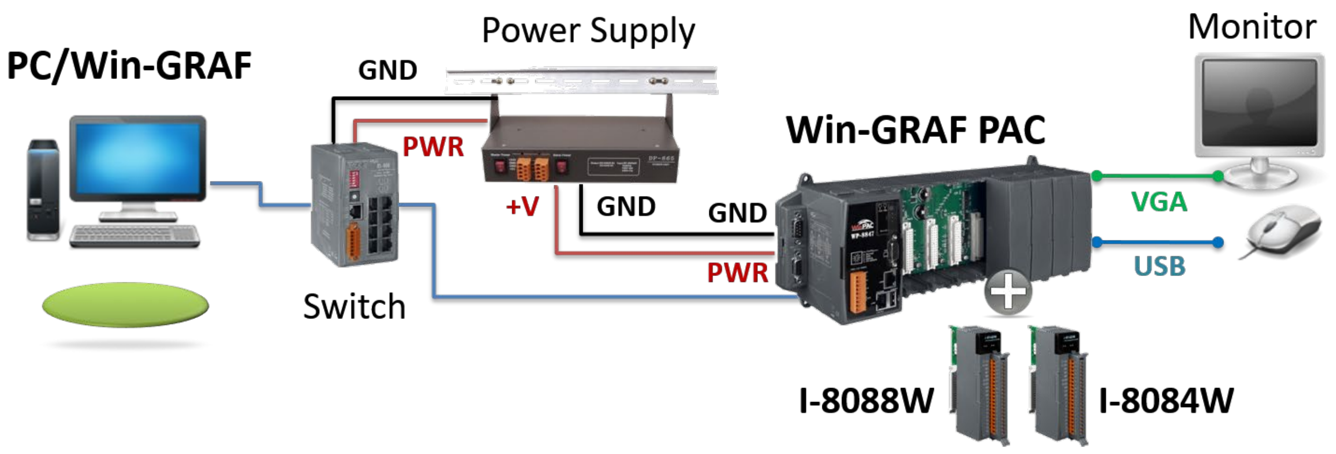
** Multi-language support is determined by the language of the OS. i.e. When the OS is in German, the Workbench language can switchover smoothly between German and English.*

1.3 Setting IP Address of the Win-GRAF PAC

For Runtime Version users only

Following figure shows how to complete wiring and set up the PAC IP. Take WinPAC (WP-5xx8-CE7, WP-8x48, WP-8x28-CE7) for example:

Hardware Wiring Diagram



Note: *PC/Win-GRAF IP and the PAC IP must be in the same network segment.*

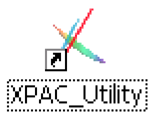
For example,

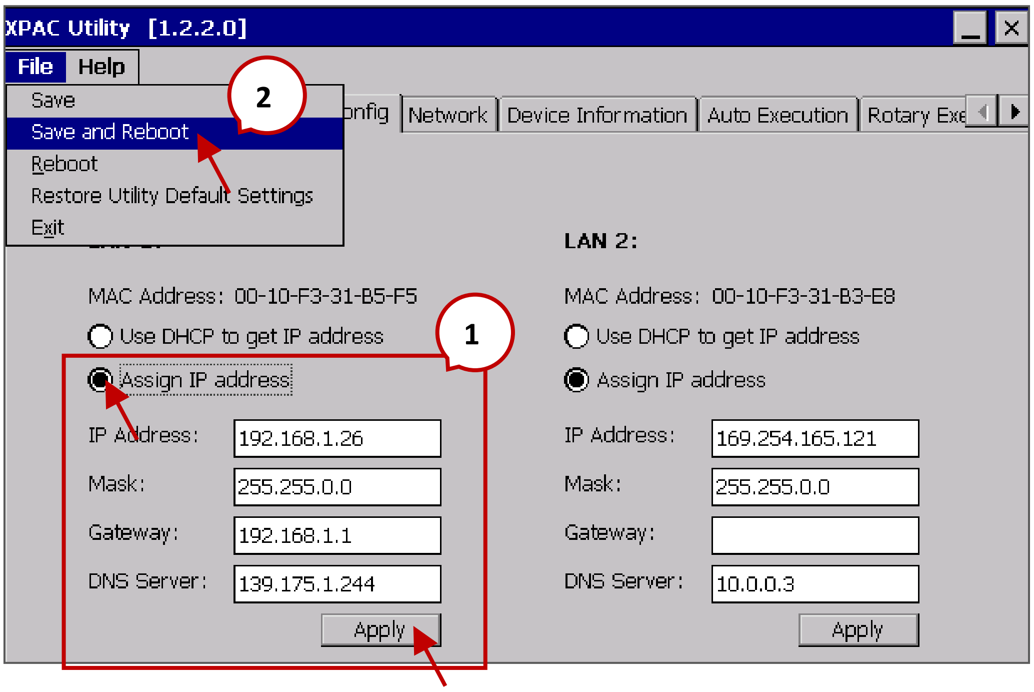
IP of PC: "192.168.1.20" (Mask: 255.255.255.0)

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

PAC Side Setting

Download and run PAC_Utility. Click ①"Assign IP address" and enter the IP address for LAN1 (or LAN2) in the "IP Config" Tab. Next, click the "Apply" button. Go to Toolbar, click File and choose ②"Save and Reboot" to apply settings.





LAN 2:

MAC Address: 00-10-F3-31-B5-F5

○ Use DHCP to get IP address

● Assign IP address

IP Address: 192.168.1.26

Mask: 255.255.0.0

Gateway: 192.168.1.1

DNS Server: 139.175.1.244

MAC Address: 00-10-F3-31-B3-E8

○ Use DHCP to get IP address

● Assign IP address

IP Address: 169.254.165.121

Mask: 255.255.0.0

Gateway:

DNS Server: 10.0.0.3

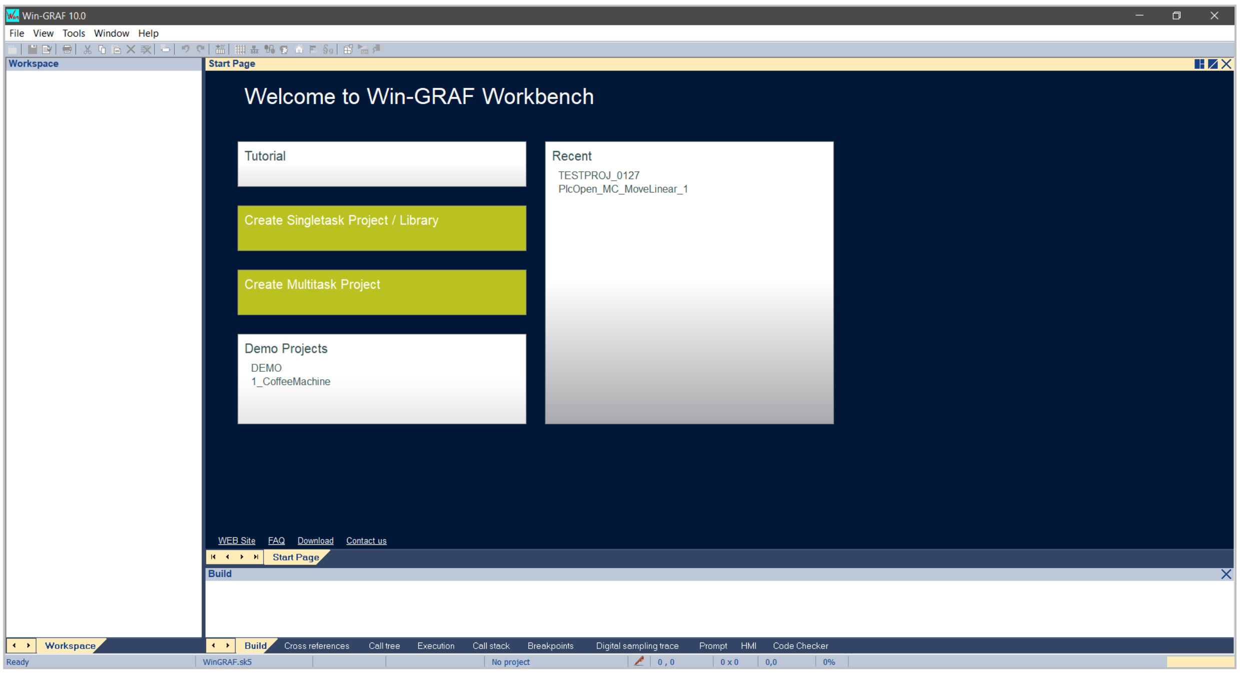
2 A Simple Win-GRAF Program

2.1 Creating a Project

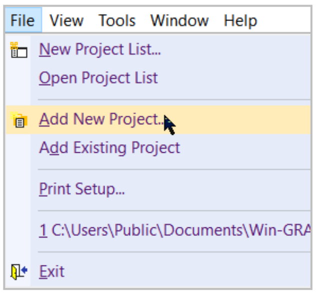
2.1.1 Create a Single-tasking Project

The basic steps for creating a single-tasking PLC project using Win-GRAF workbench:

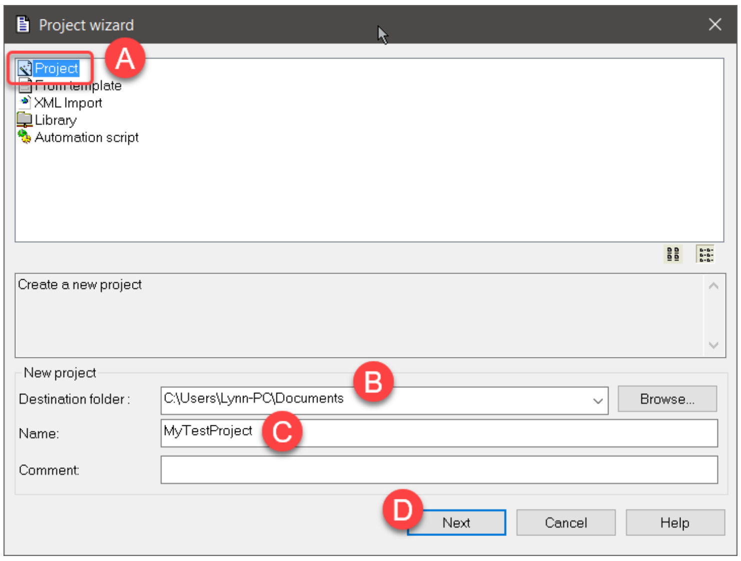
1. Run the Win-GRAF workbench. The *"Workspace"* is empty and the *"Start Page"* list the available manuals, demo projects and the recent opened projects. In addition, two green command boxes are listed for directly creating a single- or multi-task project.


Win-GRAF 10.0
File View Tools Window Help
Workspace
Start Page
Welcome to Win-GRAF Workbench
Tutorial
Create Singletask Project / Library
Create Multitask Project
Demo Projects
DEMO
1_CoffeeMachine
Recent
TESTPROJ_0127
PlcOpen_MC_MoveLinear_1
WEB Site FAQ Download Contact us
Build
Cross references Call tree Execution Call stack Breakpoints Digital sampling trace Prompt HMI Code Checker
Ready WinGRAF.sk5 No project

2. To open the project wizard for single-task project click either the green command button in the "*Start Page*" or go to "*File\Add New Project...*"



3. Click the "Project", select the destination folder and enter the project name. Click "Next".



Create a new project

New project

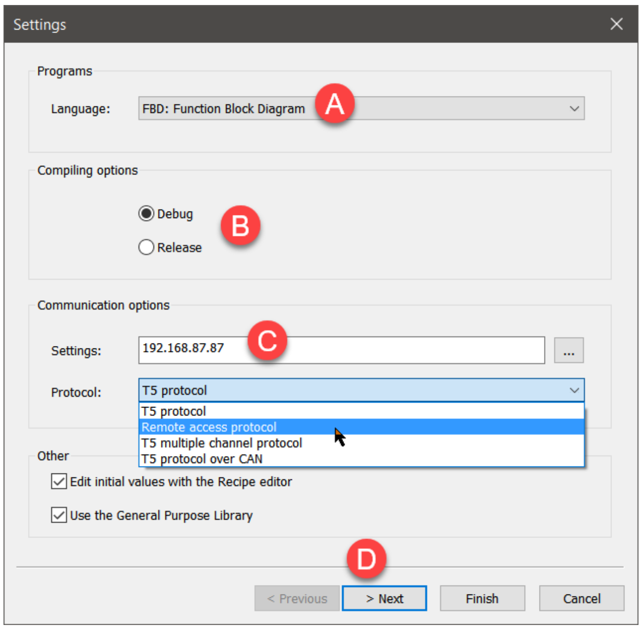
Destination folder : C:\Users\Lynn-PC\Documents

Name: MyTestProject

Comment:

Note: For the project several folders and files are being generated. For easier maintains it is therefore suggested to create a new folder for the project or use an empty destination folder.

4. Set the programming language of the first program (POU), compiling option, enter the IP address of the remote device and leave the protocol to "Logic Service". All settings still can be modified after the project has been created. Although the wizard allows you to select only one programming language for the first program, additional programs for different programming language can be added later on. Click "Next" to continue the configuration.



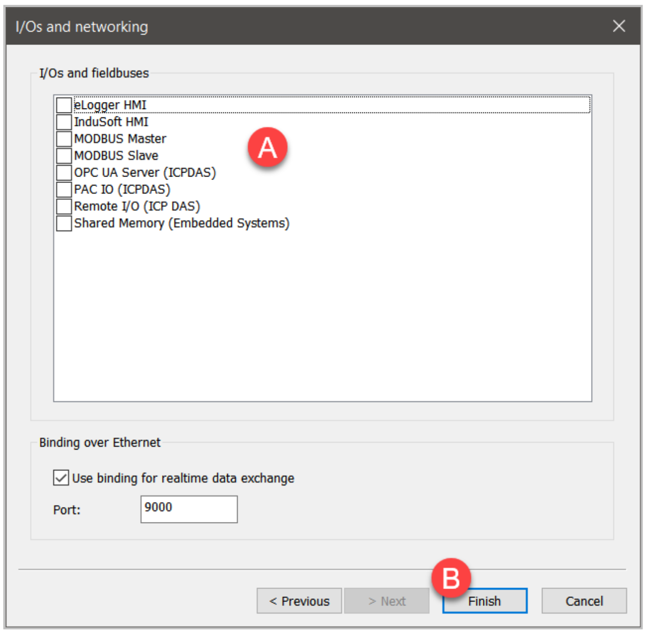
Programs
Language: FBD: Function Block Diagram

Compiling options
(● Debug) ○ Release

Communication options
Settings: 192.168.87.87
Protocol: T5 protocol
T5 protocol
Remote access protocol
T5 multiple channel protocol
T5 protocol over CAN

Other
☑ Edit initial values with the Recipe editor
☑ Use the General Purpose Library

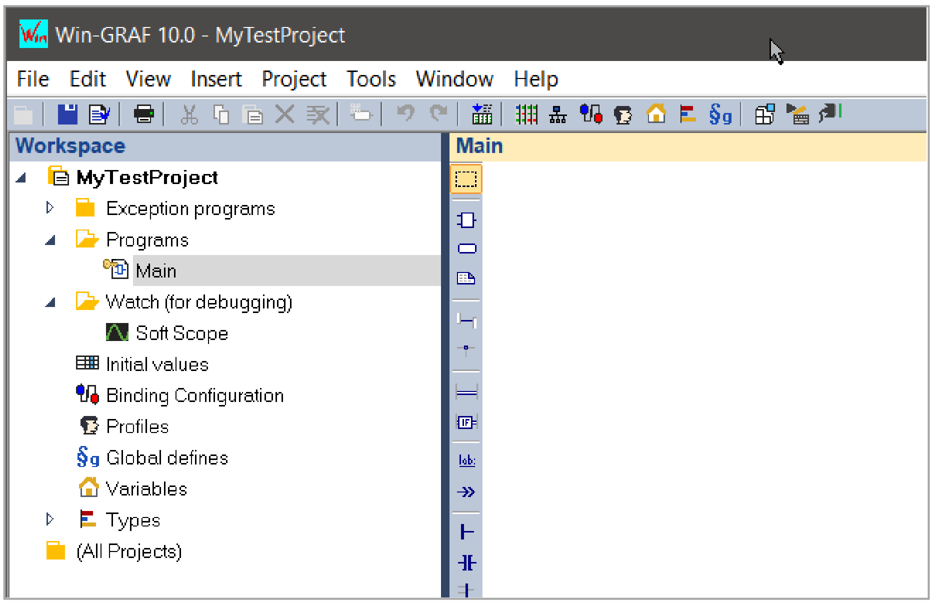
5. Select the additional components to use. If you are unsure which components to add to the project at the current stage, then leave the field unchecked. They can be manually added at any time during the project development. Do not change the *"Binding"* setting. Confirm the setting by clicking "*Finish*". A new project with the current setting will be generated.



I/Os and fieldbuses
☐ eLogger HMI
☐ InduSoft HMI
☐ MODBUS Master
☐ MODBUS Slave
☐ OPC UA Server (ICPDAS)
☐ PAC IO (ICPDAS)
☐ Remote I/O (ICP DAS)
☐ Shared Memory (Embedded Systems)

Binding over Ethernet
☑ Use binding for realtime data exchange
Port: 9000

6. Following figure shows the *"Workspace"* setting of a new single-tasking project. Double click the *"Main"* item to open the main programming editor.

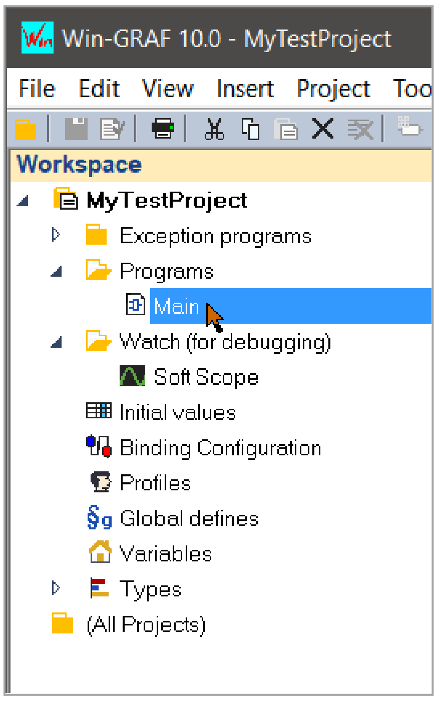


2.2 Edit a Program

The main focus of this section is to give a brief introduction to the user interface of the workbench and show how to use the tools provided by the workbench to edit the logic for a PLC program. The Win-GRAF Workbench supports all the five PLC programming languages defined by IEC-61131. For each programming language a separated editor is provided.

Basic procedure to declare variables and edit a function using the FBD programming editor:

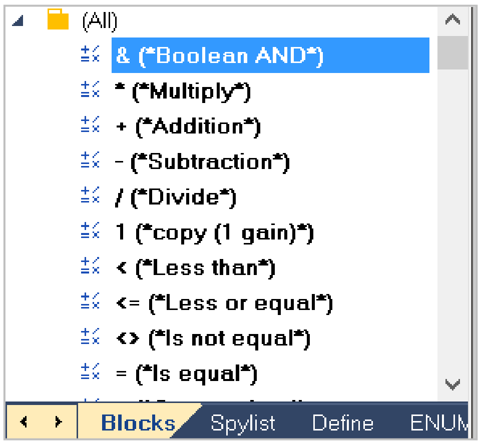
1. Double click the name of the 'Main' program in the workspace to open the program editor:



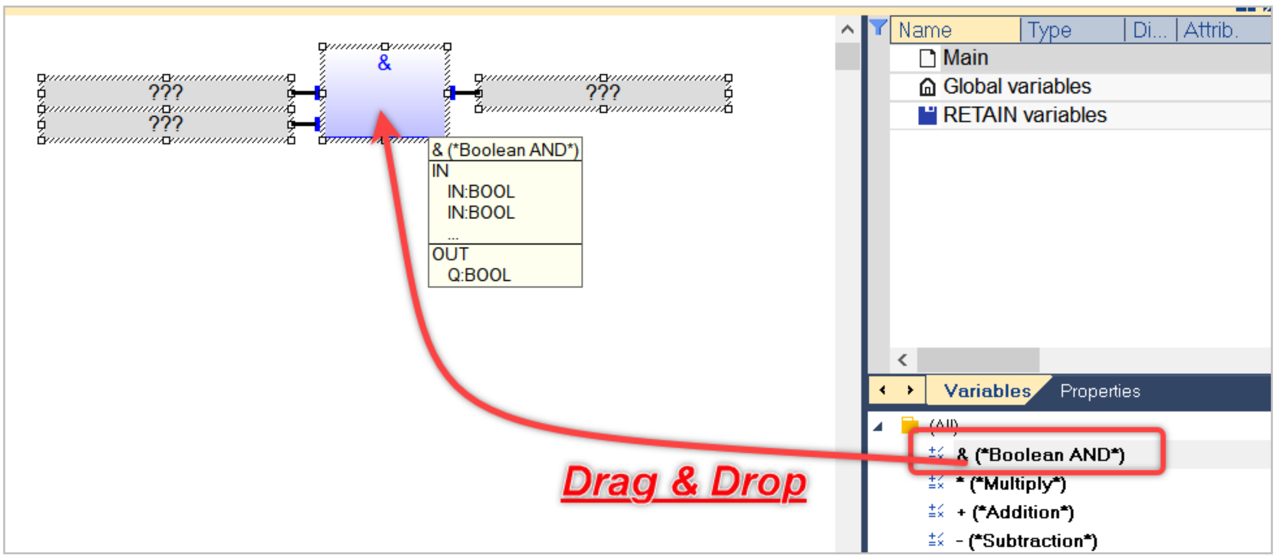
Note: Double click the program name and not the icon.

The PLC logic can now be edited using the FBD language. In the following steps demonstrate how to add a function block to the editor and declare its in- and output variables.

2. Add the "AND" function block:
   All the supported function blocks are listed in the "Blocks" tab of the Info window on the right. If the Info window is not visible, then go to "View/Infos Tab2" in the menu bar to display the window.
   The function blocks are listed according to different categories. The "(All)" category list all the supported function blocks.
   Click the "(All)" tree node to display all function blocks.

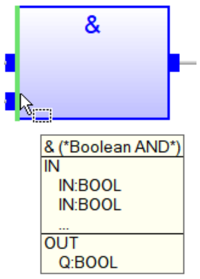


   Click the "& (*Boolean AND*)" function block and drag it onto the editor area.

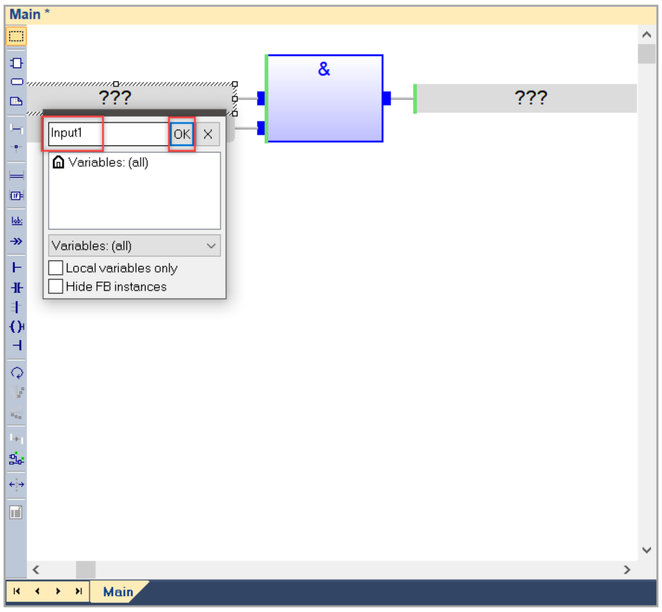


The "AND" function block has got two input and one output variable. The "???" at the inputs and output indicate that no variable has been assigned yet.
The size of the function block can be changed by clicking once on the block and pressing the "+" or "-" key on the keyboard.

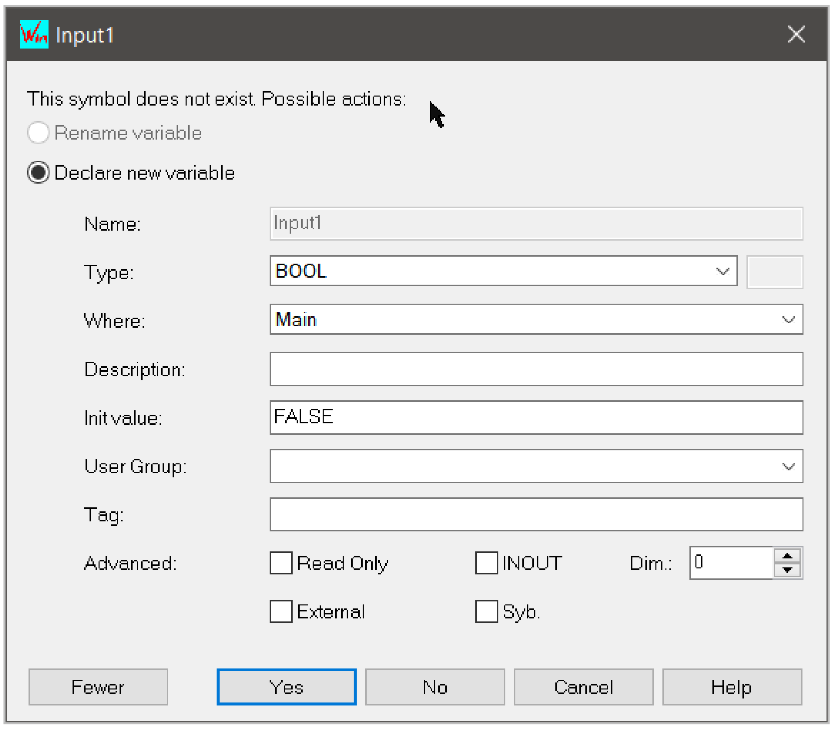
Moving the mouse pointer over the function displays the in- and output data type required:



3. Assigning variables to the function block:
   - Double click on the grey input field with the question marks.
   - Enter the name of the variable
   - Click "OK"



As the variable has not been declared in the project before a windows pops up which allows you to select the data type, initial value, etc...
- The function block input data type is boolean, therefore select BOOL
- The initial value is set to FALSE
- Click "Yes" to add the variable to the project


Input1

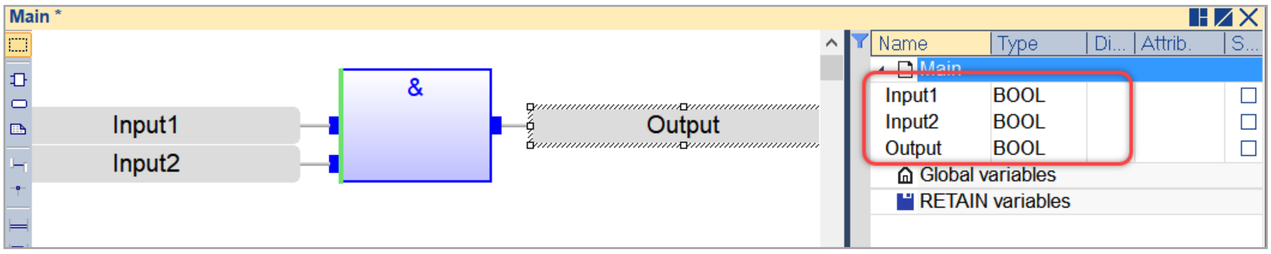
This symbol does not exist. Possible actions:

Rename variable

[✓ Declare new variable]

Name:	Input1
Type:	BOOL
Where:	Main
Description:	
Init value:	FALSE
User Group:	
Tag:	
Advanced:	☐ Read Only ☐ INOUT Dim.: 0
	☐ External ☐ Syb.

Repeat the above procedure to add the variable "Input2" to the second function block input and a "Output" variable to the output. All the newly declared variables are listed in the variables view, on the right of the screen.

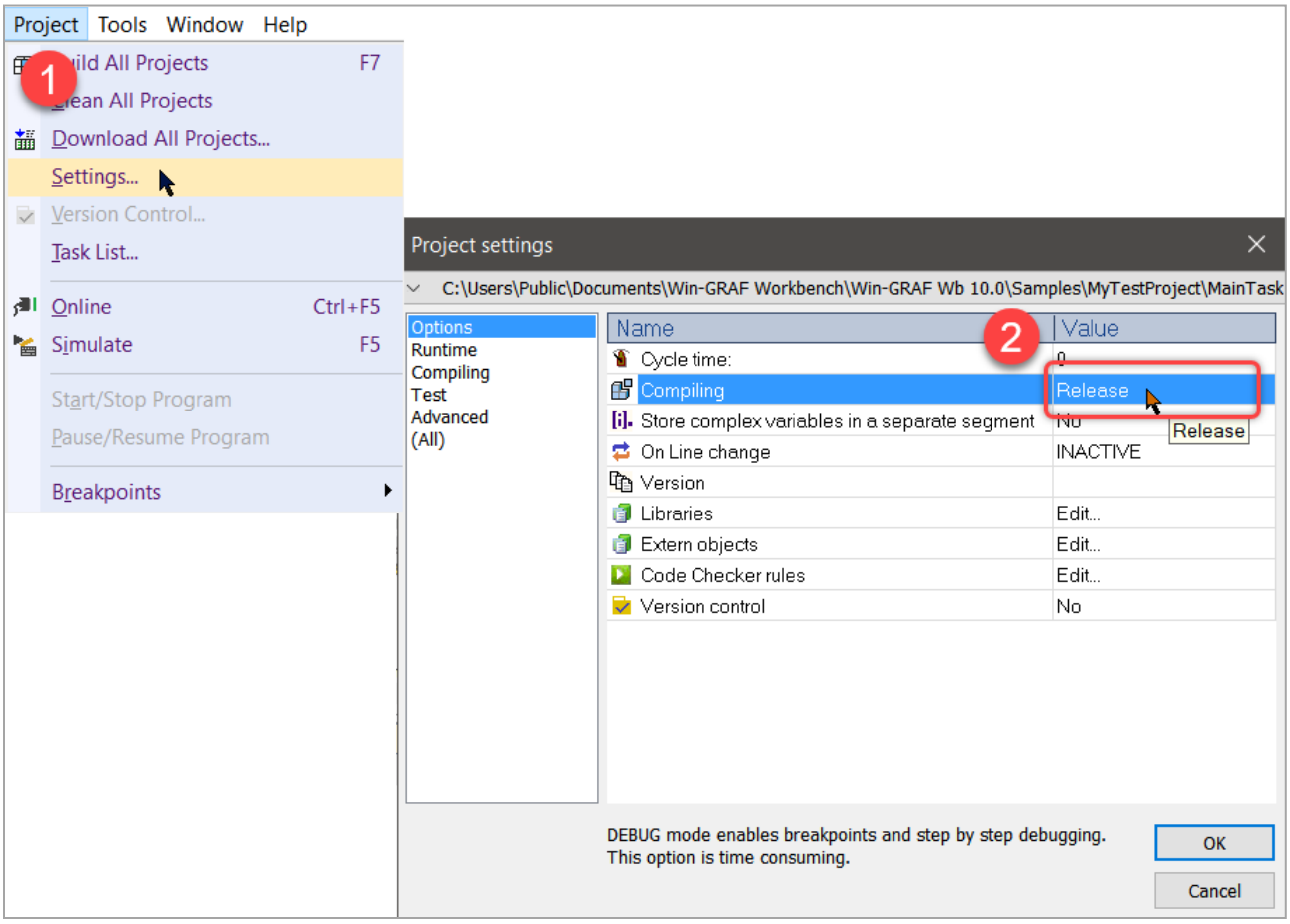


Name	Type	Di...	Attrib.	S...
Main				
Input1	BOOL			☐
Input2	BOOL			☐
Output	BOOL			☐
Global variables				
RETAIN variables				

2.3 Build/ Compile Application

The workbench supports two types of code generation: Release and Debug mode. Application compiled in 'Debug' mode supports cycle by cycle execution, breakpoints and step by step debugging. Breakpoints can be placed anywhere in the source code of the application. The debugger also shows the call stack of the UDFBs and sub-programs when in step by step execution. An application compiled in 'Debug' mode includes additional information for stepping. This leads to bigger code size and less performances.
**It is recommended to compile your applications in 'Release' mode.*

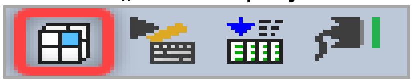
Following steps show how to build an application:
1. Click „Project“ from Workbench menu, and select „Settings“. A „Project settings“ dialog shows up.
2. Find „Options“ in the left side menu, on the right side shows a list of Workbench functions. Select „Compiling“ and double click to switch over the „Release“ or „Debug“ method. Then click „OK“.



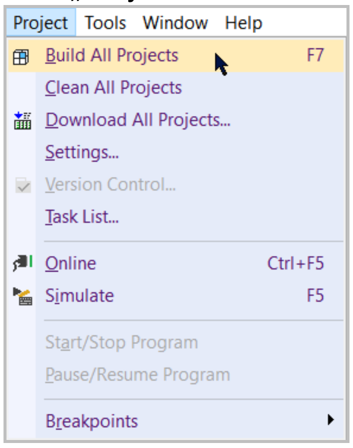
Name	Value
Cycle time:	0
Compiling	Release
Store complex variables in a separate segment	No
On Line change	INACTIVE
Version	
Libraries	Edit...
Extern objects	Edit...
Code Checker rules	Edit...
Version control	No

DEBUG mode enables breakpoints and step by step debugging.
This option is time consuming.

3. To build the project, choose from one of following methods:
   a. Click the „Build all project" button from the Toolbar;

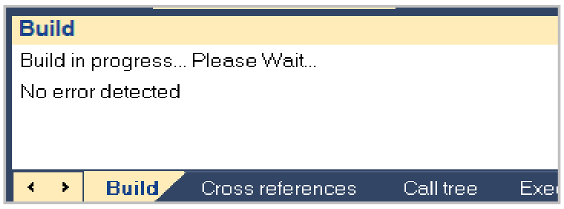


   b. Click „Project" from Workbench menu and select „Build all projects";



   c. Press „F7".

Then you will get a string in the *Output Message* window indicates whether the process is successful or not.



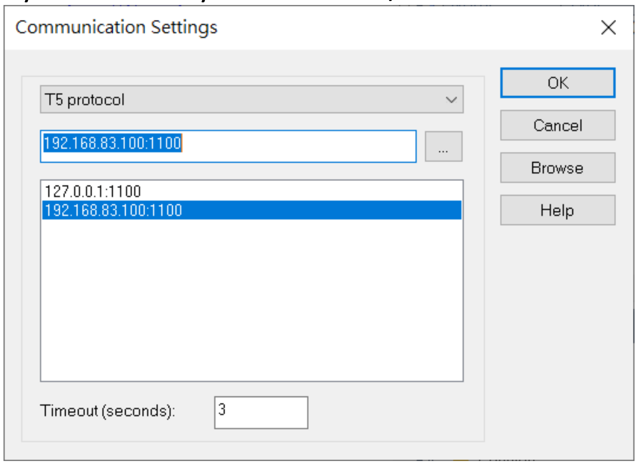
```
Build in progress... Please Wait...
No error detected
```

2.4 Download Application

After the application has been successfully compiled, the application has to be downloaded to the runtime in order to be executed. The Win-GRAF workbench exchanges data with the runtime via TCP/IP communication.
In order to establish a TCP/IP communication the workbench needs to know the IP address and the socket port number of the target runtime. Consult the user manual of the target device to determine how to set and get the communication configuration data.

Following steps show how to download the compiled application:

1. Set the communication parameters between workbench and runtime:
   a. Workbench: Set the IP Address and socket port number of the target runtime.
      i. Select „Tools/Communication Settings...“ from the menu or double click the „offline“ section in the status bar in the bottom of the window.
      ii. Edit IP address and port number of the target runtime. Both parameters has to be separated by a colon. Only Ethernet TCP/IP communication is being supported. Click „OK“.


Communication Settings

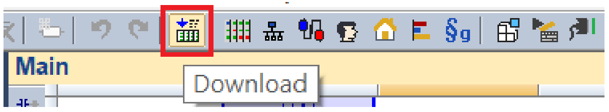
T5 protocol

192.168.83.100:1100

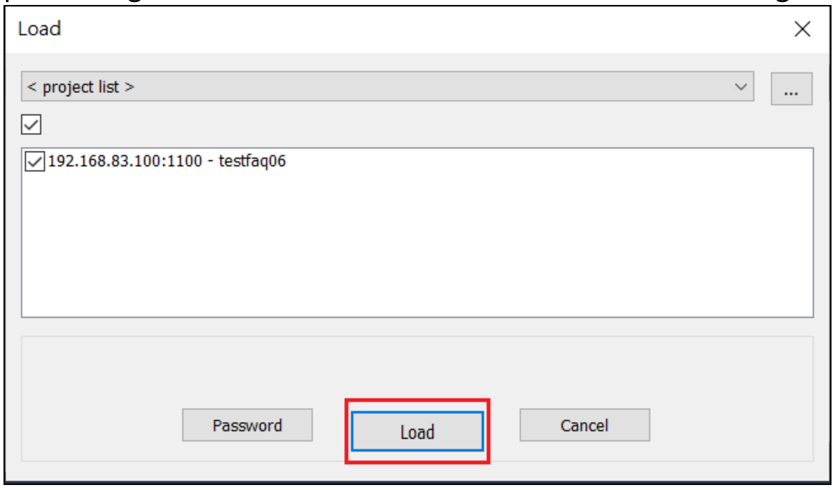
127.0.0.1:1100
192.168.83.100:1100

Timeout (seconds): 3

2. Download program to the PAC
   a. Press the download icon as following captured picture:



   b. Confirm again the settings of the PAC IP and files, press “Load” button to run the download processing to the PAC. While the finish screen is as following:


Load

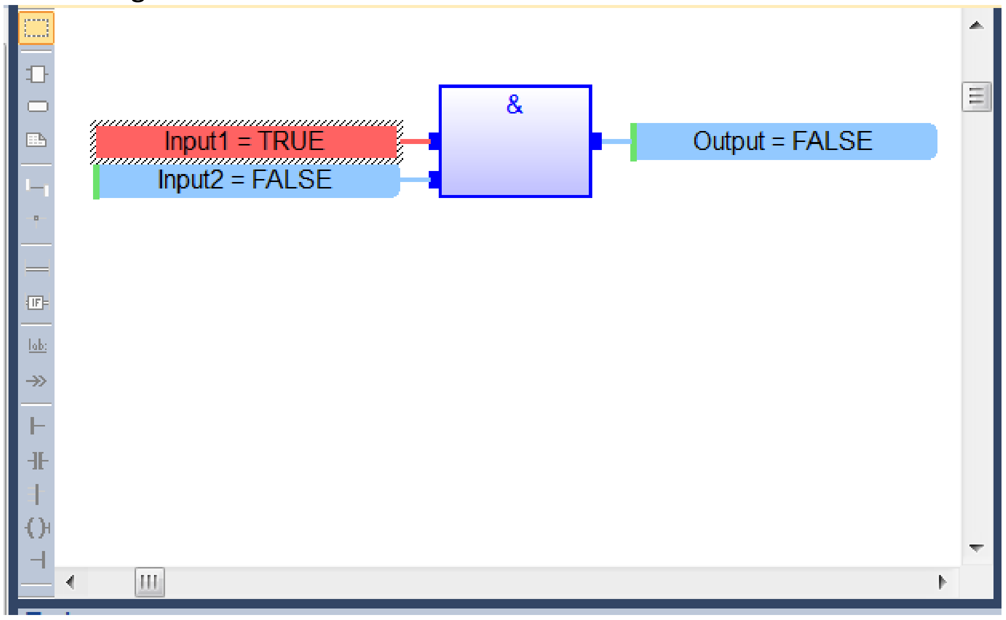
< project list >

192.168.83.100:1100 - testfaq06

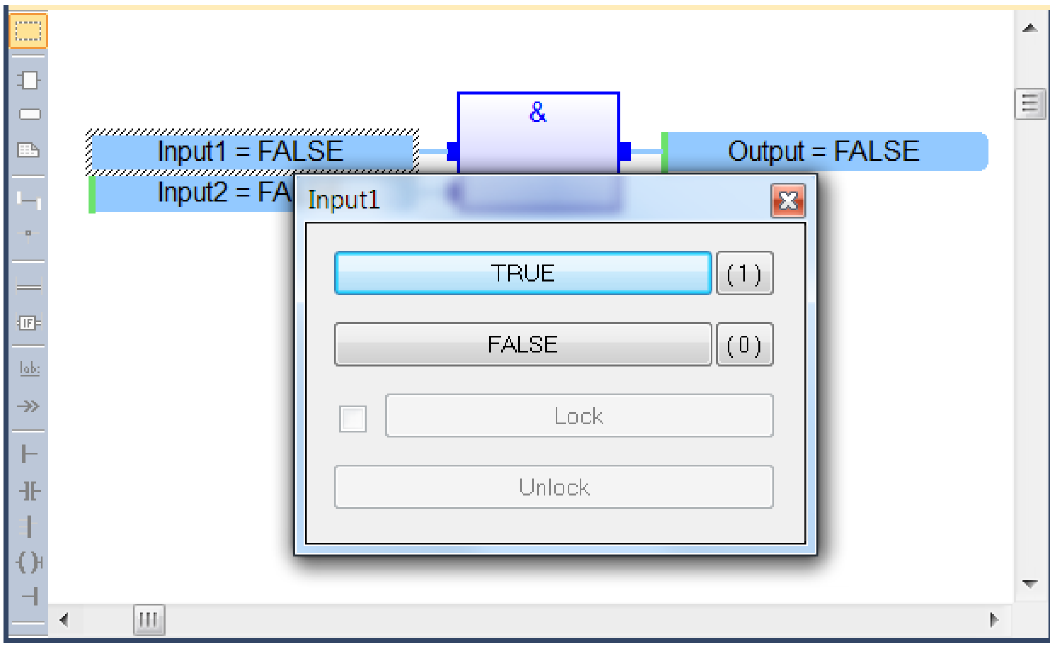
2.5 Debugging

The workbench allows the user to directly change the variable values while the PLC application is running. The workbench has to be connected to the runtime to display the current variable value. This chapter describes how to monitor the PLC program and manipulate the variables via the workbench.

The following procedure describes how to directly modify a PLC variable via the workbench. It is assumed that the PLC application has already been download and is running.
Step 1: Establish a TCP/IP connection between the workbench and runtime:
Click the 'Online' button:
After a connection has been established all the current values of each variables are displayed next to the variable names. These variable values are updates in each task cycle if the runtime is idling.
Step 2: Variable values can be directly changed via the workbench:
Double click the 'Input1' variable next to the function block and click the TRUE button of the popup window.
The input variable changes now from FALSE to TRUE:



All PLC data type can be manipulated in the described way. This allows direct testing of the PLC program.



3 I/O Modules in Workbench

ICP DAS bundle in almost all I/O modules library into this version of Win-GRAF Workbench thus users can now easily double-click and select the main PAC unit and the modules put on the PAC slot. Please note that this library is not updated all the time. If you found modules not list while your selecting, please contact with your Sales Representative.

3.1 Add PAC I/O

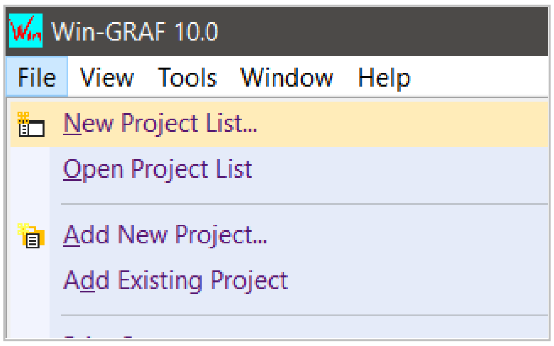
First, take a glance at following sheet for number of PAC slot and corresponding I/O modules:

PAC Model	Slot No. (from left to right)	The supported PAC I/O modules
WP-9x28-CE7	0 to 7	**Support:** I-9K and I-97K series I/O modules.
WP-8x28-CE7	0 to 7	**Support:** I-8K and I-87K series (High Profile) I/O modules. (Eg., I-8017HW and I-87055W) **Not Support:** I-8K and I-87K series (Low Profile) I/O modules. (Eg., I-8017H and I-87055)
XP-8x38-CE6 (*)	0 to 7	
VP-x2x8-CE7 (*)	0 to 2	
WP-5238-CE7	-	**The Palm-size PAC** which can put-in one XV board. (Eg., XV107, XV116, XV308, etc.)

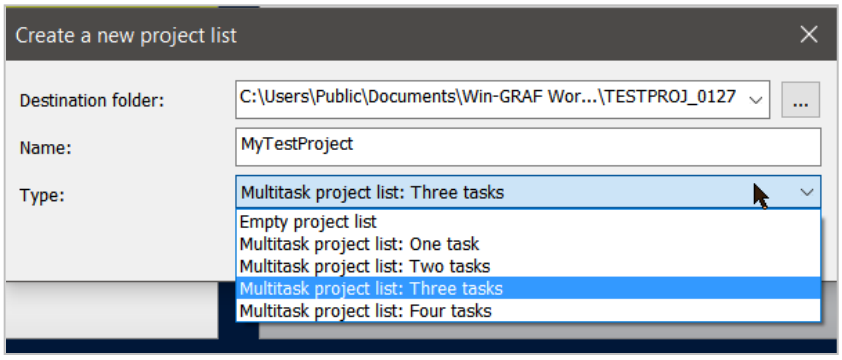
(*) The XP-8038-CE6, VP-x208, WP-5xx8-CE7 are the PAC without slot.

Following we will introduce how to add PAC along with the relative PAC I/O modules.

1. Click the Toolbar, select and Create a "New Project List".



2. Key in the Name and choose the type of your project list. The destination folder will be selected automatically. We suggest to use the default route. Click OK.

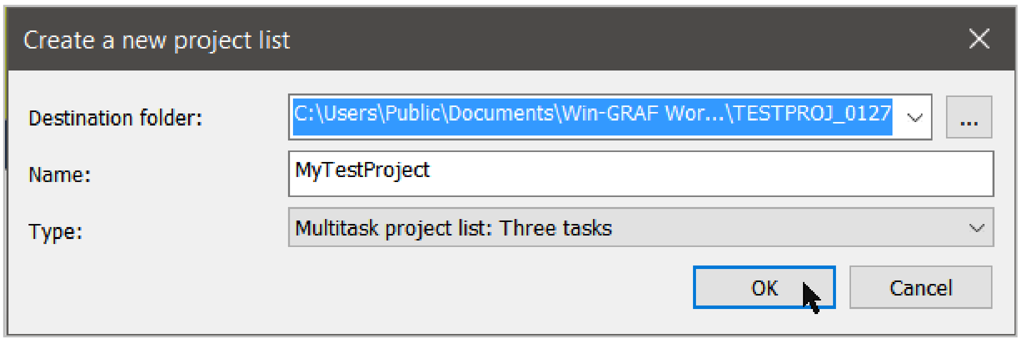

Create a new project list

Destination folder: C:\Users\Public\Documents\Win-GRAF Wor...\TESTPROJ_0127

Name: MyTestProject

Type: Multitask project list: Three tasks

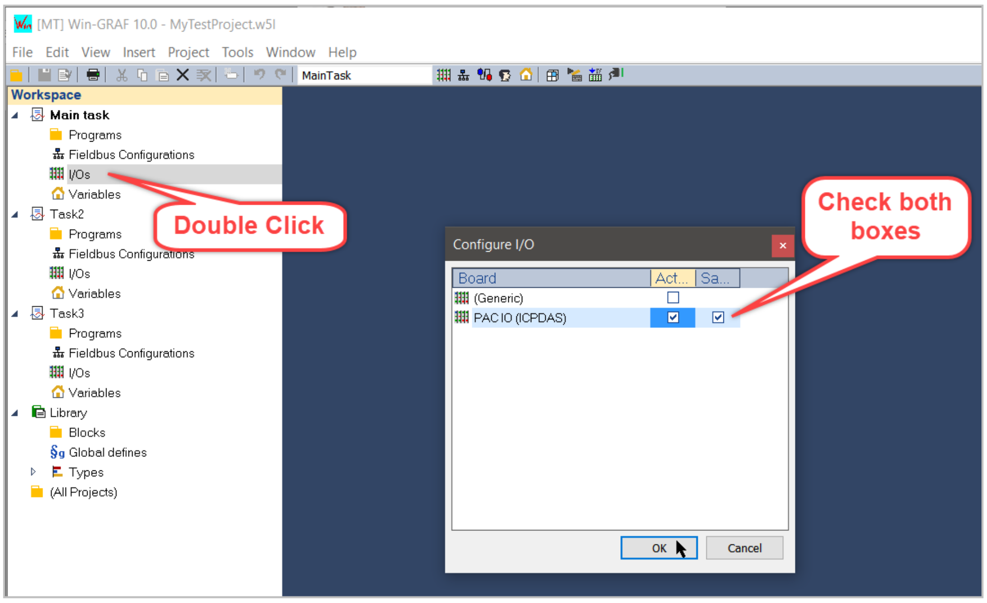
Empty project list
Multitask project list: One task
Multitask project list: Two tasks
Multitask project list: Three tasks
Multitask project list: Four tasks



Create a new project list

Destination folder:	C:\Users\Public\Documents\Win-GRAF Wor...\TESTPROJ_0127
Name:	MyTestProject
Type:	Multitask project list: Three tasks

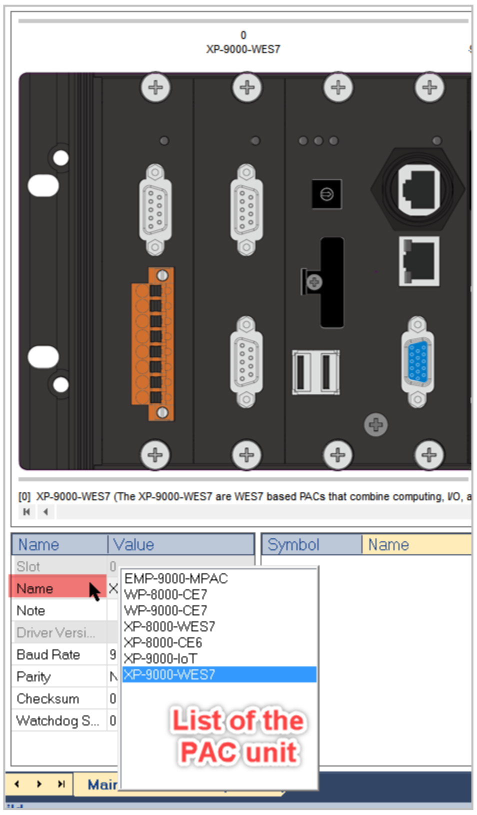
3. Double-click I/Os and there pop-up a small window. Cancel the Generic checkbox and check both checkboxes of PAC IO (ICPDAS) and click OK.



[MT] Win-GRAF 10.0 - MyTestProject.w5l

Board	Act...	Sa...
(Generic)	☐	
PAC IO (ICPDAS)	☑	☑

4. After click OK will have a figure below. (Default PAC is XP-9000-WES7, which is not yet released). To change to the PAC on hand, double-click "name" column beneath the product picture and there shows a list of PAC unit.

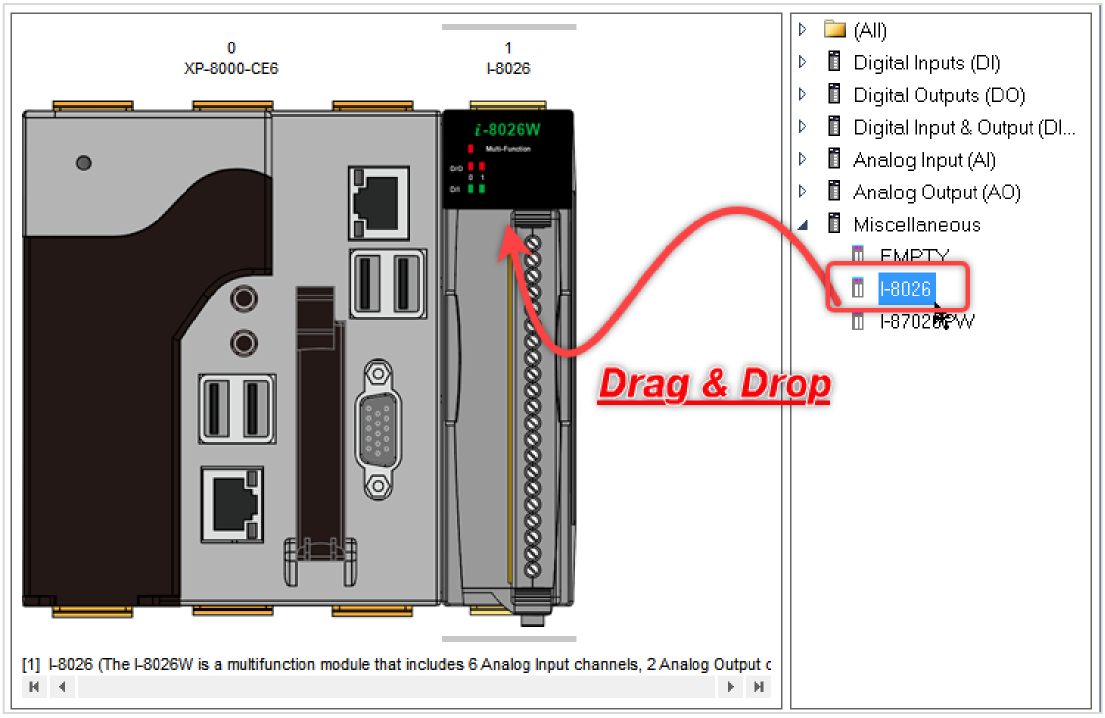


[0] XP-9000-WES7 (The XP-9000-WES7 are WES7 based PACs that combine computing, I/O, a

Name	Value
Slot	0
Name	X
Note	
Driver Versi...	
Baud Rate	9
Parity	N
Checksum	0
Watchdog S...	0

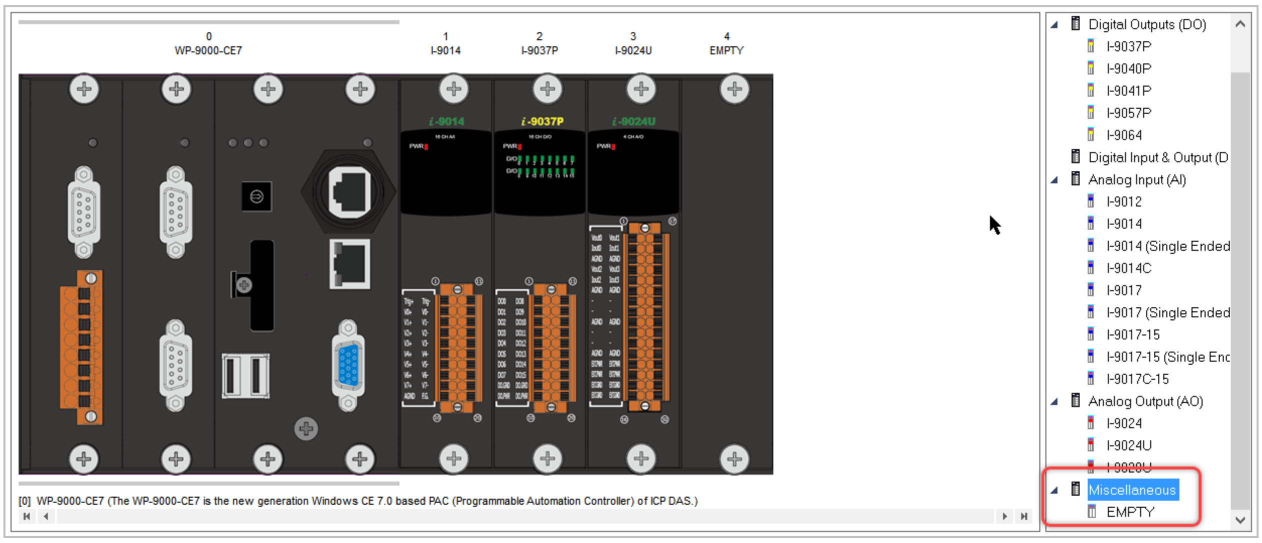
EMP-9000-MPAC
WP-8000-CE7
WP-9000-CE7
XP-8000-WES7
XP-8000-CE6
XP-9000-IoT
XP-9000-WES7

5. Add PAC I/O modules to the PAC extension slots:
Open the module list on the upper-right of the window, select module number and drag left over the space area of main PAC. In following figure, a multifunction module I-8026 is added to the first slot.



[1] I-8026 (The I-8026W is a multifunction module that includes 6 Analog Input channels, 2 Analog Output c

6. Following figure shows a 4 slots WP-9428-CE7 PAC with 3 I/O modules putting in. ICP DAS provide different PAC with different numbers of I/O slots, eg., 1/4/8 or 1/3/7. Please double check the website to confirm the maximum I/O slot numbers of the PAC you have on hand. If in the Win-GRAF Workbench you need less I/O modules than the exact number of slot, please drag an "EMPTY" module from Miscellaneous list.

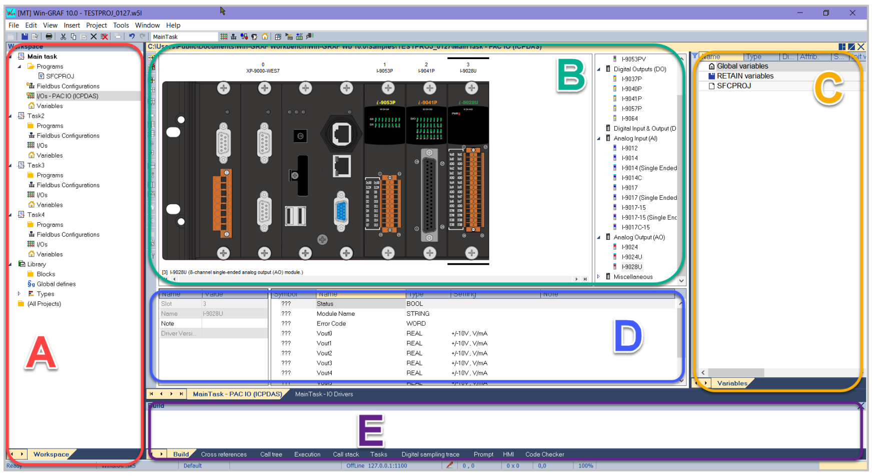


[0] WP-9000-CE7 (The WP-9000-CE7 is the new generation Windows CE 7.0 based PAC (Programmable Automation Controller) of ICP DAS.)

3.2 Add Remote I/O

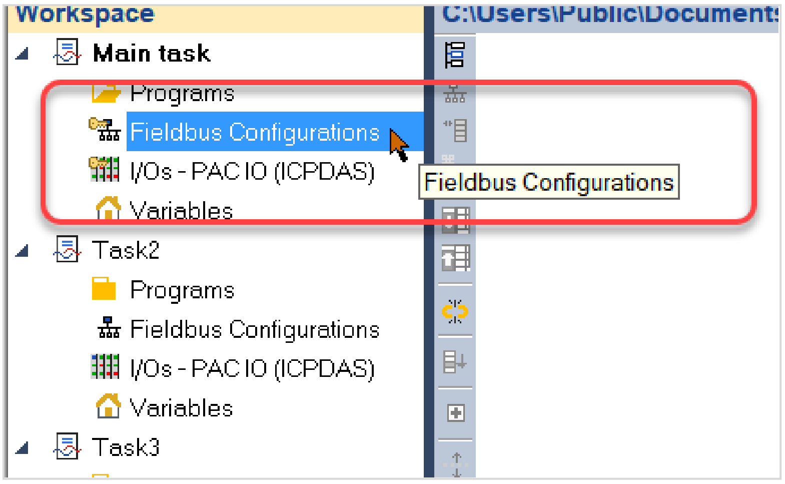
3.2.1 Fieldbus Configuration

After adding the main PAC along with the I/O modules, we can get a small system as in following figure. Rectangle B+D is similiar to previous Chapter 1 we introduced the User Interface. We won't go into details here.

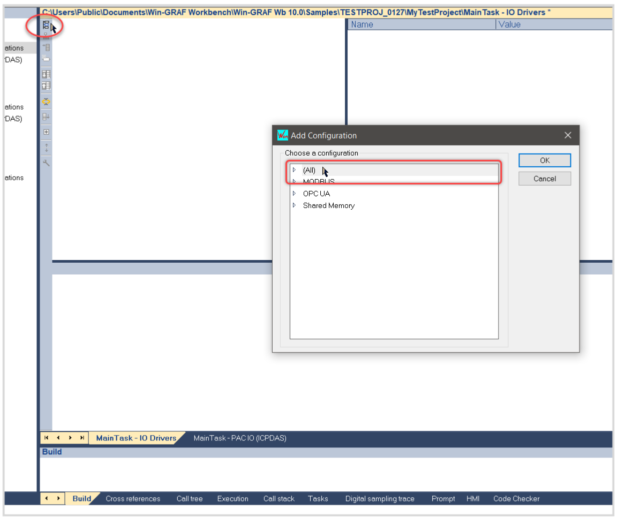


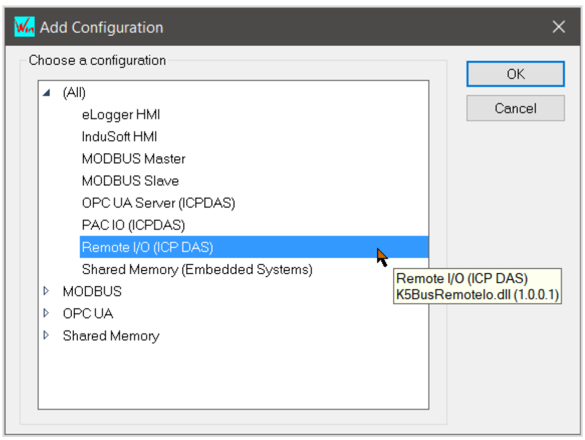
Next, we have to define variables and bind to the individual channels to make the I/O work for us.

1. Double click "Fieldbus Configuration" and get a window like following figure.

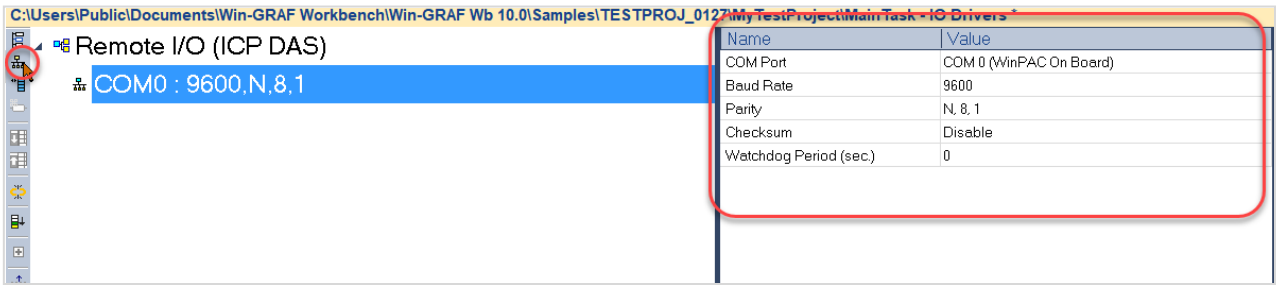


2. Click left-top to "Insert Configuration" and here shows a pop-up window. Double click "All" to choose "Remote I/O (ICPDAS)", click OK.


C:\Users\Public\Documents\Win-GRAF Workbench\Win-GRAF Wb 10.0\Samples\TESTPROJ_0127\MyTestProject\MainTask - IO Drivers *
Name
Value
Add Configuration
Choose a configuration
(All)
MODBUS
OPC UA
Shared Memory
OK
Cancel
MainTask - IO Drivers
MainTask - PAC IO (ICPDAS)
Build



3. Then click second-left-top "Insert Master/Port" to add a COM0. On the right side shows the information of the COM0.



C:\Users\Public\Documents\Win-GRAF Workbench\Win-GRAF Wb 10.0\Samples\TESTPROJ_0127\MyTestProject\MainTask - IO Drivers *

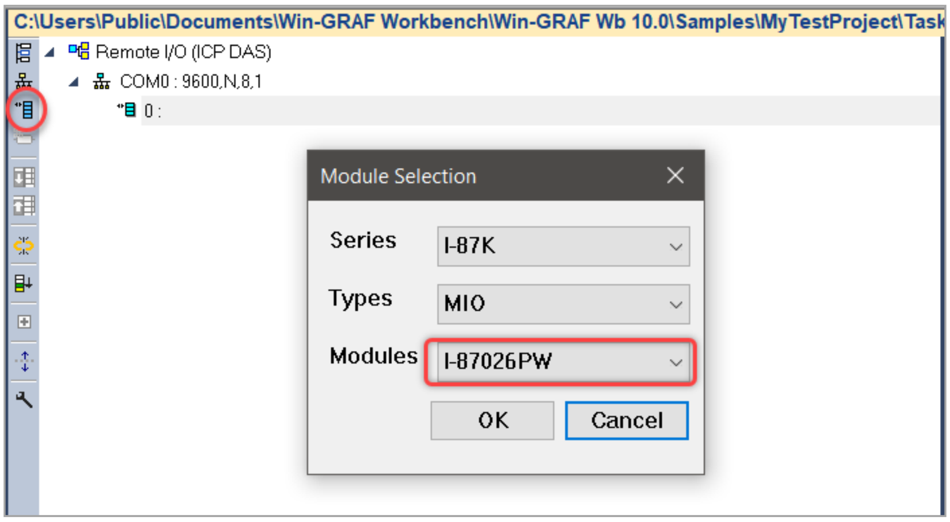
Remote I/O (ICP DAS)
COM0 : 9600,N,8,1

Name	Value
COM Port	COM 0 (WinPAC On Board)
Baud Rate	9600
Parity	N, 8, 1
Checksum	Disable
Watchdog Period (sec.)	0

Note: Base on different PAC items (eg. XPAC/WinPAC...), they use different COM port to do slot communication. For detail information, please refer to the hardware manual if you need help.

Note: Baudrate, Parity and Checksum should be matched with the hardware settings. The hardware setting can be acquired by DCON Utility on PAC.

4. Click third-left-top icon "Insert Slave/Data Block" to add module of your main PAC. Here we have XP-8000 for the main unit and I/O module we choose I-87026PW. Click OK.



C:\Users\Public\Documents\Win-GRAF Workbench\Win-GRAF Wb 10.0\Samples\MyTestProject\Task

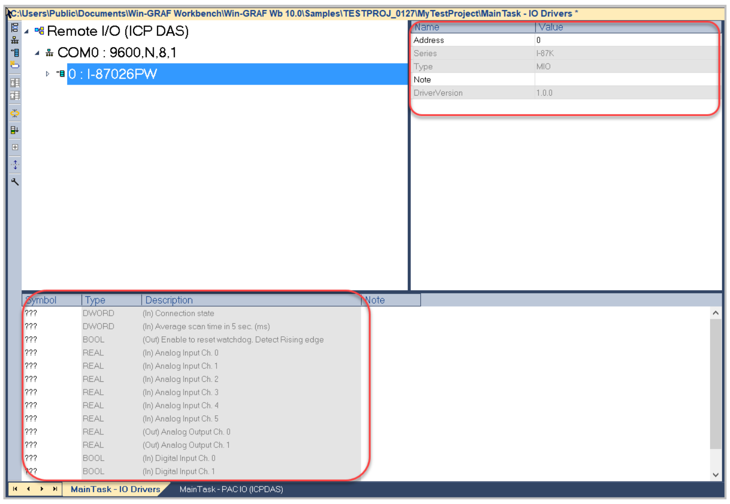
Remote I/O (ICP DAS)
COM0 : 9600,N,8,1
0 :

Module Selection

Series: I-87K
Types: MIO
Modules: I-87026PW

OK Cancel

5. After add I-87026PW module, at top-right window shows the slot address, I/O series and I/O type of this Remote I/O. The window beneath shows type of the data type and the detail description.



C:\Users\Public\Documents\Win-GRAF Workbench\Win-GRAF Wb 10.0\Samples\TESTPROJ_0127\MyTestProject\MainTask - IO Drivers *

Remote I/O (ICP DAS)
COM0 : 9600,N,8,1
0 : I-87026PW

Name	Value
Address	0
Series	I-87K
Type	MIO
Note	
DriverVersion	1.0.0

Symbol	Type	Description	Note
???	DWORD	(In) Connection state	
???	DWORD	(In) Average scan time in 5 sec. (ms)	
???	BOOL	(Out) Enable to reset watchdog. Detect Rising edge	
???	REAL	(In) Analog Input Ch. 0	
???	REAL	(In) Analog Input Ch. 1	
???	REAL	(In) Analog Input Ch. 2	
???	REAL	(In) Analog Input Ch. 3	
???	REAL	(In) Analog Input Ch. 4	
???	REAL	(In) Analog Input Ch. 5	
???	REAL	(Out) Analog Output Ch. 0	
???	REAL	(Out) Analog Output Ch. 1	
???	BOOL	(In) Digital Input Ch. 0	
???	BOOL	(In) Digital Input Ch. 1	

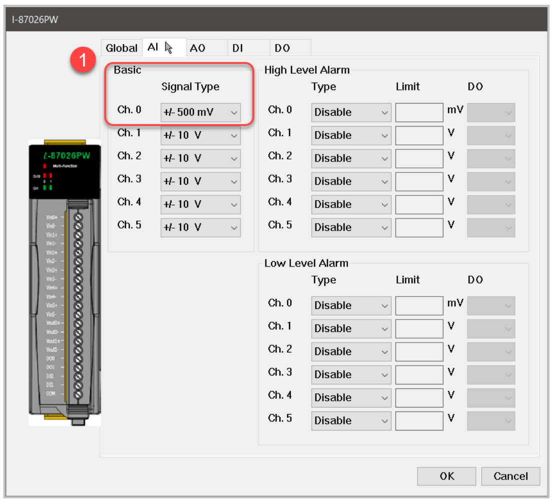
MainTask - IO Drivers | MainTask - PAC IO (ICPDAS)

Note: Address should be matched with the hardware settings. The hardware setting can be acquired by DCON Utility on PAC.

3.2.2 Variable Binding

In most situation, variables should be "binding" to carry settings. Take I-87026PW for example:

Step 1. Double click the I-87026PW picture to show this dialog. Select "AI" Tab to set the Ch.0 Signal Type to +/-500mV. Click OK.



Basic

	Signal Type
Ch. 0	+/- 500 mV
Ch. 1	+/- 10 V
Ch. 2	+/- 10 V
Ch. 3	+/- 10 V
Ch. 4	+/- 10 V
Ch. 5	+/- 10 V

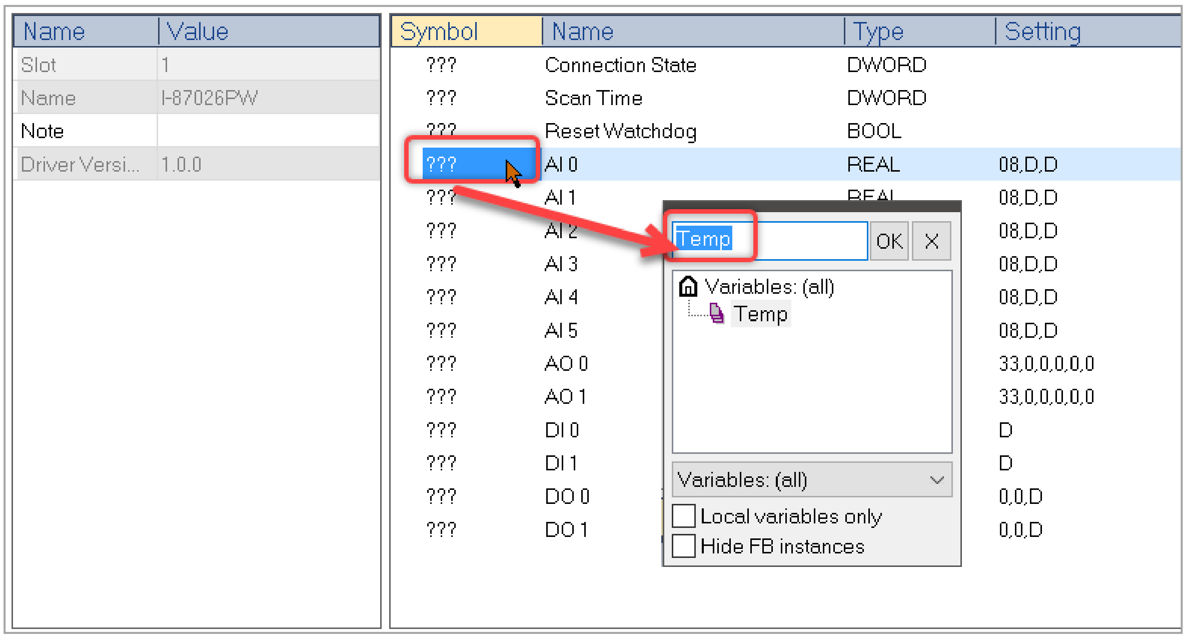
High Level Alarm

	Type	Limit		DO
Ch. 0	Disable		mV	
Ch. 1	Disable		V	
Ch. 2	Disable		V	
Ch. 3	Disable		V	
Ch. 4	Disable		V	
Ch. 5	Disable		V	

Low Level Alarm

	Type	Limit		DO
Ch. 0	Disable		mV	
Ch. 1	Disable		V	
Ch. 2	Disable		V	
Ch. 3	Disable		V	
Ch. 4	Disable		V	
Ch. 5	Disable		V	

Step 2. Double click "???" column of AI0 in the Variable Area beneath the PAC picture. Key in variable name in the space (here we key in Temp) and click OK.



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

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

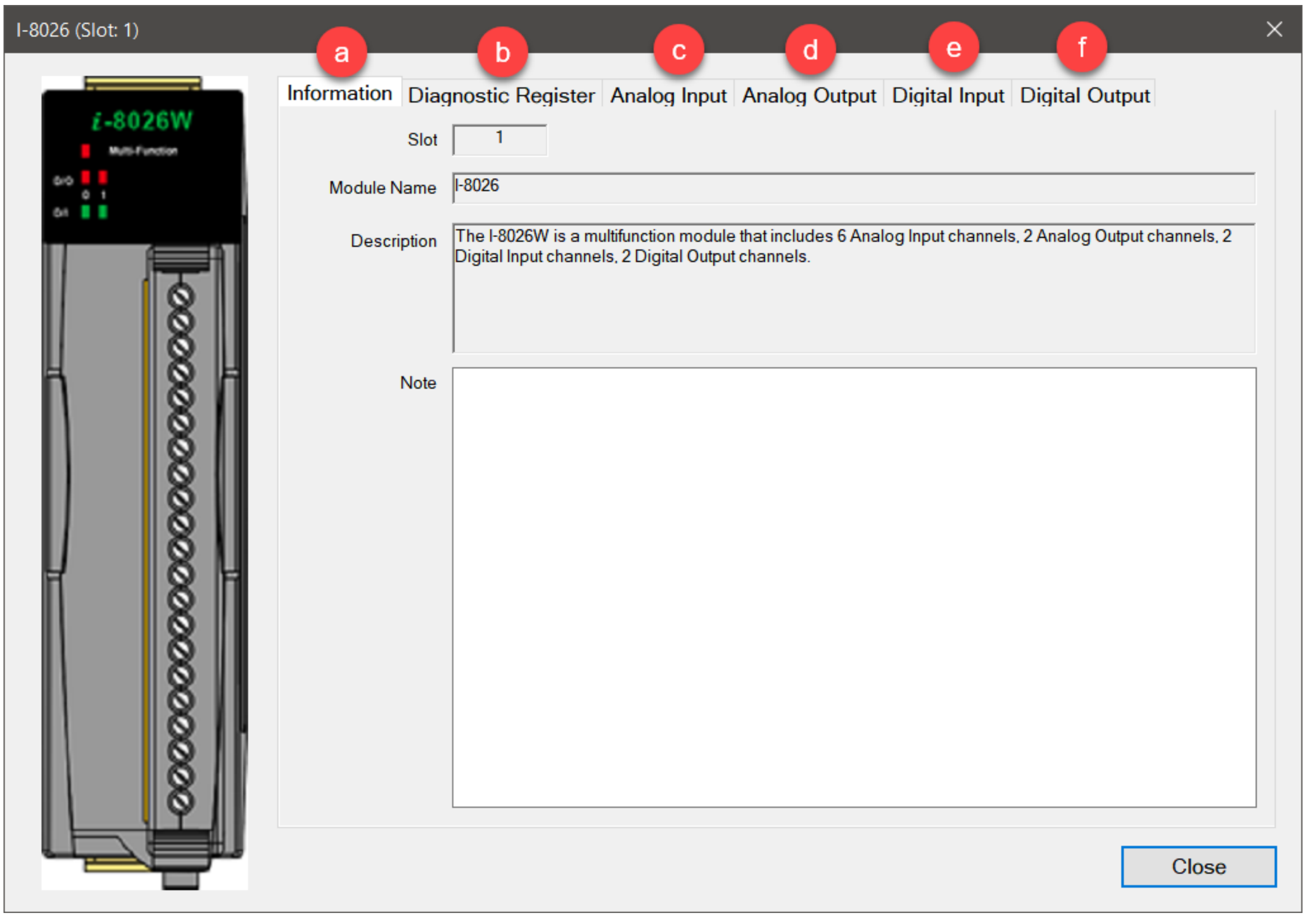
Thus, the signal type (or configurations of channels) will be carried by the binded variable to runtime and the binding setting is done correctly.

3.3 I/O Module Settings

In previous section we have added PAC and I/O modules into our Win-GRAF Workbench. We also finish setting up the binding variable to allow data read/write. In following part, we will start to make some basic settings of these PAC I/O products. We also provide some hints users should know about to complete the settings.

3.3.1 I-8K Series

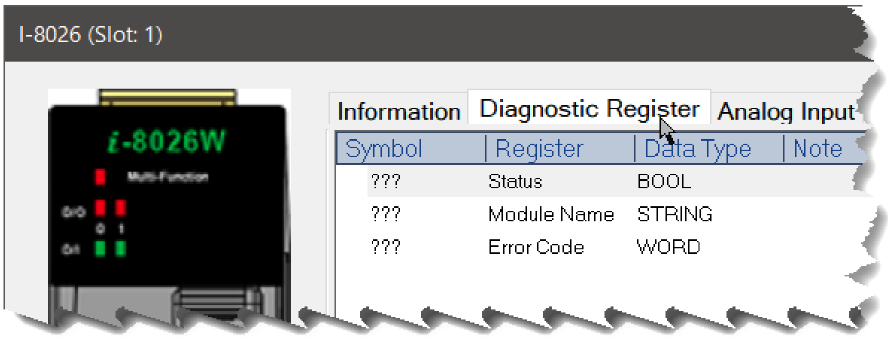
Take I-8K series (I-8026W with 6AI, 2AO, 2DI, and 2DO) for example:
As long as you double click the module you'd like to do setting up, below figure shows the product picture and few bookmarks about this module.



Slot: 1
Module Name: I-8026
Description: The I-8026W is a multifunction module that includes 6 Analog Input channels, 2 Analog Output channels, 2 Digital Input channels, 2 Digital Output channels.
Note:

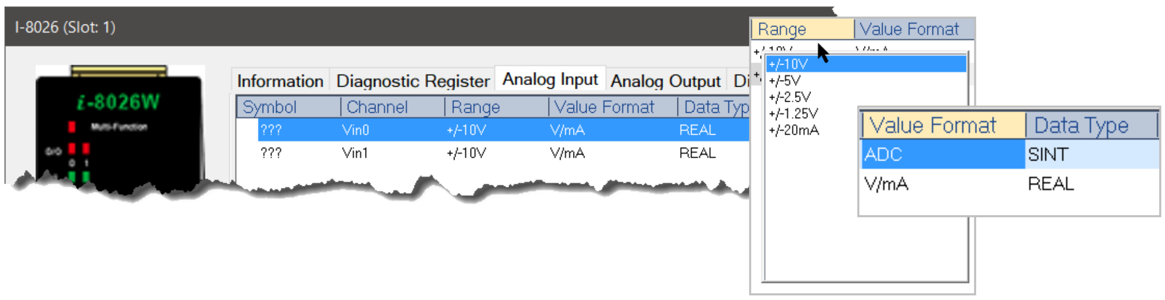
a. Information: shows basic module detail and product description.

b. Diagnostic Register: Shows data type of this module and Allow PLC program to directly access the module name, communication status and the error type for failed module access.



Symbol	Register	Data Type	Note
???	Status	BOOL	
???	Module Name	STRING	
???	Error Code	WORD	

c. Analog Input: Find "Range" and double click below column to choose the Voltage input, and then double click column of "Value Format" to change into ADC and "Data Type" will accordingly change to SINT.



I-8026 (Slot: 1)

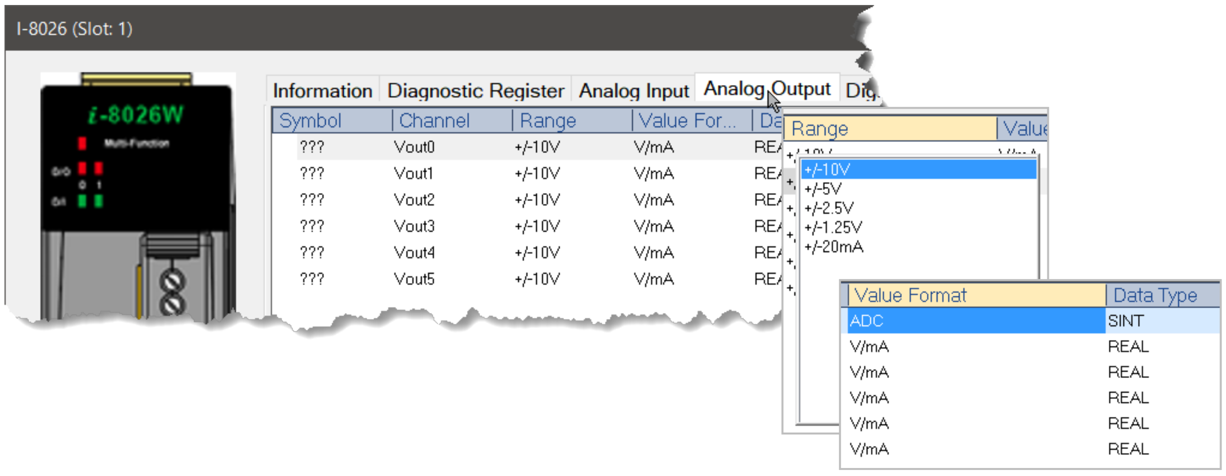
Information | Diagnostic Register | Analog Input | Analog Output | Di

Symbol	Channel	Range	Value Format	Data Typ
???	Vin0	+/-10V	V/mA	REAL
???	Vin1	+/-10V	V/mA	REAL

Range
+/-10V
+/-5V
+/-2.5V
+/-1.25V
+/-20mA

Value Format	Data Type
ADC	SINT
V/mA	REAL

d. Analog Output: As step c., the same to change the "Range" and the "Value Format".



I-8026 (Slot: 1)

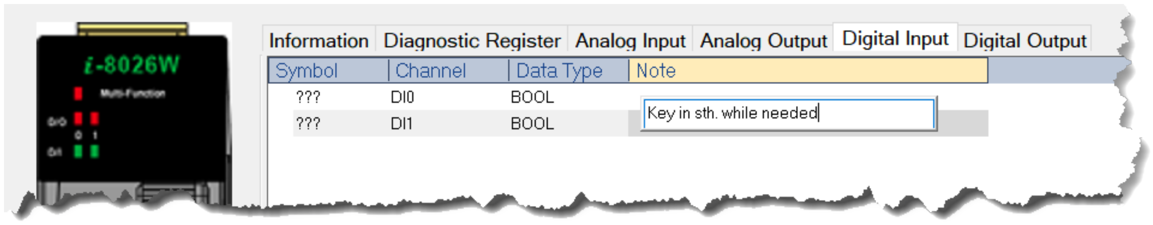
Information | Diagnostic Register | Analog Input | Analog Output | Dig

Symbol	Channel	Range	Value For...	Da
???	Vout0	+/-10V	V/mA	REA
???	Vout1	+/-10V	V/mA	REA
???	Vout2	+/-10V	V/mA	REA
???	Vout3	+/-10V	V/mA	REA
???	Vout4	+/-10V	V/mA	REA
???	Vout5	+/-10V	V/mA	REA

Range
+/-10V
+/-5V
+/-2.5V
+/-1.25V
+/-20mA

Value Format	Data Type
ADC	SINT
V/mA	REAL
V/mA	REAL
V/mA	REAL
V/mA	REAL
V/mA	REAL

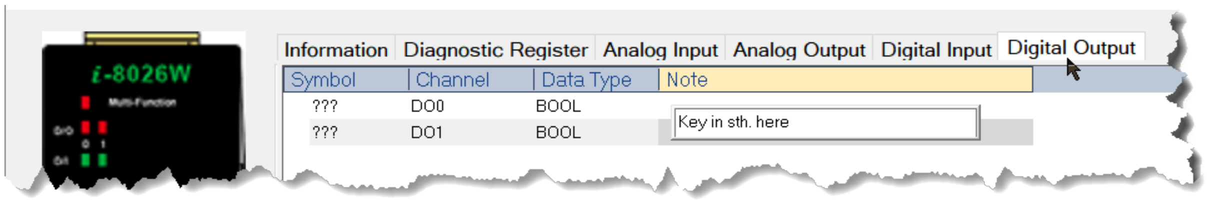
e. Digital Input: This figure shows Digital Input of this module. Double click column of "Note" to key in something if needed.



Information | Diagnostic Register | Analog Input | Analog Output | Digital Input | Digital Output

Symbol	Channel	Data Type	Note
???	DI0	BOOL	
???	DI1	BOOL	Key in sth. while needed

f. Digital Output: As step e., double click column to key in something in the below column of "Note" if needed.

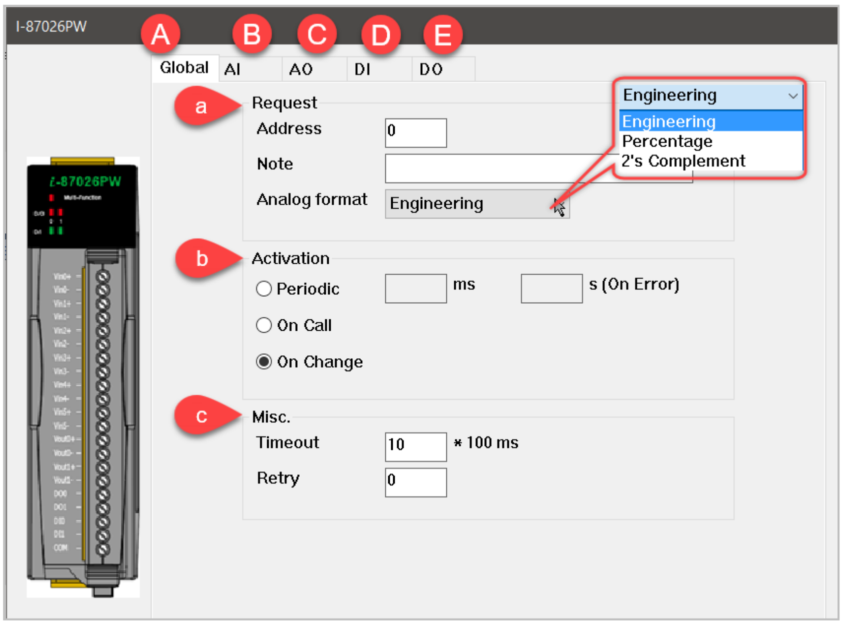


Information | Diagnostic Register | Analog Input | Analog Output | Digital Input | Digital Output

Symbol	Channel	Data Type	Note
???	DO0	BOOL	
???	DO1	BOOL	Key in sth. here

3.3.2 I-87K Series

Take I-87K series (I-87026PW with 6AI, 2AO, 2DI and 2DO) for example:
Compared with I-8K modules, we provide different environment in setting up I-87K modules. Please check the following figure.



Request
Address: 0
Note:
Analog format: Engineering

Activation
○ Periodic ms s (On Error)
○ On Call
● On Change

Misc.
Timeout: 10 * 100 ms
Retry: 0

A. Global: In Global tab shows the picture and information about the module.

a. Request

a-1. Address: A unique number for a module on the serial port.

a-2. Note: User's note.

a-3. Analog Format (AIO modules only): There are 3 different types of Analog format. Please choose one that suits your settings from the drop-down menu.

b. Activation

b-1. Periodic:

When module connected, PAC will send IO commands periodically with settings.
When module disconnected, PAC will send initial commands to connect module periodically with ‚On Error' setting.

b-2. On Call:

When module connected and enabled variable (auto generated) is triggered, PAC will send IO commands.
When module disconnected and enabled variable is triggered, PAC will send initial commands to connect module.

b-3. On Change:

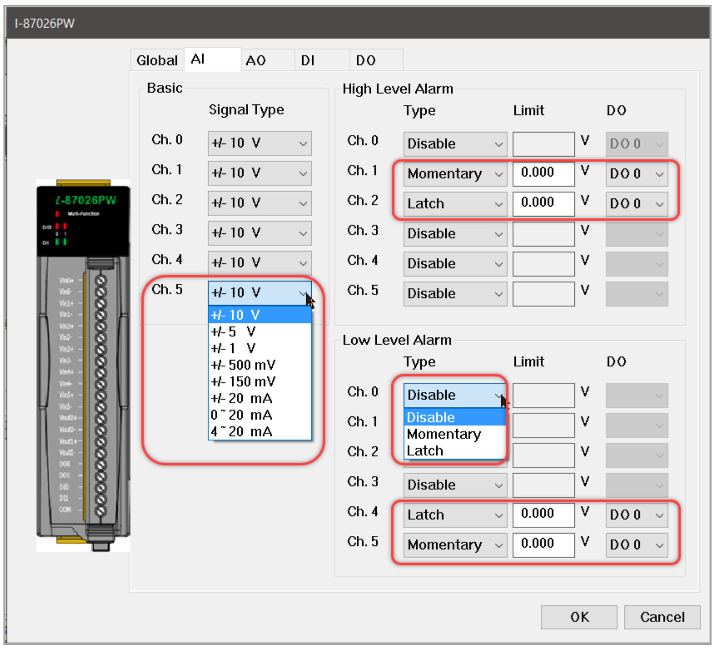
When module connected, PAC will send DO or AO commands when the output values changing and send DI or AI commands continuously.
When module disconnected, PAC will send initial commands by every 1 second.

c. Misc.

c-1. Timeout:
When the module responds with timeout, the PAC determines that the module is disconnected.

c-2. Retry:
When the module responds timeout, the PAC will resend the last command according to the set number of times. After the PAC resend reaches the set number of times, it will start to send the initial command.

B. AI: AI tab shows "Signal Type" of the analog input signal. Left-click the drop-down list and there are signal ranges to choose from.
Furthermore, AI channels of I-87026PW module can make the "High Level Alarm" or "Low Level Alarm". Left-click the drop-down list, choose type "Momentary" or "Latch", key-in limit data of signal and set the output channel in the last part of DO.



Basic

	Signal Type
Ch. 0	+/- 10 V
Ch. 1	+/- 10 V
Ch. 2	+/- 10 V
Ch. 3	+/- 10 V
Ch. 4	+/- 10 V
Ch. 5	+/- 10 V

+/- 10 V
+/- 5 V
+/- 1 V
+/- 500 mV
+/- 150 mV
+/- 20 mA
0 ~ 20 mA
4 ~ 20 mA

High Level Alarm

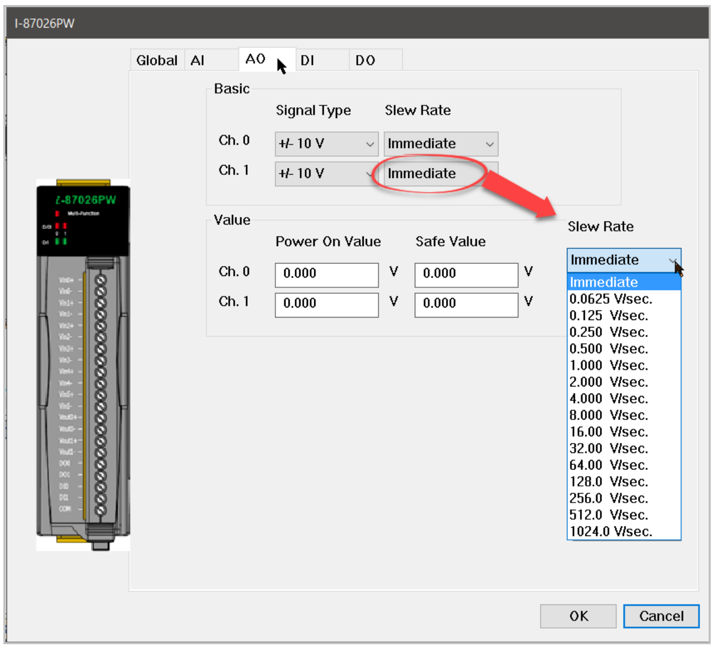
	Type	Limit		DO
Ch. 0	Disable		V	DO 0
Ch. 1	Momentary	0.000	V	DO 0
Ch. 2	Latch	0.000	V	DO 0
Ch. 3	Disable		V	
Ch. 4	Disable		V	
Ch. 5	Disable		V	

Low Level Alarm

	Type	Limit		DO
Ch. 0	Disable		V	
Ch. 1			V	
Ch. 2			V	
Ch. 3	Disable		V	
Ch. 4	Latch	0.000	V	DO 0
Ch. 5	Momentary	0.000	V	DO 0

Disable
Momentary
Latch

C. AO: Let's get to AO tab. Here shows 2-channel AOs of I-87026PW and the signal type is the same as previous part. Most important part of AO is to set up the "Slew Rate" which means how fast the output channel can respond to an abrupt change of input level. Maximum rate range from 0.0625V/sec. to 1024V/sec.



Basic

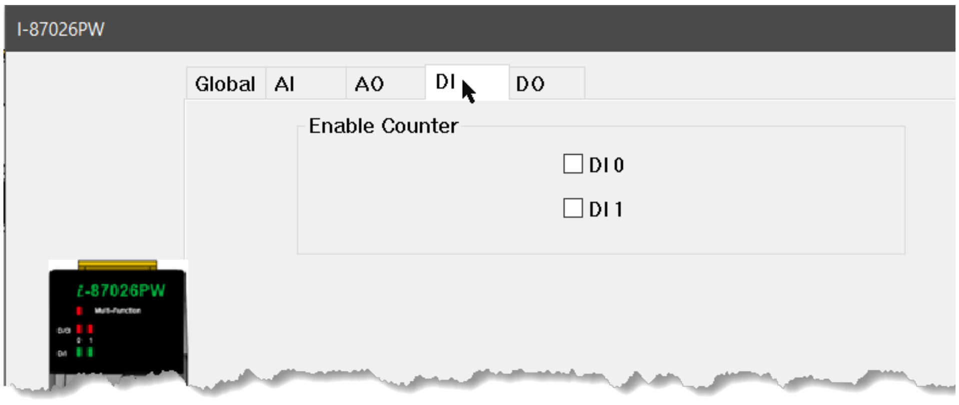
	Signal Type	Slew Rate
Ch. 0	+/- 10 V	Immediate
Ch. 1	+/- 10 V	Immediate

Value

	Power On Value		Safe Value	
Ch. 0	0.000	V	0.000	V
Ch. 1	0.000	V	0.000	V

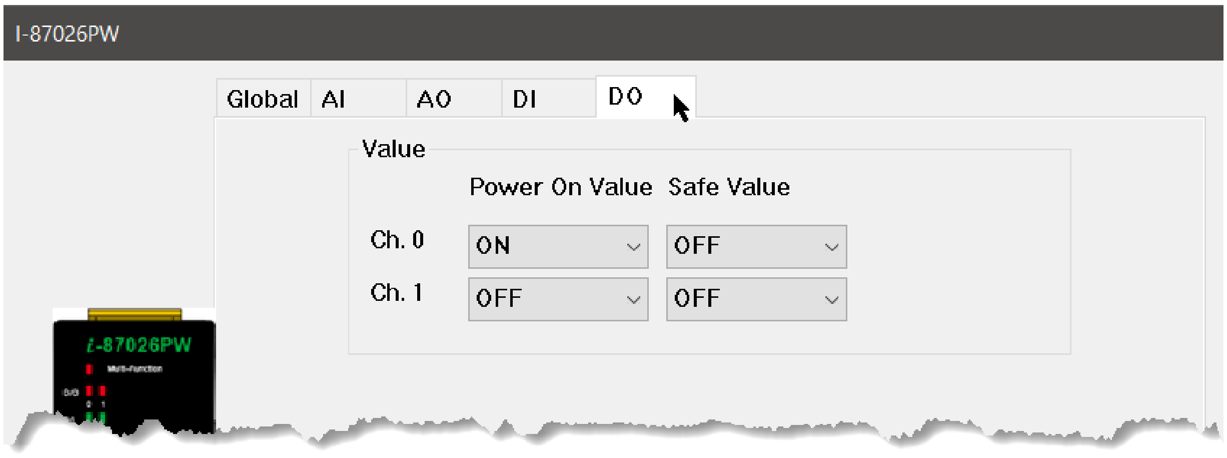
Slew Rate
Immediate
Immediate
0.0625 V/sec.
0.125 V/sec.
0.250 V/sec.
0.500 V/sec.
1.000 V/sec.
2.000 V/sec.
4.000 V/sec.
8.000 V/sec.
16.00 V/sec.
32.00 V/sec.
64.00 V/sec.
128.0 V/sec.
256.0 V/sec.
512.0 V/sec.
1024.0 V/sec.

D. DI: Check the box to enable DI counters. The DI channel of I-87K module supports digital input and low speed (100Hz) counter input. The two function can work simultaneously. Enabling the counter, the update rate of the module becomes slower.



Enable Counter
☐ DI 0
☐ DI 1

E. DO: Set up the Power on Value and the Safe Value here if needed. Drop-down menu shows "ON" or "OFF" to enable this function.



I-87026PW

Value

	Power On Value	Safe Value
Ch. 0	ON	OFF
Ch. 1	OFF	OFF

4 Retain Variables

What is Retain Variable? A retain variable is a variable which:

- is non-volatile and is stored in a normal disk file.

- is known by all programs (when its content is changed, the change is propagated to all equations in which this variable is used)

- normally does not contain real-time critical data.

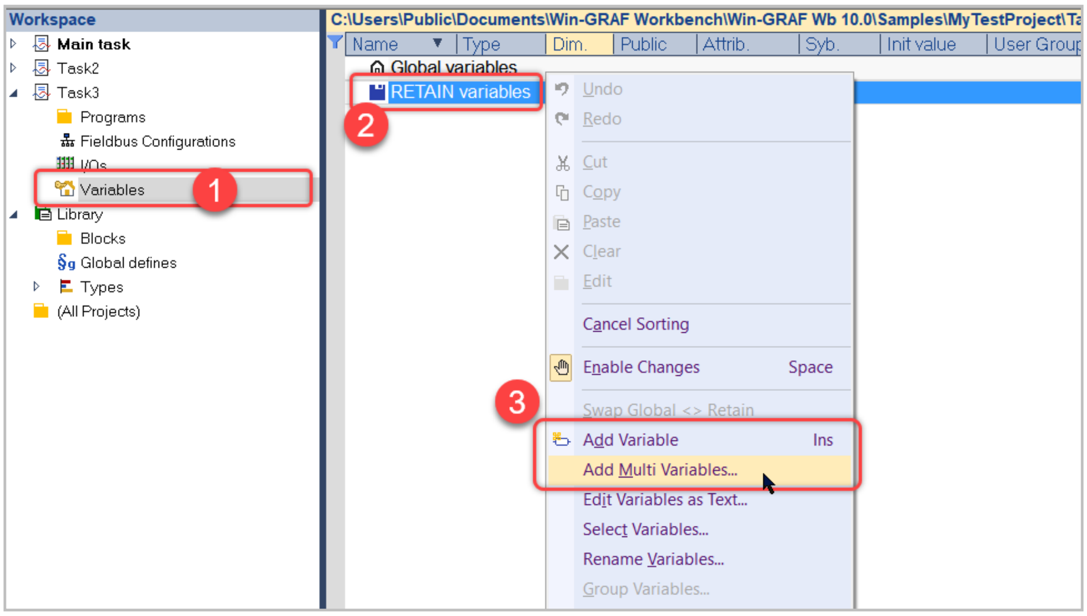
In other words, retain variables are used to protect important data. The system can retain the latest variable data while an unexpected shut down occurred.

4.1 Retain Variables

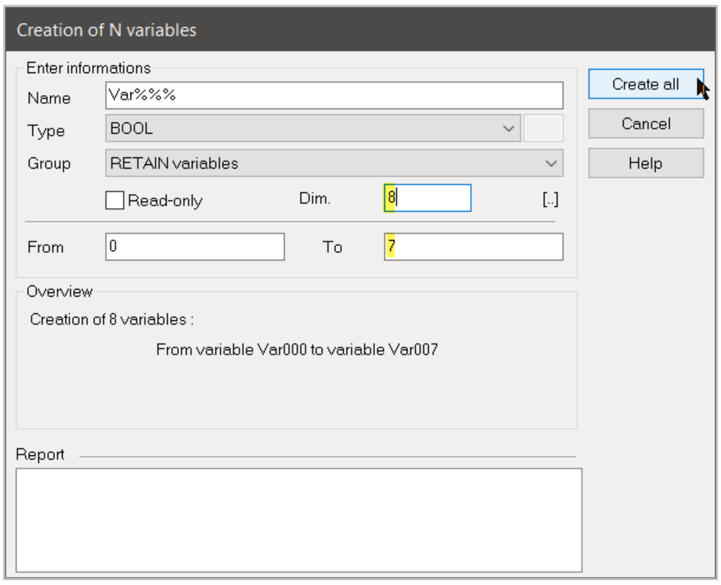
4.1.1 Add retain variables in Workbench

Retain variables are declared in the section of 'RETAIN variables' in variable editor. (Figure below)

1. Double-click „Variables“ in the Workspace.
2. Right-click „RETAIN variables“ in the editor area.
3. Select from the list to add one or multi variables.



<Figure 4.1.1: Retain Variable declaration>


Creation of N variables

Enter informations

Name	Var%%%
Type	BOOL
Group	RETAIN variables
Read-only	☐
Dim.	8
From	0
To	7

Overview

Creation of 8 variables :

From variable Var000 to variable Var007

Report

Here as an example, we choose „Add Multi Variables“, then there shows a pop-up dialog.

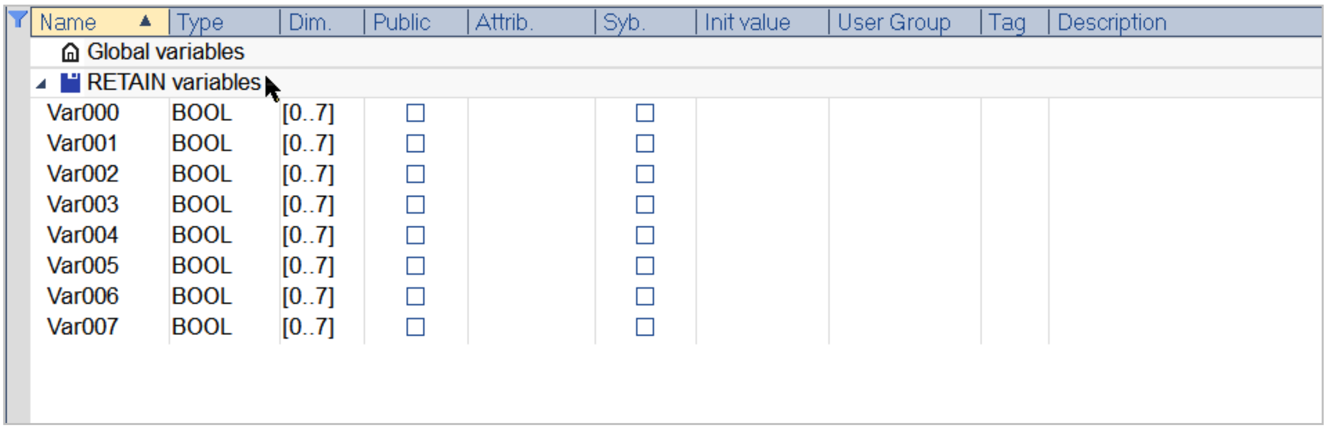
First confirm the „Group“ of variables shows „RETAIN variables“.

Key in total number (dimensions) of the variable. Here we key in „8“ as an example.

Next part, we declare variables from 0 to 7, thus we’ll create 8 retain variables.

Click „Create all“ button. Then click „Cancel“ again to close this dialog.

After closing the dialog, in the RETAIN variables area shows a list with the 8 added variables.

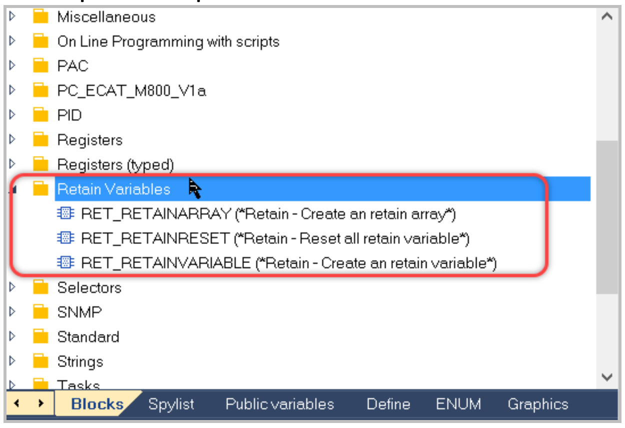


Name	Type	Dim.	Public	Attrib.	Syb.	Init value	User Group	Tag	Description
Global variables									
RETAIN variables									
Var000	BOOL	[0..7]	☐		☐				
Var001	BOOL	[0..7]	☐		☐				
Var002	BOOL	[0..7]	☐		☐				
Var003	BOOL	[0..7]	☐		☐				
Var004	BOOL	[0..7]	☐		☐				
Var005	BOOL	[0..7]	☐		☐				
Var006	BOOL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

Note: When retain variables are modified, the saved retain value will be all erased.

4.2 Persistent Retain Variables

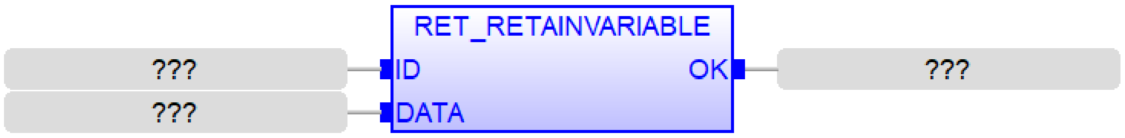
Due to the note in previous section, ICP DAS provide another „Persistent Retain“ to prevent from all value erased. The general idea is ‚each has his own task.‘ We recognize ID of the variables rather than the name to read or write data.
There are 3 function blocks can implement persistent retain:



- Miscellaneous
- On Line Programming with scripts
- PAC
- PC_ECAT_M800_V1a
- PID
- Registers
- Registers (typed)
- Retain Variables
  - RET_RETAINARRAY (*Retain - Create an retain array*)
  - RET_RETAINRESET (*Retain - Reset all retain variable*)
  - RET_RETAINVARIABLE (*Retain - Create an retain variable*)
- Selectors
- SNMP
- Standard
- Strings
- Tasks

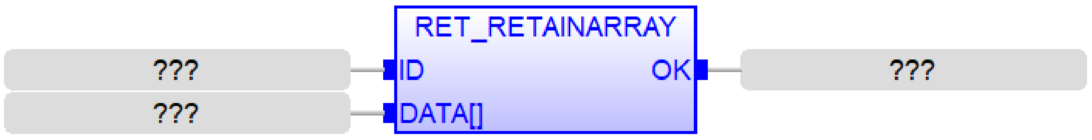
4.2.1 RET_RetainVariable

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



4.2.2 RET_RetainArray

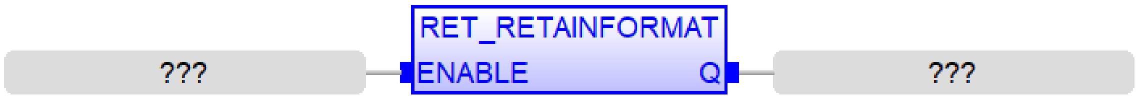
RET_RetainArray		
This Function Block is used to create a Retain Array.		
NAME	TYPE	DESCRIPTION
Input		
ID	UINT	A unique ID number in the Project
Data[ ]	ANY[ ]	Retained array
Output		
OK	BOOL	TRUE: Successful / FALSE: Fail



HINT: If you need a large scale of Retain Data, we suggest to use the method of „ARRAY“ to be more effective.

4.2.3 RET_RetainFormat

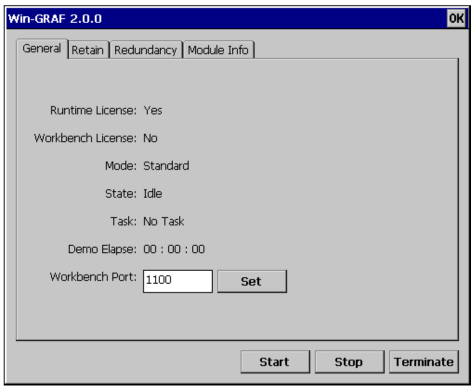
RET_RetainFormat		
This Function Block is used to clear all the Retain Variables.		
NAME	TYPE	DESCRIPTION
Input		
ENABLE	BOOL	Rising edge to trigger CLEAR
Output		
Q	ANY	No output



4.3 Import & Export Retain Variables

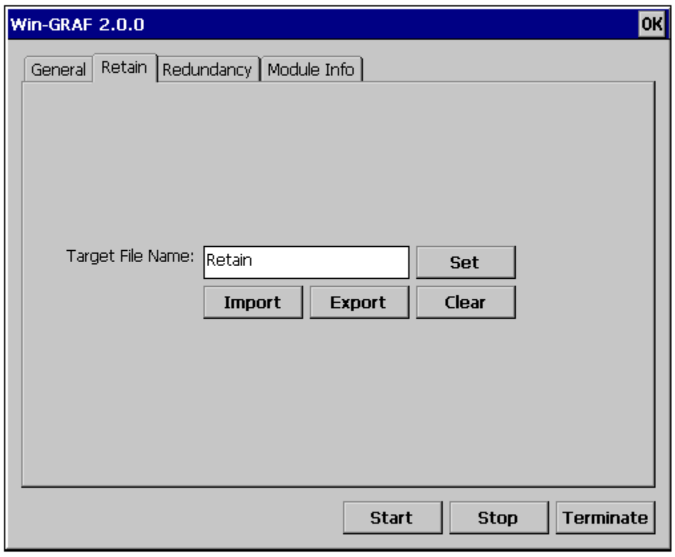
Most of the condition, we can not get or know those variables that are temperarory stored in SRAM. In this section, we will introduce how we get out variables/ settings in SRAM to backup and move to another runtime.

1. Stop your Runtime Project.


Win-GRAF 2.0.0

Runtime License: Yes
Workbench License: No
Mode: Standard
State: Idle
Task: No Task
Demo Elapse: 00 : 00 : 00
Workbench Port: 1100

2. Key in the Target File Name and click „Set“ button. Then click the button „Export“. A dialog pop-ups. Click „OK“ to close it.
Export: SRAM to binery file.
Import: Binery file to SRAM.



Target File Name: Retain

3. Open the System disk of PAC, find out a file name with „Retain“, file type with ‚.RET‘. Left click to copy and backup this file to your PC or another external storage device.

4. Copy the RET file back to system disk of a new PAC. Click the botton „Import“. A dialog pop-ups. Click „OK“ to close it.

5. Click „Start“ to execute Runtime again. Now the data in new PAC continue recording.

4.4 Retain Memory Space

Depend on PACs, the total memory size that store retain variables is different. Following table shows the details:

PAC Number	Memory Type	Total Size	Retain Variable Size	Persistent Retain Size
WP-5238	FRAM	8 KB	4 KB	4 KB
WP-8x28	MRAM	7 KB	3.5 KB	3.5 KB
WP-9x28	MRAM	7 KB	3.5 KB	3.5 KB
XP-8x38	MRAM	131 KB	65.5 KB	65.5 KB
VP-x238	MRAM	7 KB	3.5 KB	3.5 KB
VP-x208	MRAM	7 KB	3.5 KB	3.5 KB

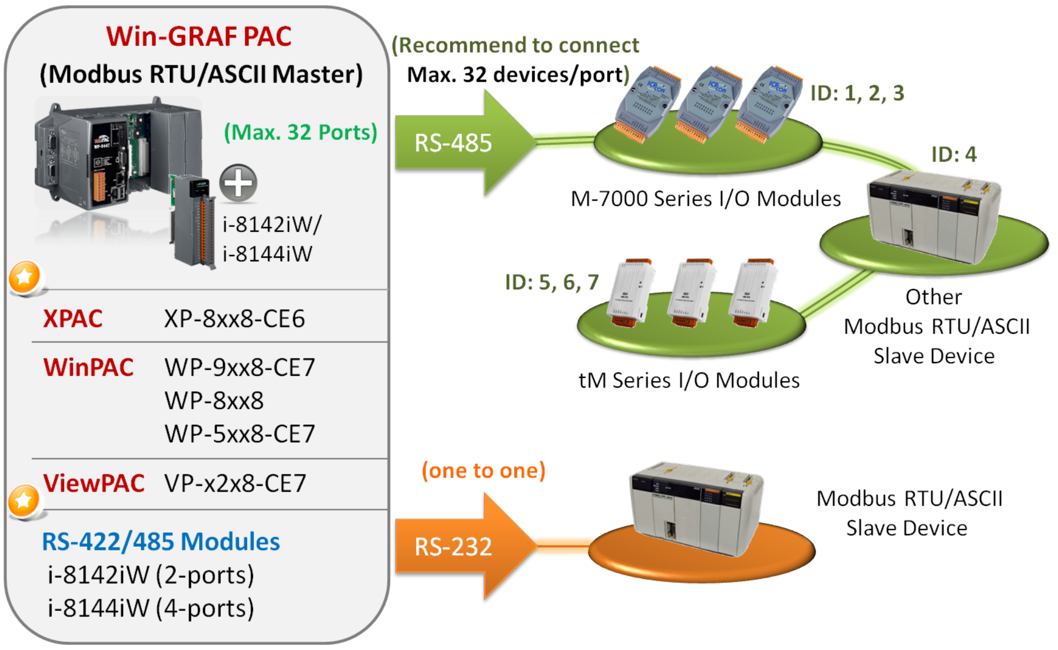
5 Modbus Master: Connecting to Modbus Slave Devices

Win-GRAF Modbus master supports three communication protocols: Modbus TCP, Modbus RTU and Modbus ASCII. Several Modbus masters can be implemented by the same PLC application. Only one master (Modbus RTU, Modbus ASCII) can be created per serial communication port (RS232, RS485, RS422) and several Modbus TCP (Ethernet) masters.

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

The following shows how to create a Modbus RTU master which communicates via the serial port COM1 to a network of Modbus slaves.

Application Diagram:



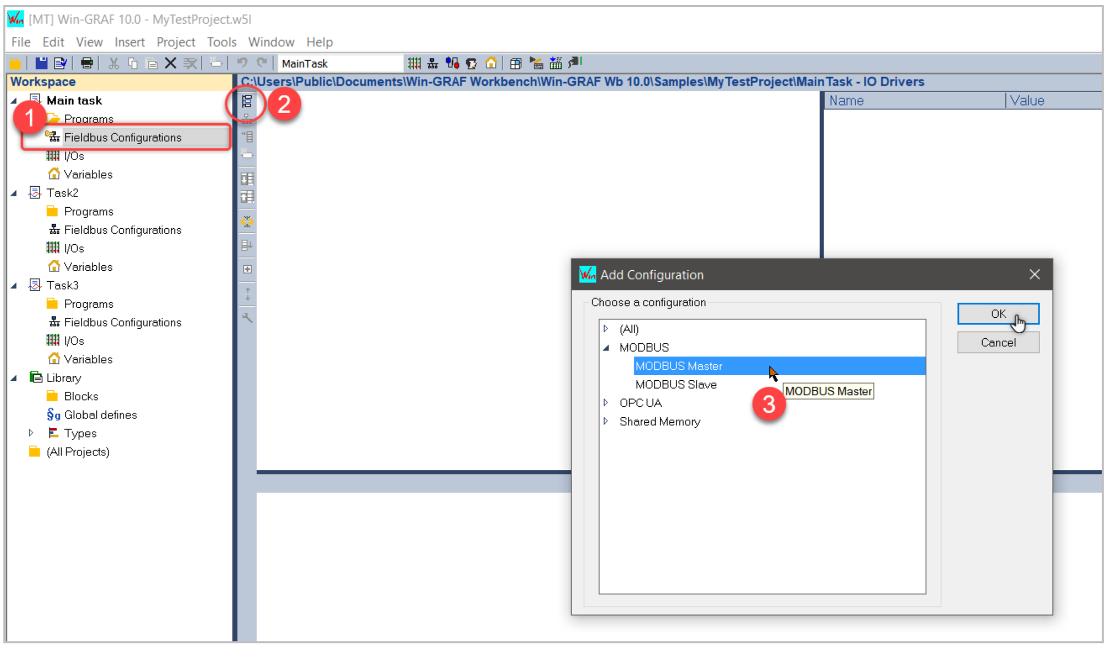
(Check P1-1 to view all PAC models)

5.1.1 Configure Communication Interface

❖ **Select Modbus Master as the fieldbus configuration**

1. Double click ① "Fieldbus Configuration" button, and click ② to "Insert Configuration".

2. From the drop-down list of this pop-up dialog, select ③ "MODBUS Master" and click OK.

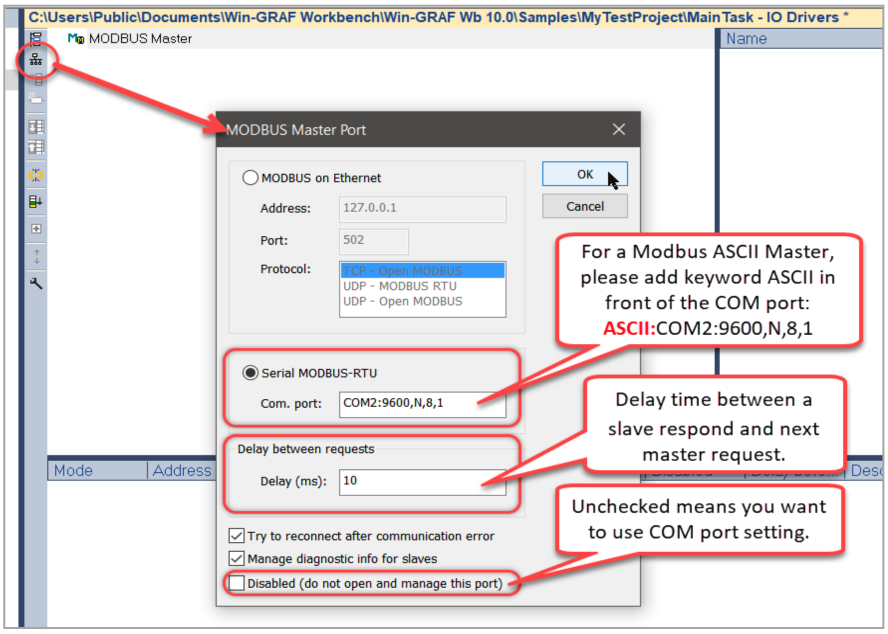


❖ **Add a Modbus RTU master to the workbench and configure it**

1. Click “Insert Master/Port” to open a pop-up dialog. Select lower part of “Serial MODBUS-RTU” and key in relative data to set up COM port. Click OK.

   Note: Serial Port Configuration Format

   [Com Port,Speed,Parity,Data Bits,Stop Bits] → eg. COM3,9600,N,8,1



C:\Users\Public\Documents\Win-GRAF Workbench\Win-GRAF Wb 10.0\Samples\MyTestProject\MainTask - IO Drivers *

MODBUS Master Port

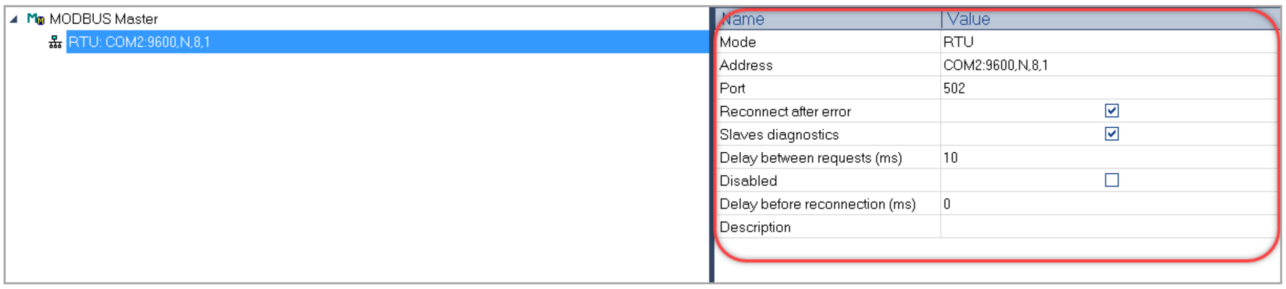
- MODBUS on Ethernet
  - Address: 127.0.0.1
  - Port: 502
  - Protocol: TCP - Open MODBUS / UDP - MODBUS RTU / UDP - Open MODBUS
- [✓ Serial MODBUS-RTU]
  - Com. port: COM2:9600,N,8,1
- Delay between requests
  - Delay (ms): 10
- ☑ Try to reconnect after communication error
- ☑ Manage diagnostic info for slaves
- ☐ Disabled (do not open and manage this port)

For a Modbus ASCII Master, please add keyword ASCII in front of the COM port: **ASCII:**COM2:9600,N,8,1

Delay time between a slave respond and next master request.

Unchecked means you want to use COM port setting.

2. The master communication parameters are shown in the property window and can be modified by either double clicking the master or by directly modifying the settings in the property window.



MODBUS Master
RTU: COM2:9600,N,8,1

Name	Value
Mode	RTU
Address	COM2:9600,N,8,1
Port	502
Reconnect after error	☑
Slaves diagnostics	☑
Delay between requests (ms)	10
Disabled	☐
Delay before reconnection (ms)	0
Description	

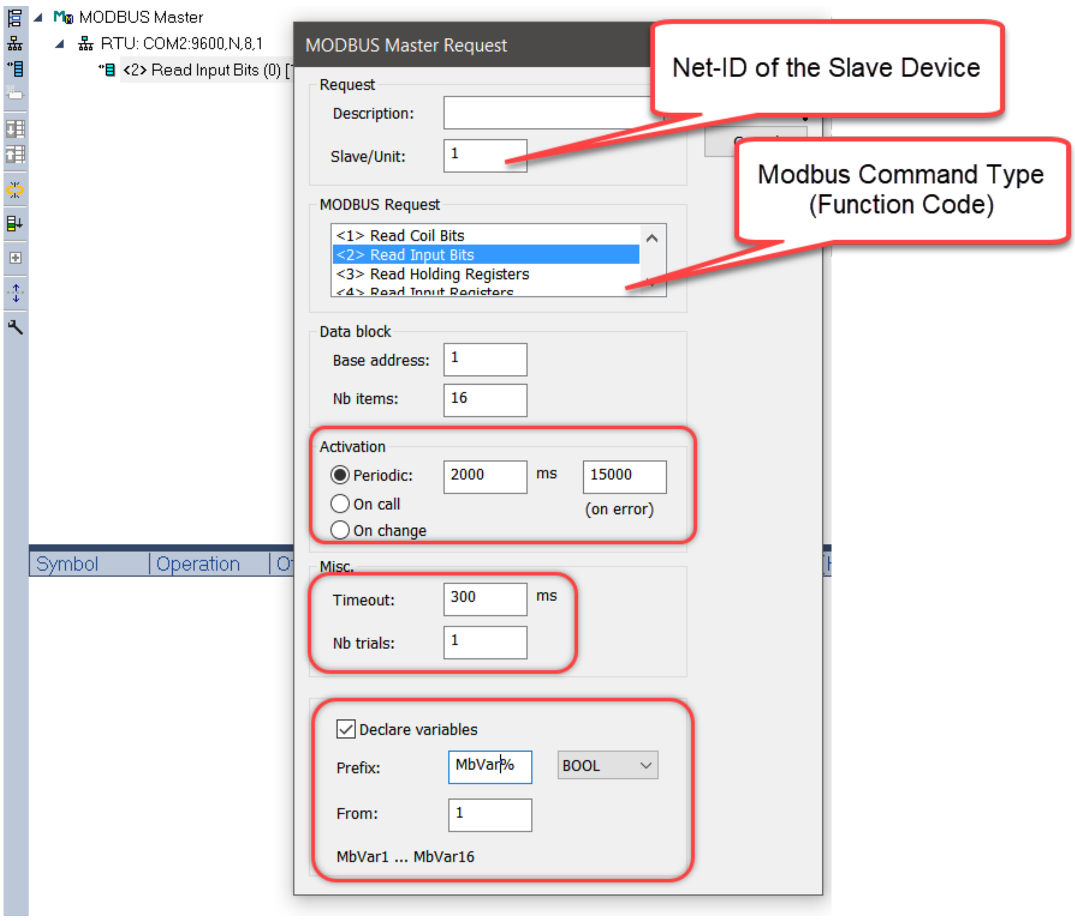
❖ **Add a data block to the master which stores input data received from the slave and output data to be sent to the slave.**

Configure the data block which holds data from the slave. The data block setup of the master should be an exact representation of the slave table. In other word it should be an image of the slave data block in size and data type. Data read from the slave is being stored to the read section of the data block and data which has been modified in the write section of the data block is being sent to the slave.

Type	Function Code	Modbus Request	Description
Read	1	Read coil bits	Read digital output (DO) data
	2	Read input bits	Read digital input (DI) data
	3	Read holding registers	Read analog output (AO) data
	4	Read input registers	Read analog input (AI) data
Write	5	Write single coil bit	Write digital output (DO) data
	6	Write single holding register	Write one analog output (AO) data (16-bits)
	15	Write coil bits	Write multiple digital output (DO) data.
	16	Write Holding Registers	Write multiple analog output (AO) data (16/32 bits)

<Table 5.1: Function Code Table>

1. Following figure creates an input data block which holds 16 input bits and start reading from the slave discreate input table at address 1. (Example: the number is set to 16).
   a) Click the 'Insert Slave/Data Block' button on the left side to create a data block.
   b) Slave/Unit: Enter the Net-ID of the Slave device. (Example: Remote slave ID is '1').
   c) MODBUS Request: Select the Modbus command type (function code) to be used for accessing the slave table <Function code Table>. Example: <2> Read Input Bits



Request
Description:
Slave/Unit: 1

MODBUS Request
<1> Read Coil Bits
<2> Read Input Bits
<3> Read Holding Registers
<4> Read Input Registers

Data block
Base address: 1
Nb items: 16

Activation
(●) Periodic: 2000 ms 15000 (on error)
( ) On call
( ) On change

Misc.
Timeout: 300 ms
Nb trials: 1

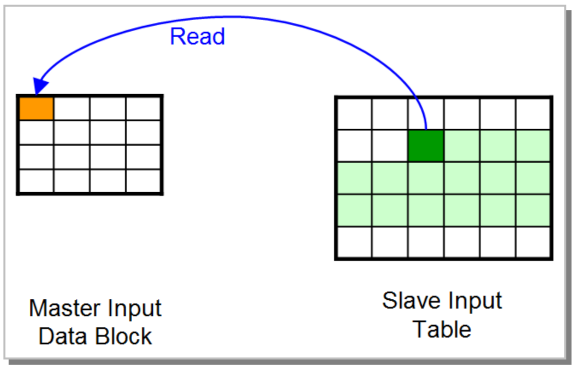
[✓] Declare variables
Prefix: MbVar% BOOL
From: 1
MbVar1 ... MbVar16

<Pic 5.1: Modbus Data Block Setting>

2. Each data block is identified by a MODBUS slave number, a base address and a number of items/entries (bits or words). The number of items/entries is limited by MODBUS (2000 bits read, 1968 bits forced, 125 words read or 120 words forced).

3. **Base address**: Starts from '1' by default. It indicates the start address of the Modbus slave table from which the master starts to read. The data read from the start address will be copied to the first position of the master input block data area.
   Note: If you want to change the 'Base address', right-click the 'MODBUS Master' and then select the 'MODBUS Master Addresses' to modify the value.

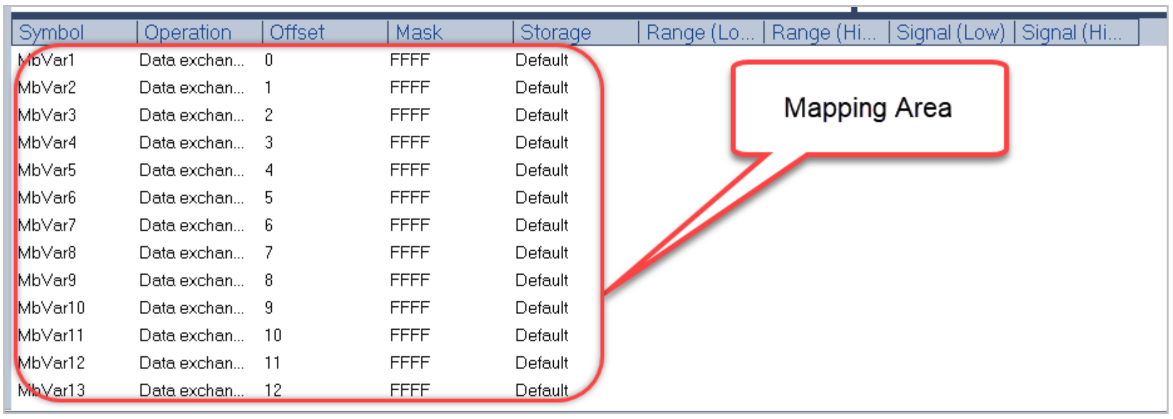
4. **Nb items**: The number of slave table entries to read.



<Base address set to 9 and Nb to 16>

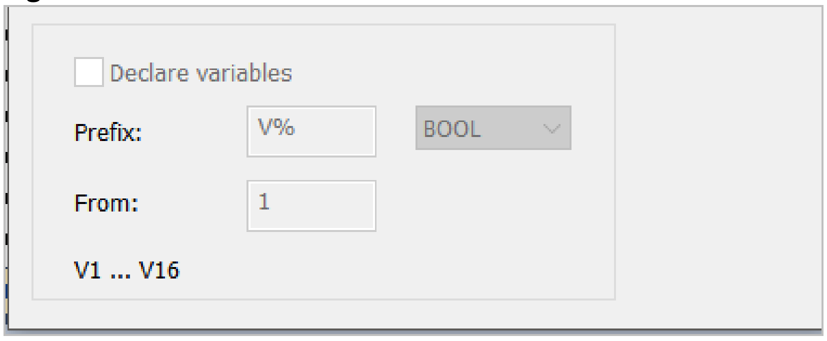
5. **Activation**: Set how the Modbus request is being triggered by the master.
6. **Periodic**: Sending the request periodically. (Example: send once every two seconds.) 'on error' means the next sending time after a Modbus exception occurred (e.g., 15 seconds).
7. **On call**: The request is activated via a program call by using a variable. If the variable mapped to 'Command (one shot)' operation turns from false to true, a Modbus request is being triggered.
8. **On change**: The request is triggered once a variable mapped to the output area has changed. This option is can only be used for writing and not reading commands.
9. **Timeout**: Set a timeout value. If the slave does not respond to the master request within the timeout period, a timeout error will be generated. (The recommended timeout value for the Modbus RTU/ASCII communication is between 200 and 1000 ms.).
10. **Nb Trials**: If the slave fails to respond or the master receives an invalid response, the master will then retry for the configured number of retries before moving on to the next command in the list.
11. **Declare variables**: This option allows the user to declare new variables and automatically map them to the Modbus master data block. Enter a variable name and add the symbol '%' at the end of the name and enter the start number for the '%' symbol. The workbench replaces the '%' with a number which increases incrementally for each new variable as indicated at the bottom of the window. The new variables are automatically mapped to the Modbus master data block (See following figure).

    Note: Disable this option if variables have already been declared in the variable editor. In this case the variables have to be mapped manually by dragging the variable from the variable editor and dropping it to the mapping area.



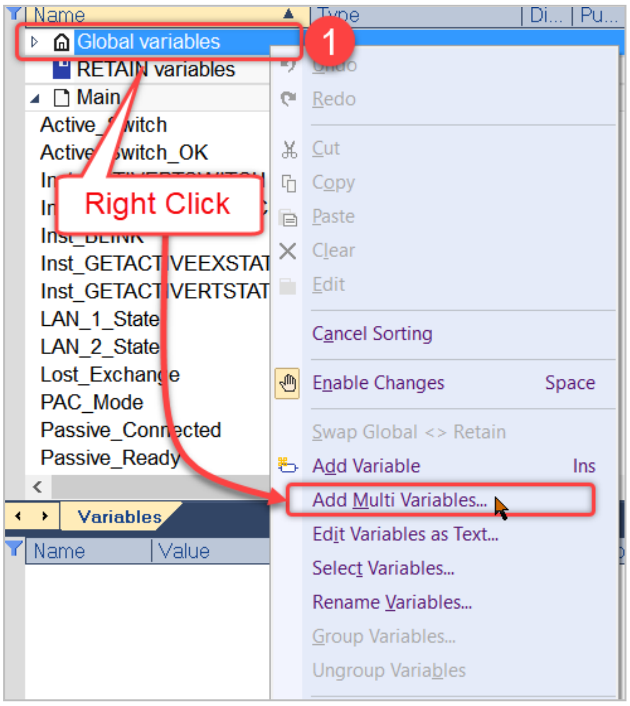
Symbol	Operation	Offset	Mask	Storage	Range (Lo...	Range (Hi...	Signal (Low)	Signal (Hi...
MbVar1	Data exchan...	0	FFFF	Default				
MbVar2	Data exchan...	1	FFFF	Default				
MbVar3	Data exchan...	2	FFFF	Default				
MbVar4	Data exchan...	3	FFFF	Default				
MbVar5	Data exchan...	4	FFFF	Default				
MbVar6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

❖ **Following step has only to be done if the 'Declare variables' option has not been selected as Figure below.**

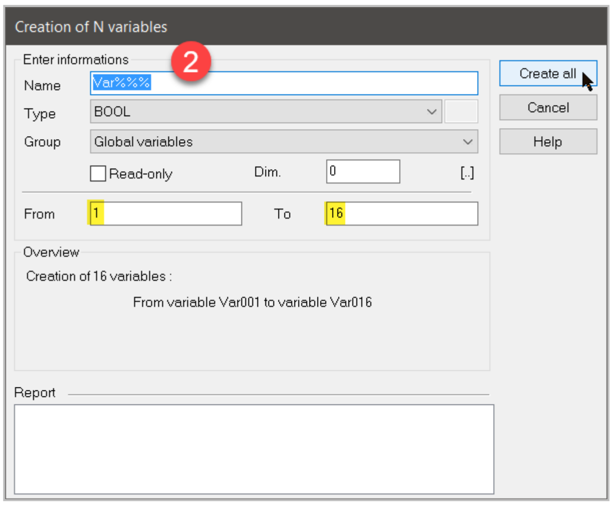

Declare variables
Prefix: V% BOOL
From: 1
V1 ... V16

Declare new variables and drag and drop the variable to the data block mapping area.

1. Right click ① "Global Variables" to add Multi Variables.



2. ②Key in Variable Name with "%%%" in the end, select Type and the variable numbers (here as an example from 1 to 16), and click "Create all". Variables will show in the top-right column. Click Cancel to close the N variable dialog.


Creation of N variables

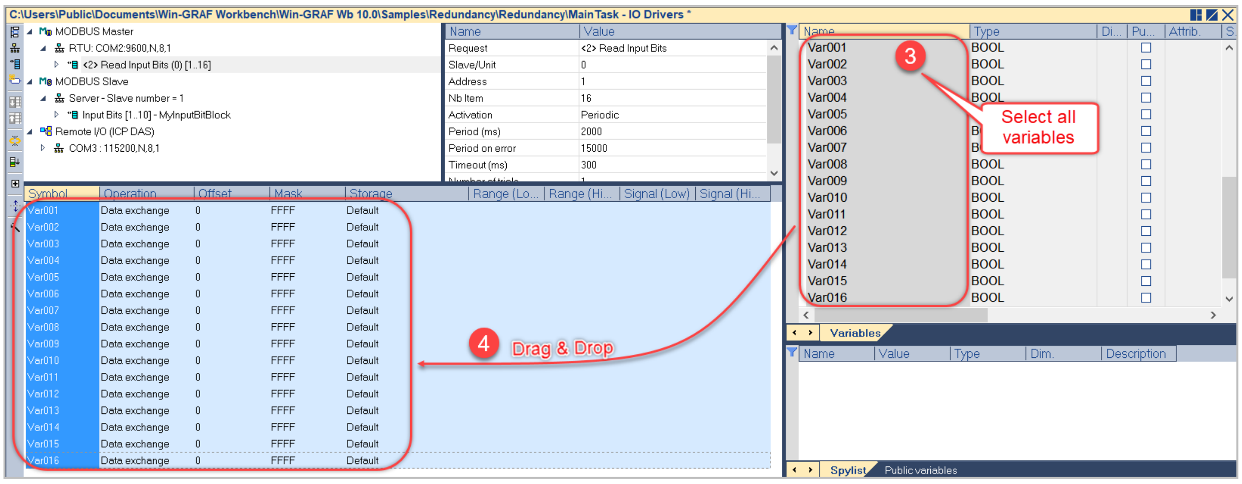
Enter informations
Name: Var%%%
Type: BOOL
Group: Global variables
Read-only
Dim. 0 [.]
From 1 To 16

Create all
Cancel
Help

Overview
Creation of 16 variables :
From variable Var001 to variable Var016

Report

3. ③Select all 16 variables just created and ④drag and drop back to the Tabs area. Now the offset numbers are all "0".


C:\Users\Public\Documents\Win-GRAF Workbench\Win-GRAF Wb 10.0\Samples\Redundancy\Redundancy\MainTask - IO Drivers *

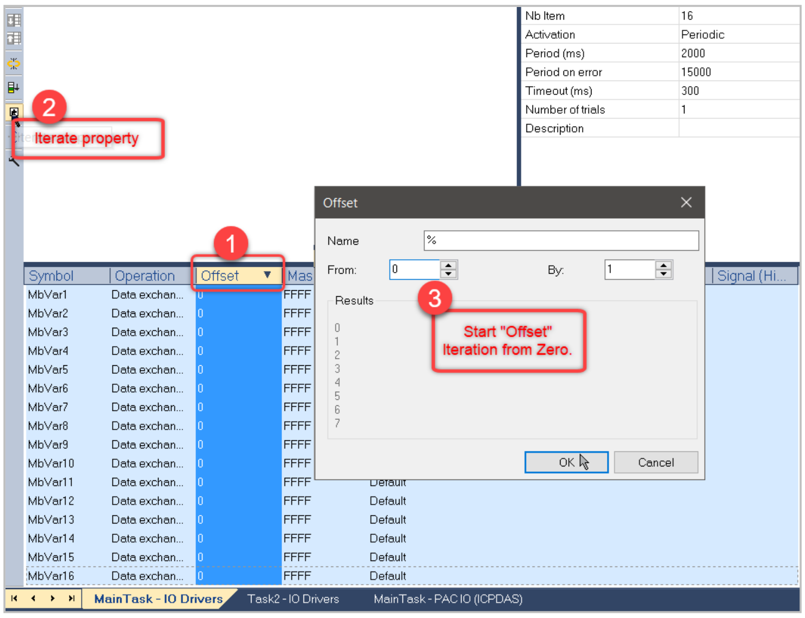
- MODBUS Master
  - RTU: COM2:9600,N,8,1
    - <2> Read Input Bits (0) [1..16]
- MODBUS Slave
  - Server - Slave number = 1
    - Input Bits [1..10] - MyInputBitBlock
- Remote I/O (ICP DAS)
  - COM3 : 115200,N,8,1

Name	Value
Request	<2> Read Input Bits
Slave/Unit	0
Address	1
Nb Item	16
Activation	Periodic
Period (ms)	2000
Period on error	15000
Timeout (ms)	300

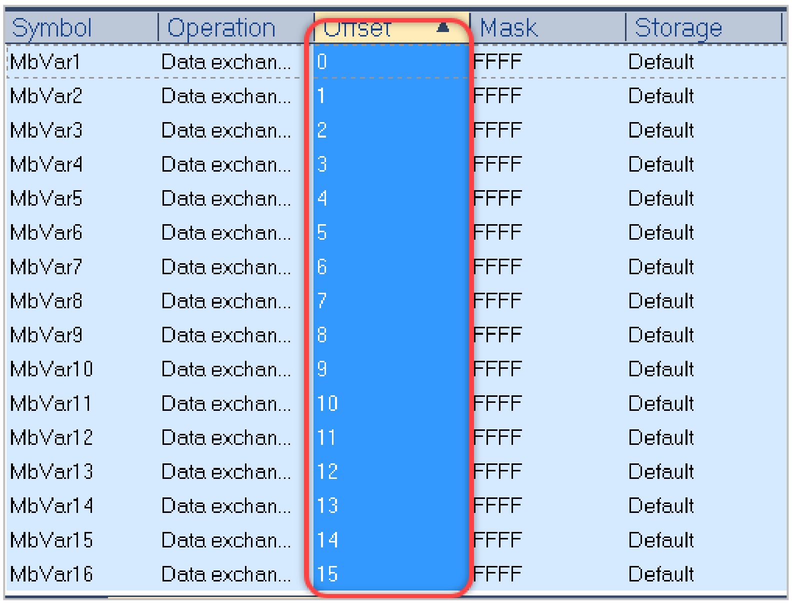
Symbol	Operation	Offset	Mask	Storage
Var001	Data exchange	0	FFFF	Default
Var002	Data exchange	0	FFFF	Default
Var003	Data exchange	0	FFFF	Default
Var004	Data exchange	0	FFFF	Default
Var005	Data exchange	0	FFFF	Default
Var006	Data exchange	0	FFFF	Default
Var007	Data exchange	0	FFFF	Default
Var008	Data exchange	0	FFFF	Default
Var009	Data exchange	0	FFFF	Default
Var010	Data exchange	0	FFFF	Default
Var011	Data exchange	0	FFFF	Default
Var012	Data exchange	0	FFFF	Default
Var013	Data exchange	0	FFFF	Default
Var014	Data exchange	0	FFFF	Default
Var015	Data exchange	0	FFFF	Default
Var016	Data exchange	0	FFFF	Default

Name	Type
Var001	BOOL
Var002	BOOL
Var003	BOOL
Var004	BOOL
Var005	BOOL
Var006	BOOL
Var007	BOOL
Var008	BOOL
Var009	BOOL
Var010	BOOL
Var011	BOOL
Var012	BOOL
Var013	BOOL
Var014	BOOL
Var015	BOOL
Var016	BOOL

4. **Offset**:
Set the data block offset of each variable. Each variable should have a different offset number otherwise memory overlap occurs.


Name	Value
Nb Item	16
Activation	Periodic
Period (ms)	2000
Period on error	15000
Timeout (ms)	300
Number of trials	1
Description	

Offset
Name: %
From: 0 By: 1
Results: 0 1 2 3 4 5 6 7

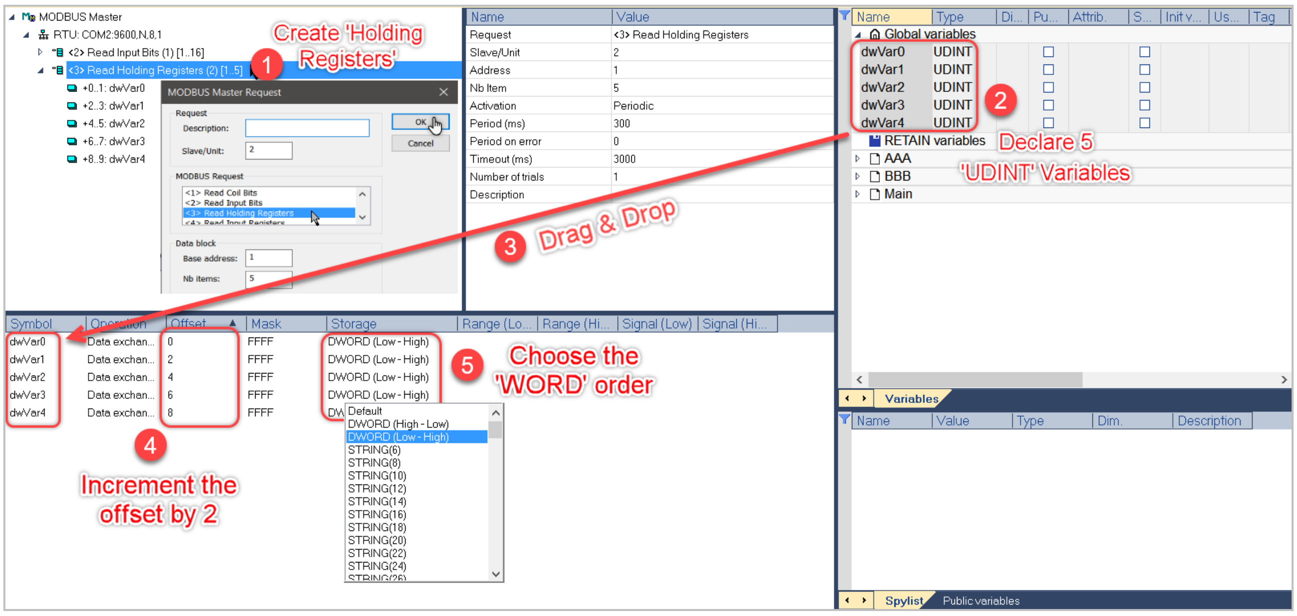

Symbol	Operation	Offset	Mask	Storage
MbVar1	Data exchan...	0	FFFF	Default
MbVar2	Data exchan...	1	FFFF	Default
MbVar3	Data exchan...	2	FFFF	Default
MbVar4	Data exchan...	3	FFFF	Default
MbVar5	Data exchan...	4	FFFF	Default
MbVar6	Data exchan...	5	FFFF	Default
MbVar7	Data exchan...	6	FFFF	Default
MbVar8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

5. **Mask**:
   The mask in the mapping area only has to be set for input and holding registers and is being ignored for coil and input bits.
   A Modbus register has a memory size of 16 bit (same as INT, UINT, WORD). The mask (hex number) can be used to filter out bits from the register.

<Example>

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

6. **Storage**:
   The storage column is only relevant for the input and holding register setting. For *exchanging 32 bit variables* (DINT, REAL...), two consecutive Modbus register can be mapped to one variable. For *exchanging strings* multiple register can be mapped to one string.



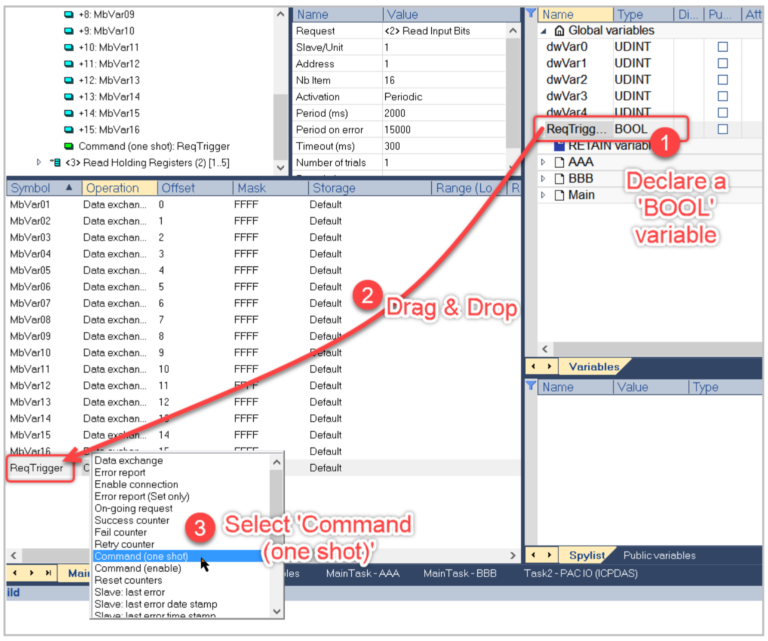
Name	Value
Request	<3> Read Holding Registers
Slave/Unit	2
Address	1
Nb Item	5
Activation	Periodic
Period (ms)	300
Period on error	0
Timeout (ms)	3000
Number of trials	1
Description	

Name	Type	Di...	Pu...	Attrib.	S...	Init v...	Us...	Tag
Global variables								
dwVar0	UDINT		☐		☐			
dwVar1	UDINT		☐		☐			
dwVar2	UDINT		☐		☐			
dwVar3	UDINT		☐		☐			
dwVar4	UDINT		☐		☐			
RETAIN variables								
AAA								
BBB								
Main								

Symbol	Operation	Offset	Mask	Storage	Range (Lo...	Range (Hi...	Signal (Low)	Signal (Hi...
dwVar0	Data exchan...	0	FFFF	DWORD (Low - High)				
dwVar1	Data exchan...	2	FFFF	DWORD (Low - High)				
dwVar2	Data exchan...	4	FFFF	DWORD (Low - High)				
dwVar3	Data exchan...	6	FFFF	DWORD (Low - High)				
dwVar4	Data exchan...	8	FFFF	DW				

<Pic 5.2: Mapping 32-bit variable to holding register>

❖ **This step has only to be done if the *'On call'* checkbox has been selected ('Activation' part in Pic. 5.1).** If the master is in the *'On call'* mode, the Win-GRAF runtime does not automatically send a request to the slave. The logic program of the PLC has to trigger a request. Two command types are available to initiate a request:
1. Command (one shot)
- A request will be sent once the attached flag (Pic 5.3: *'ReqTrigger'*) changes from FALSE to TRUE. Each time the PLC logic switches the flag to TRUE one Modbus command is being sent to the slave. After the request has been sent the flag will automatically be reset to zero by the runtime.
2. Command (Enable)
- Once the BOOL variable mapped to the *'Command (Enable)'* operation is set TRUE by the PLC logic the Modbus master will send requests to the slave until the variable is set to FALSE. No commands will be sent if the variable is set to FALSE.



Name	Value
Request	<2> Read Input Bits
Slave/Unit	1
Address	1
Nb Item	16
Activation	Periodic
Period (ms)	2000
Period on error	15000
Timeout (ms)	300
Number of trials	1

Symbol	Operation	Offset	Mask	Storage
MbVar01	Data exchan...	0	FFFF	Default
MbVar02	Data exchan...	1	FFFF	Default
MbVar03	Data exchan...	2	FFFF	Default
MbVar04	Data exchan...	3	FFFF	Default
MbVar05	Data exchan...	4	FFFF	Default
MbVar06	Data exchan...	5	FFFF	Default
MbVar07	Data exchan...	6	FFFF	Default
MbVar08	Data exchan...	7	FFFF	Default
MbVar09	Data exchan...	8	FFFF	Default
MbVar10	Data exchan...	9	FFFF	Default
MbVar11	Data exchan...	10	FFFF	Default
MbVar12	Data exchan...	11	FFFF	Default
MbVar13	Data exchan...	12	FFFF	Default
MbVar14	Data exchan...	13	FFFF	Default
MbVar15	Data exchan...	14	FFFF	Default
MbVar16	Data exchan...	15	FFFF	Default
ReqTrigger	C			Default

Name	Type
Global variables	
dwVar0	UDINT
dwVar1	UDINT
dwVar2	UDINT
dwVar3	UDINT
dwVar4	UDINT
ReqTrigg...	BOOL
RETAIN variabl	
AAA	
BBB	
Main	

<Pic 5.3: Create a "On call" variable>

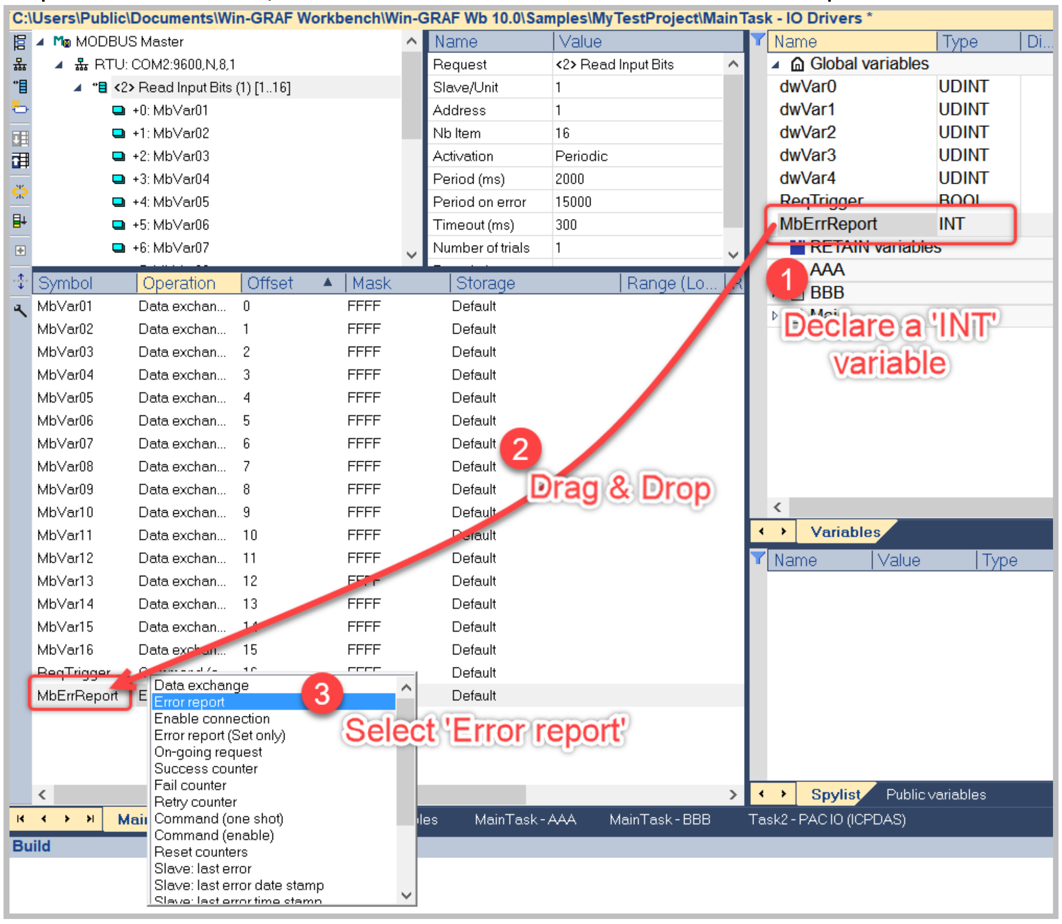
❖ **Map a variable to the diagnostic and status information of the master.** Information like communication timeout, invalid data address, invalid command, number of failed request etc. are recorded by the master. In the following a variable will be mapped to the *'Error report'* operation which records the error of the last request. Table 5.2 shows all the error codes used by the *'Error report'* operation.

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

<Table 5.2: Error code for 'Error report' operation>

Procedure for implementing the 'Error report' procedure (Pic 5.4):
1. Declare a INT variable (e.g. 'MbErrReport').
2. Drag and drop the new variable to the mapping list.

3. Double click the 'Operation' column next to the variable and select 'Error report'. The offset and storage column setting is ignored by this operation. The variable will show an error code when a Modbus request error occurs, and will be reset to zero after the next request was successful.



C:\Users\Public\Documents\Win-GRAF Workbench\Win-GRAF Wb 10.0\Samples\MyTestProject\MainTask - IO Drivers *

- MODBUS Master
  - RTU: COM2:9600,N,8,1
    - <2> Read Input Bits (1) [1..16]
      - +0: MbVar01
      - +1: MbVar02
      - +2: MbVar03
      - +3: MbVar04
      - +4: MbVar05
      - +5: MbVar06
      - +6: MbVar07

Name	Value
Request	<2> Read Input Bits
Slave/Unit	1
Address	1
Nb Item	16
Activation	Periodic
Period (ms)	2000
Period on error	15000
Timeout (ms)	300
Number of trials	1

Name	Type	Di...
Global variables		
dwVar0	UDINT	
dwVar1	UDINT	
dwVar2	UDINT	
dwVar3	UDINT	
dwVar4	UDINT	
ReqTrigger	BOOL	
MbErrReport	INT	
RETAIN variables		
AAA		
BBB		

Symbol	Operation	Offset	Mask	Storage	Range (Lo...
MbVar01	Data exchan...	0	FFFF	Default	
MbVar02	Data exchan...	1	FFFF	Default	
MbVar03	Data exchan...	2	FFFF	Default	
MbVar04	Data exchan...	3	FFFF	Default	
MbVar05	Data exchan...	4	FFFF	Default	
MbVar06	Data exchan...	5	FFFF	Default	
MbVar07	Data exchan...	6	FFFF	Default	
MbVar08	Data exchan...	7	FFFF	Default	
MbVar09	Data exchan...	8	FFFF	Default	
MbVar10	Data exchan...	9	FFFF	Default	
MbVar11	Data exchan...	10	FFFF	Default	
MbVar12	Data exchan...	11	FFFF	Default	
MbVar13	Data exchan...	12	FFFF	Default	
MbVar14	Data exchan...	13	FFFF	Default	
MbVar15	Data exchan...	14	FFFF	Default	
MbVar16	Data exchan...	15	FFFF	Default	
ReqTrigger	Command (o...	16	FFFF	Default	
MbErrReport	E			Default	

Variables

Name	Value	Type

Spylist Public variables

MainTask - AAA MainTask - BBB Task2 - PAC IO (ICPDAS)

Build

<Pic 5.4: Error Report Setting>

6 Modbus Slave: Connecting to Modbus Master Devices

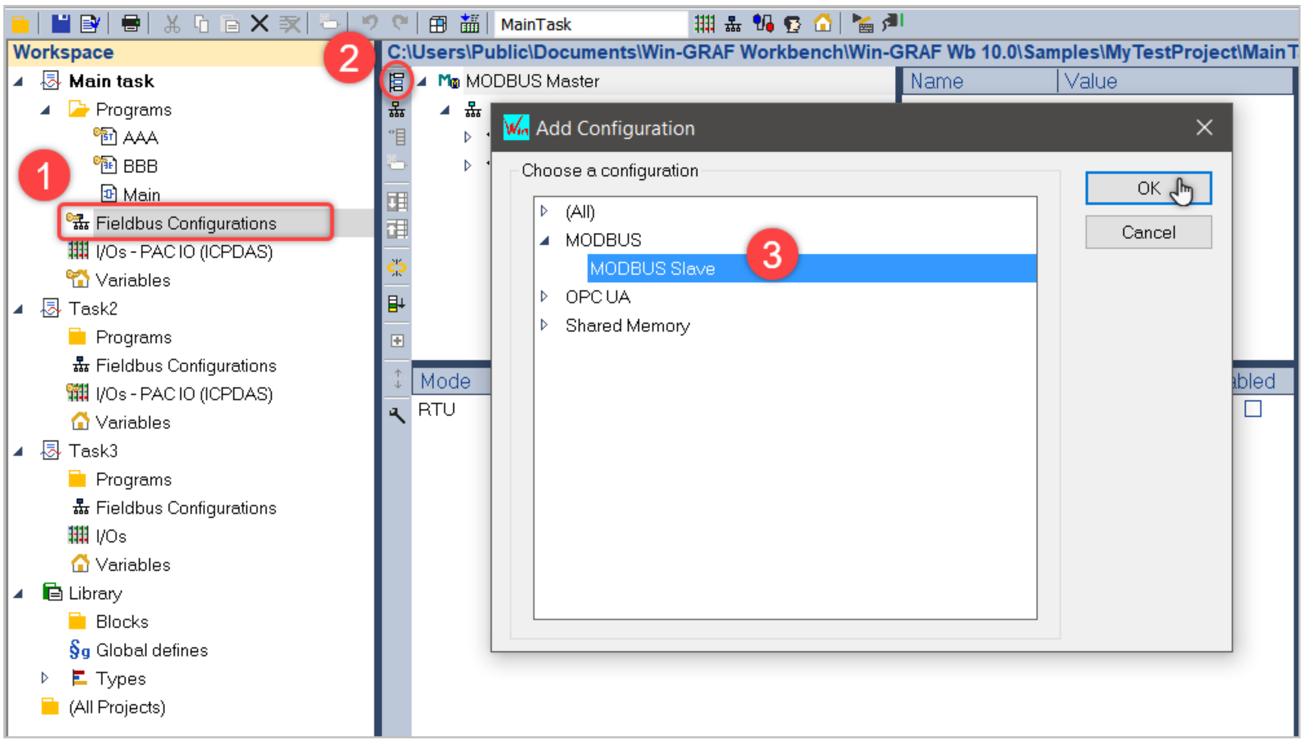
This chapter describes how to setup the runtime to act as a Modbus slave. Three types of slaves can be installed: Modbus TCP, Modbus RTU and Modbus ASCII. The Win-GRAF runtime can act as a multiple slave.

The Modbus slave of the Win-GRAF runtime first has to be configured via the workbench in order for the remote Modbus master to access it. The communication layer (Ethernet, serial), the register types and number of registers (data block) of the slave have to be set. Below we will introduce in sequence:

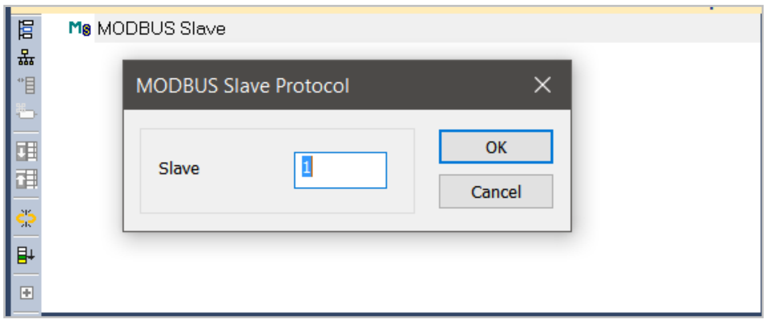
6.1 Slave Data Block Configuration

6.1.1 Selectiong Slave

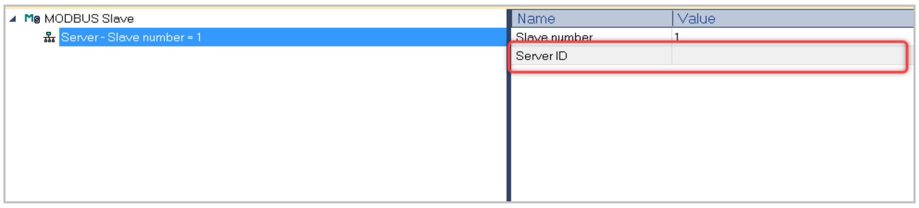
1. Double click "Fieldbus Configuration" to open IO Drivers window. Click "Insert Configuration" from the left-top window. Select Modbus→select "MODBUS Slave" then click "OK".



2. Click "Insert Master/Port" button to set the slave number (here we key in '1') and then click "OK".



Note: If you add more than one slave, assign each slave a ID to identify the server in the PLC program.



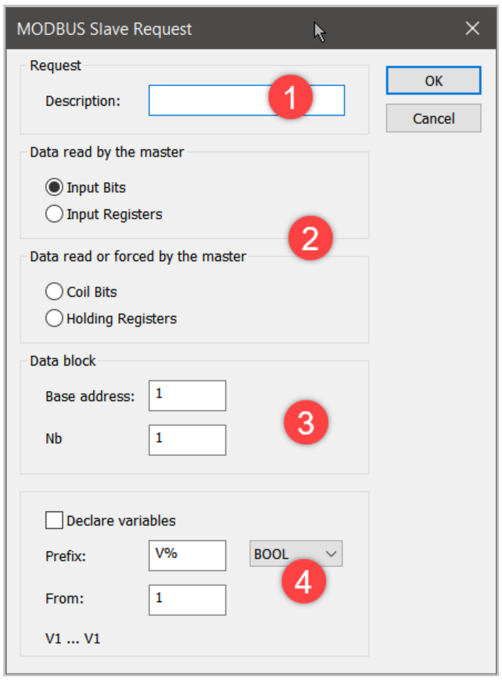
Name	Value
Slave number	1
Server ID	

6.1.2 Define Slave Register

Following standard Modbus register and coils are supported by Win-GRAF:

- Input register (read by masters)
- Holding register (read/write by master)
- Coils (read/write by master)
- Discrete inputs (read by masters)

Modbus slave register type and size have to be configured through following figure.


MODBUS Slave Request

Request

Description:

Data read by the master

(● Input Bits) ○ Input Registers

Data read or forced by the master

○ Coil Bits ○ Holding Registers

Data block

Base address: 1

Nb: 1

☐ Declare variables

Prefix: V% BOOL

From: 1

V1 ... V1

Description as below:

1. Description field allows you to shortly describe the purpose of the data block. This field can be left empty.
2. First decide whether the master should only have read access or have both read and write access to the data block. In addition, decide the data format of the block.

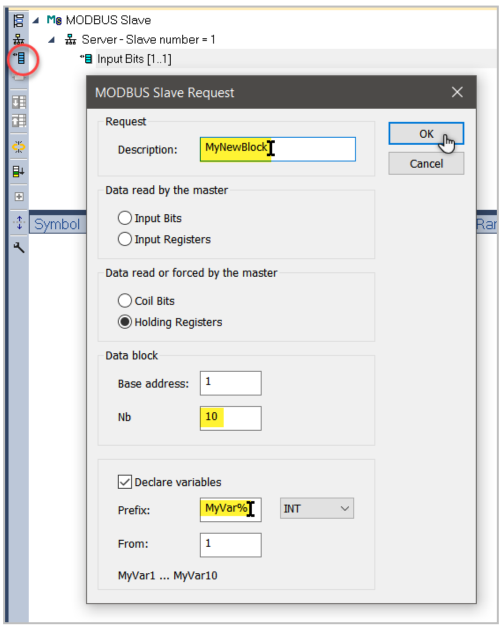
Access	Option	Data type
Read	Input Bits	BOOL
	Input Register	BYTE, INT, DINT, REAL, etc.
Read/ Write	Coil Bits	BOOL
	Holding Register	BYTE, INT, DINT, REAL, etc.

3. Set the start address (base address) of the number of register (Nb) for the data block. Depending on the selection made in 2 the register size unit is either BIT or WORD. It is recommended to set the start address to 1. If two blocks of the same register type are added, make sure the start address of the second block continuous with the end address of the first block. This allows the master to access the two blocks in one datagram and decrease the number of datagram exchange between master and slave. If the data address requested from the Modbus Master (e.g., the SCADA software) is smaller than the start address or greater than the maximum address (start address + Nb -1) than the slave of the Win-GRAF runtime will not respond.
4. Required the workbench automatically declares new variables and directly map them to the newly created data block.

6.1.3 Define Holding Register

❖ This figure describes how to add a holding register data block.

1. Click „Insert Slave/ Data Block“ to open slave configuration. Key in the name ‚MyNewBlock‘ and select ‚Holding Registers‘.
2. Set the Holding Register to have 10 registers (Nb=10) and base address at 1.
3. Check the box to declare a number of 10 variables with data type ‚INT‘ and map it to the data block. Key in the variable name as ‚MyVar%‘; the ‚%‘ represents a value which starts from 1.
4. Click ‚OK‘ to finish.


MODBUS Slave
Server - Slave number = 1
Input Bits [1..1]

MODBUS Slave Request

Request

Description: MyNewBlock

Data read by the master

○ Input Bits ○ Input Registers

Data read or forced by the master

○ Coil Bits (● Holding Registers)

Data block

Base address: 1

Nb: 10

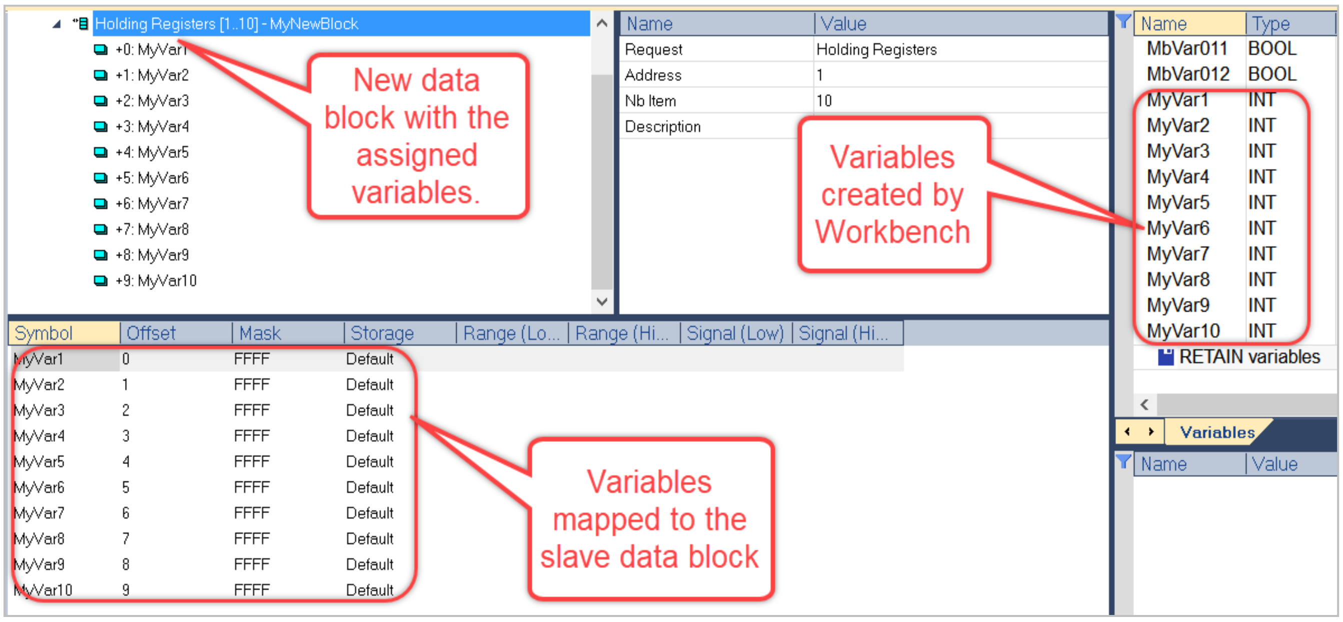
☑ Declare variables

Prefix: MyVar% INT

From: 1

MyVar1 ... MyVar10

After add “MyNewBlock” to the Modbus slave and new data block, new global variables are automatically declared and assigned to the data block.


New data block with the assigned variables.
Variables created by Workbench
Variables mapped to the slave data block
RETAIN variables
Variables

Holding Registers [1..10] - MyNewBlock
- +0: MyVar1
- +1: MyVar2
- +2: MyVar3
- +3: MyVar4
- +4: MyVar5
- +5: MyVar6
- +6: MyVar7
- +7: MyVar8
- +8: MyVar9
- +9: MyVar10

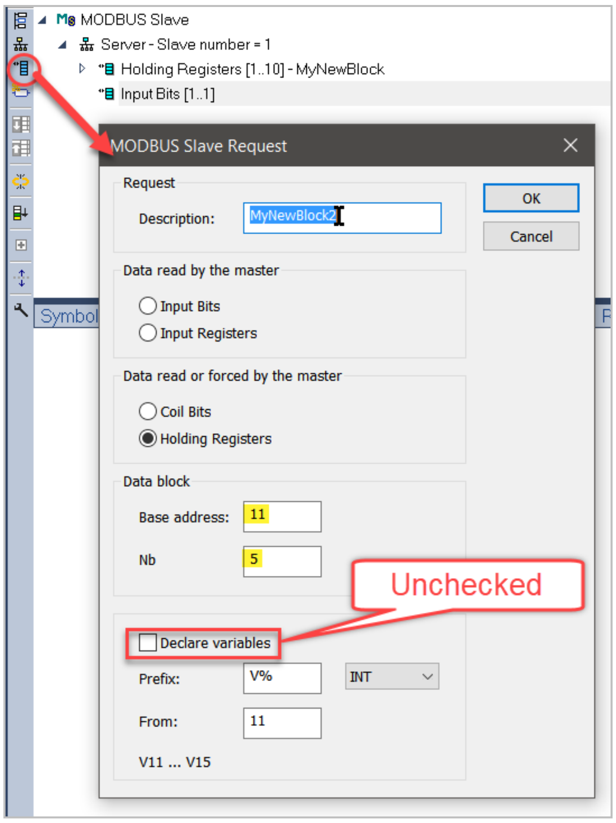
Name	Value
Request	Holding Registers
Address	1
Nb Item	10
Description	

Name	Type
MbVar011	BOOL
MbVar012	BOOL
MyVar1	INT
MyVar2	INT
MyVar3	INT
MyVar4	INT
MyVar5	INT
MyVar6	INT
MyVar7	INT
MyVar8	INT
MyVar9	INT
MyVar10	INT

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

❖ This figure describes how to add a holding register and to map global variables.

1. Click “Insert Slave/ Data Block” to open the Slave Request configuration table. Key in “MyNewBlock2” as the name.
2. Add a data block of the holding register type which holds 5 registers (Nb=5) and the first register address starts (base address) at 11. Because of the previous block ends with 10, so here we continue with 11 in order to keep consistence address range for the same Holding Register.
3. Later we will declare and map the variables for the holding register data block by users so we leave this box unchecked. Then click “OK”.



MODBUS Slave
- Server - Slave number = 1
  - Holding Registers [1..10] - MyNewBlock
  - Input Bits [1..1]

MODBUS Slave Request

Request
Description: MyNewBlock2

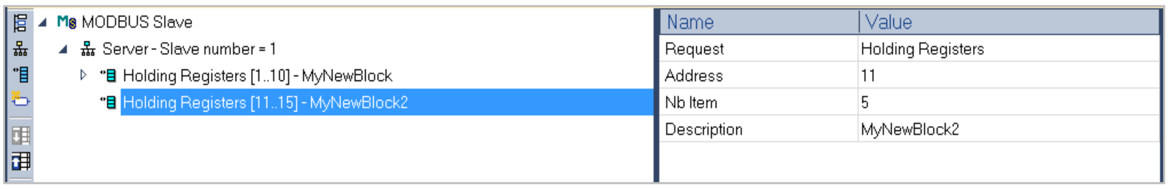
Data read by the master
- Input Bits
- Input Registers

Data read or forced by the master
- Coil Bits
- [✓ Holding Registers]

Data block
Base address: 11
Nb: 5

[ ] Declare variables
Prefix: V% INT
From: 11
V11 ... V15

Following capture shows “MyNewBlock2” in workbench window.



MODBUS Slave
- Server - Slave number = 1
  - Holding Registers [1..10] - MyNewBlock
  - Holding Registers [11..15] - MyNewBlock2

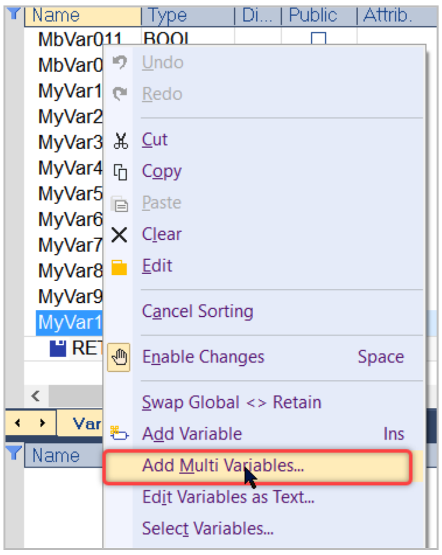
Name	Value
Request	Holding Registers
Address	11
Nb Item	5
Description	MyNewBlock2

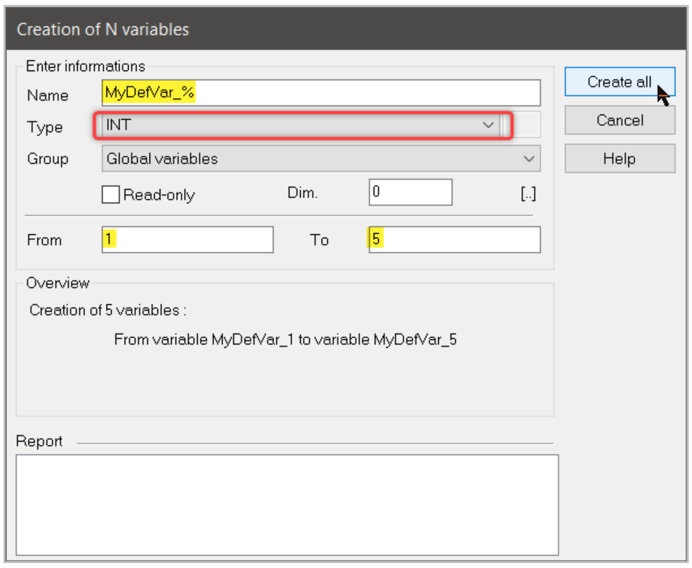
Now we start to declare 5 global variables of INT type.
1. Right click a global variable and select "Add Multi Variables…"

2. Key in variable name (here is 'MyDefVar_%'). The % symble indicates a number attached to the variable name at an incremental order. Select the variable type to declare (later we have to map INT type so we select 'INT').

3. Start number has to be entered in the 'From' and 'To' blank. Define the variable group as 'Global variables'.

4. Click 'Create all' to declare the variables then click 'Cancel' to close this window.




Creation of N variables

Enter informations

Name	MyDefVar_%
Type	INT
Group	Global variables
Read-only	☐
Dim.	0
From	1
To	5

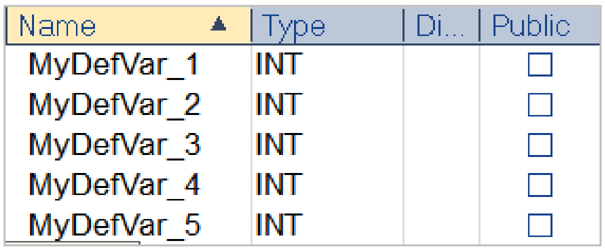
Overview

Creation of 5 variables :

From variable MyDefVar_1 to variable MyDefVar_5

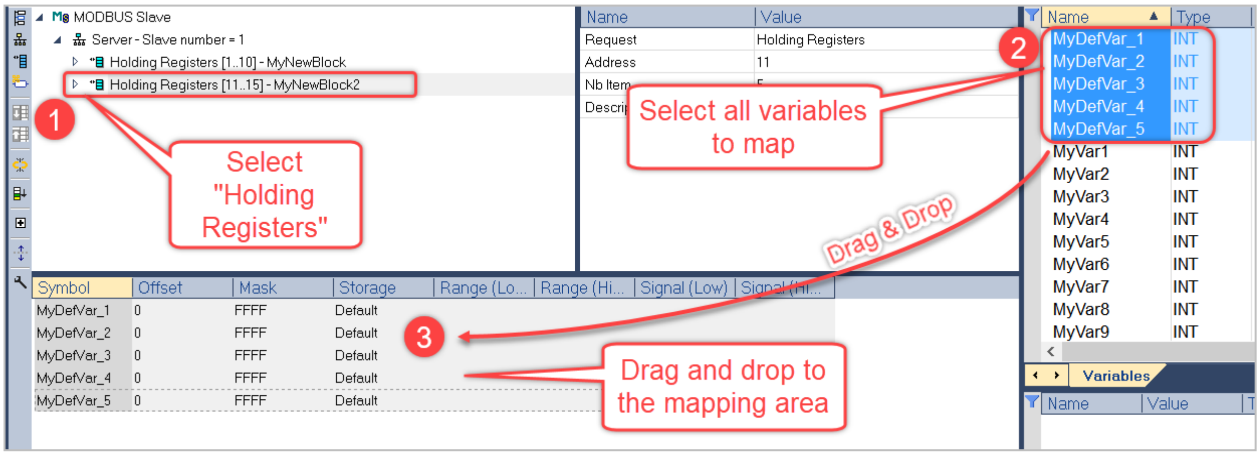
Report

Following figure shows the Workbench adds the declared variables to the variable editor.



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

Now we can map the 5 variables of "MyDefVar" to the Modbus register data block.



MODBUS Slave
Server - Slave number = 1
Holding Registers [1..10] - MyNewBlock
Holding Registers [11..15] - MyNewBlock2
1
Select "Holding Registers"
2
Select all variables to map
Drag & Drop
3
Drag and drop to the mapping area
Variables

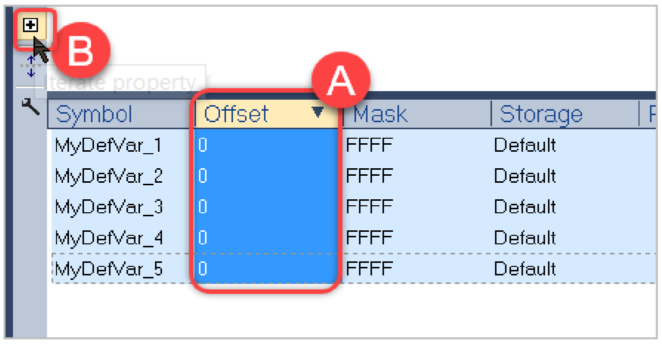
Name	Value
Request	Holding Registers
Address	11
Nb Item	5
Descri	

Name ▲	Type
MyDefVar_1	INT
MyDefVar_2	INT
MyDefVar_3	INT
MyDefVar_4	INT
MyDefVar_5	INT
MyVar1	INT
MyVar2	INT
MyVar3	INT
MyVar4	INT
MyVar5	INT
MyVar6	INT
MyVar7	INT
MyVar8	INT
MyVar9	INT

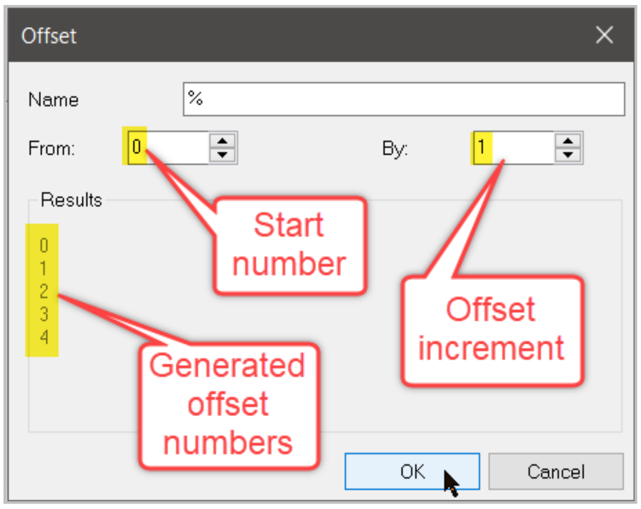
Symbol	Offset	Mask	Storage	Range (Lo...	Range (Hi...	Signal (Low)	Signal (Hi...
MyDefVar_1	0	FFFF	Default				
MyDefVar_2	0	FFFF	Default				
MyDefVar_3	0	FFFF	Default				
MyDefVar_4	0	FFFF	Default				
MyDefVar_5	0	FFFF	Default				

There are 2 ways to change the Offset of variables.

1. Click the head line of the offset columnⒶ to select all variables. Then click the ⊞ icon Ⓑ on the left-side tool bar to open the Offset window. Key in the start number and the increment value. Confirm again the result on the generated offset numbers, and click "OK".



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

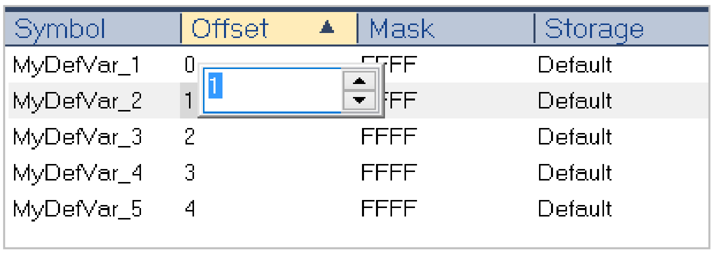


Name: %

From: 0 By: 1

Results
0
1
2
3
4

2. Double-click the offset column to assign an offset position. The first variable start at the offset at 0, the second one at 1, and etc.…

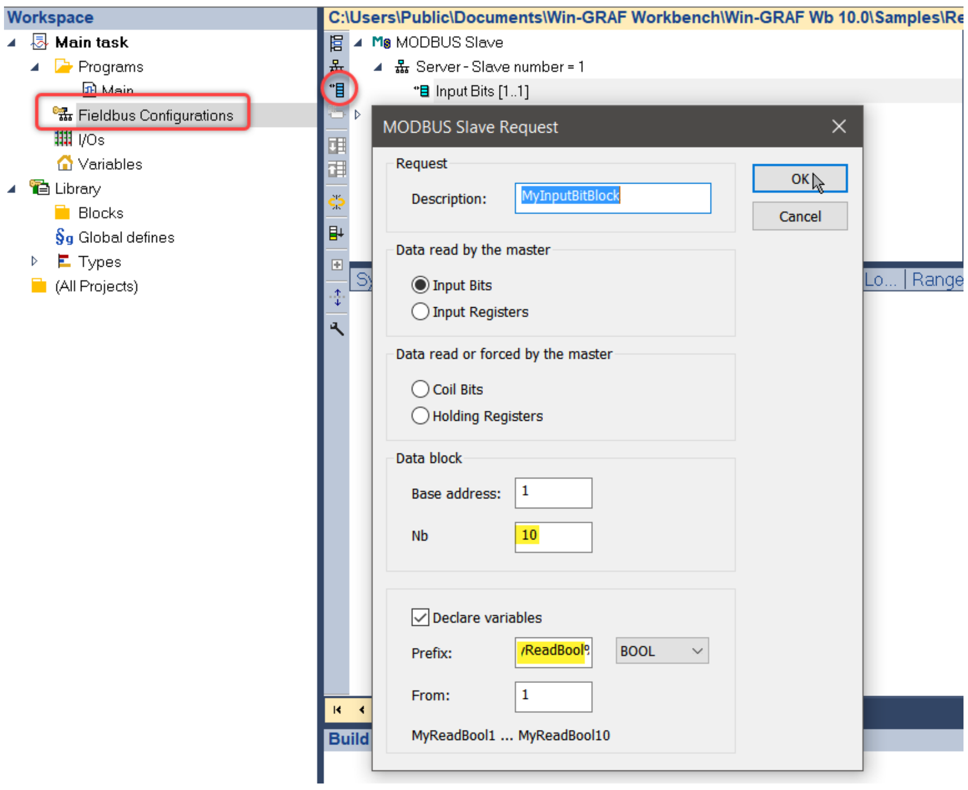


Symbol	Offset	Mask	Storage
MyDefVar_1	0	FFFF	Default
MyDefVar_2	1	FFFF	Default
MyDefVar_3	2	FFFF	Default
MyDefVar_4	3	FFFF	Default
MyDefVar_5	4	FFFF	Default

6.1.4 Define Input Bit Data Block

❖ Following procedure describes how to add an input bit data block which can only be read by the master.

❖



MODBUS Slave Request

Request
Description: MyInputBitBlock

Data read by the master
(● Input Bits) ○ Input Registers

Data read or forced by the master
○ Coil Bits ○ Holding Registers

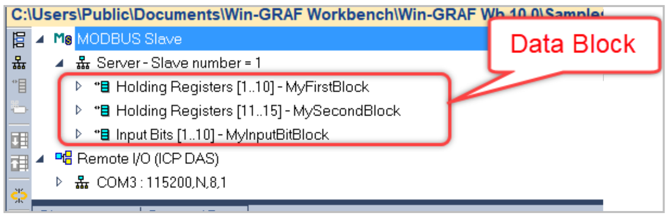
Data block
Base address: 1
Nb: 10

☑ Declare variables
Prefix: MyReadBool% BOOL
From: 1
MyReadBool1 ... MyReadBool10

6.2 Slave Type Configuration

The Modbus data block configuration done in the previous section defines the memory size and structure of the slave. This chapter explains how to create a communication interface through which the Master can exchange data with the slave data block (Figure 6.2.1~6.2.3).

1. Define a Modbus data block which contains sections for
- Input and output register
- Input and output bit

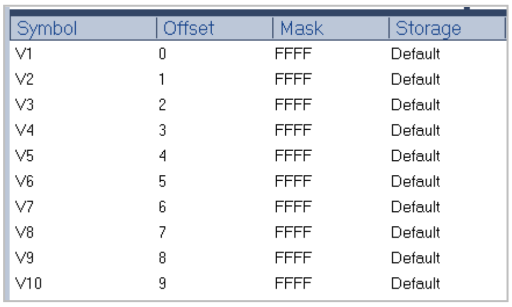


C:\Users\Public\Documents\Win-GRAF Workbench\Win-GRAF Wb 10.0\Sample

- MODBUS Slave
  - Server - Slave number = 1
    - Holding Registers [1..10] - MyFirstBlock
    - Holding Registers [11..15] - MySecondBlock
    - Input Bits [1..10] - MyInputBitBlock
- Remote I/O (ICP DAS)
  - COM3 : 115200,N,8,1

<Figure 6.2.1>

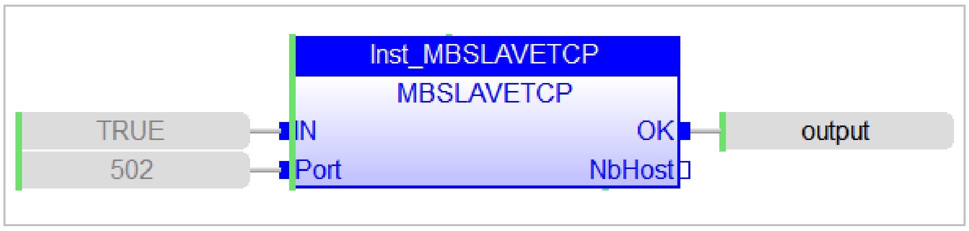
2. Map the data block to global variables



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

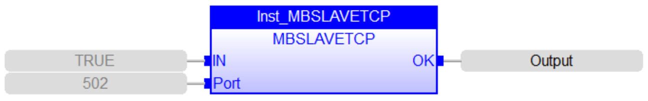
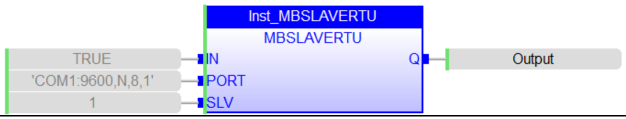
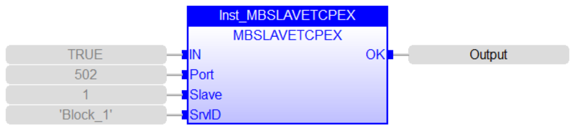
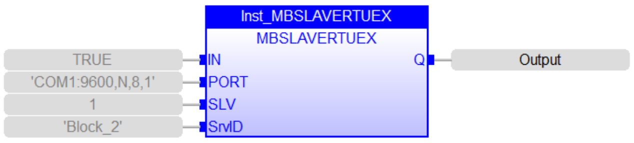
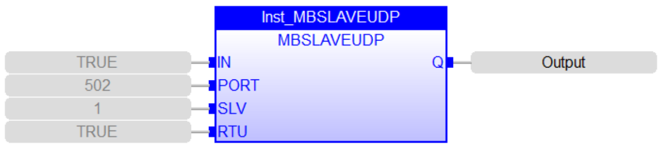
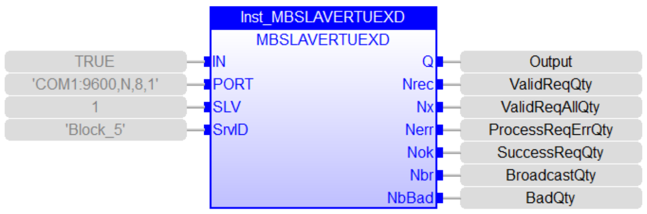
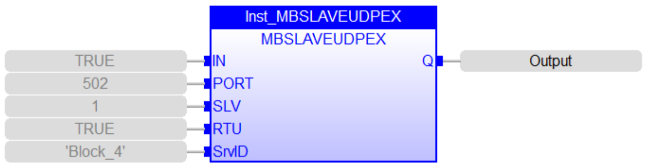
<Figure 6.2.2>

3. Set the Modbus protocol for accessing the data block
- Modbus TCP, Modbus RTU, Modbus UDP



<Figure 6.2.3>

Win-GRAF provides Modbus slave function blocks which handles the communication between master and the slave data block by processing the request received from the master. If the master request the content of the slave data block, then the slave function block reads the data from the data block and write it to the response data frame to the master. Function blocks are provided which supports Ethernet and serial communication (Table 6.2).

Modbus Slave Function Blocks	
Ethernet	Serial (RS-232, RS-485)
MBSLAVETCP 	MBSLAVERTU  Inst_MBSLAVERTU MBSLAVERTU TRUE 'COM1:9600,N,8,1' 1 IN PORT SLV Q Output
MBSLAVETCPEX  Inst_MBSLAVETCPEX MBSLAVETCPEX TRUE 502 1 'Block_1' IN Port Slave SrvID OK Output	MBSLAVERTUEX  Inst_MBSLAVERTUEX MBSLAVERTUEX TRUE 'COM1:9600,N,8,1' 1 'Block_2' IN PORT SLV SrvID Q Output
MBSLAVEUDP  Inst_MBSLAVEUDP MBSLAVEUDP TRUE 502 1 TRUE IN PORT SLV RTU Q Output	MBSLAVERTUEXD  Inst_MBSLAVERTUEXD MBSLAVERTUEXD TRUE 'COM1:9600,N,8,1' 1 'Block_5' IN PORT SLV SrvID Q Nrec Nx Nerr Nok Nbr NbBad Output ValidReqQty ValidReqAllQty ProcessReqErrQty SuccessReqQty BroadcastQty BadQty
MBSLAVEUDPEX  Inst_MBSLAVEUDPEX MBSLAVEUDPEX TRUE 502 1 TRUE 'Block_4' IN PORT SLV RTU SrvID Q Output	

<Table 6.2: Function blocks for setting the Modbus protocol type>

7 Redundancy

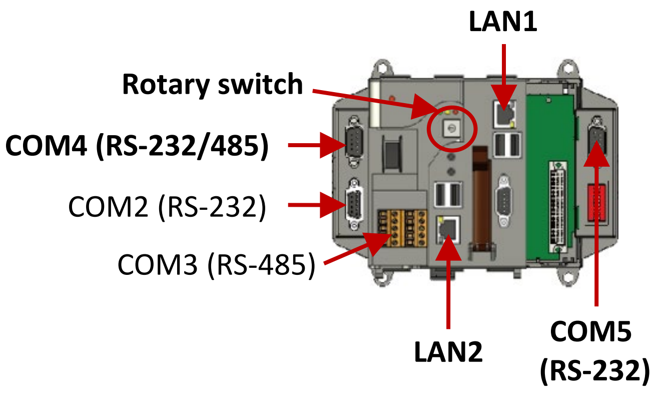
In engineering, redundancy is the duplication of critical components or functions of a system with the intention of increasing reliability of the system, usually in the form of a backup or fail-safe, or to improve actual system performance. Redundancy sometimes produces less, instead of greater reliability – it creates a more complex system which is prone to various issues, it may lead to human neglect of duty, and may lead to higher production demands which by overstressing the system may make it less safe.

ICP DAS **Win-GRAF CE PAC - XP-8xx8-CE6** series support the redundant system:

One redundant system is composed by two Win-GRAF PACs. When the PAC that running programs is crashed or need to release its control authority by user-defined event, the PAC control authority will automatically switch to the other one.

Name Definition	**Redundancy Settings**
Main-PAC	Rotary switch is set to 2
Backup-PAC	Rotary switch is set to 4
Active-PAC	The PAC that is running the program
Passive-PAC	The PAC that is synchronizing data

Note: The Main-PAC is Active-PAC at the beginning and its control authority will automatically be changed to the other PAC depends on conditions.



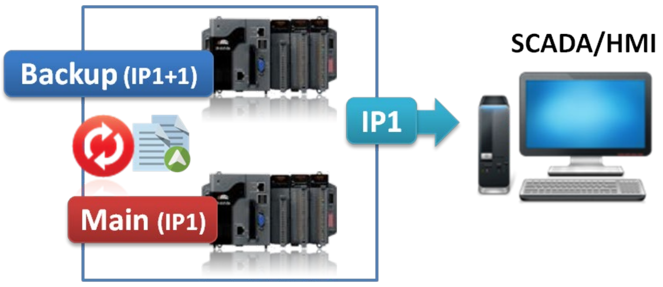
Notice:
1. Do not use two Main-PACs or two backup-PACs to form a redundant system.
2. Two PACs are communicating by using these communication ports.

Comunication Port	**Redundancy Mechanism (Rotary Switch: 2 & 4)**
A. Public IP Port	**LAN1**: used to communicate with Win-GRAF, SCADA, or HMI.
B. Replication Port	**LAN2**: data synchronization port. Two PACs are connected with an Ethernet crossover cable for high-speed data synchronization.
C. Alive Port (or Heart-beat Port)	**COM5 (RS-232)**: Used to detect if two PACs are running properly.

Features of the Win-GRAF redundancy

1. **Higher safety:**
   The redundant system conducts communication through LAN1, LAN2, and Alive-port. Active-PAC can run Win-GRAF project even if only one cable works. If three cables are failed to communicate with the PAC, the other PAC will automatically become an Active-PAC by a switching mechanism.

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 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 the Win-GRAF project if there are other projects such as C, VB.net, C# and eLogger HMI, running in the redundant system, these files need to pre-installed to the spare XPAC (or a repaired PAC) before re-installing this PAC to the redundant system.

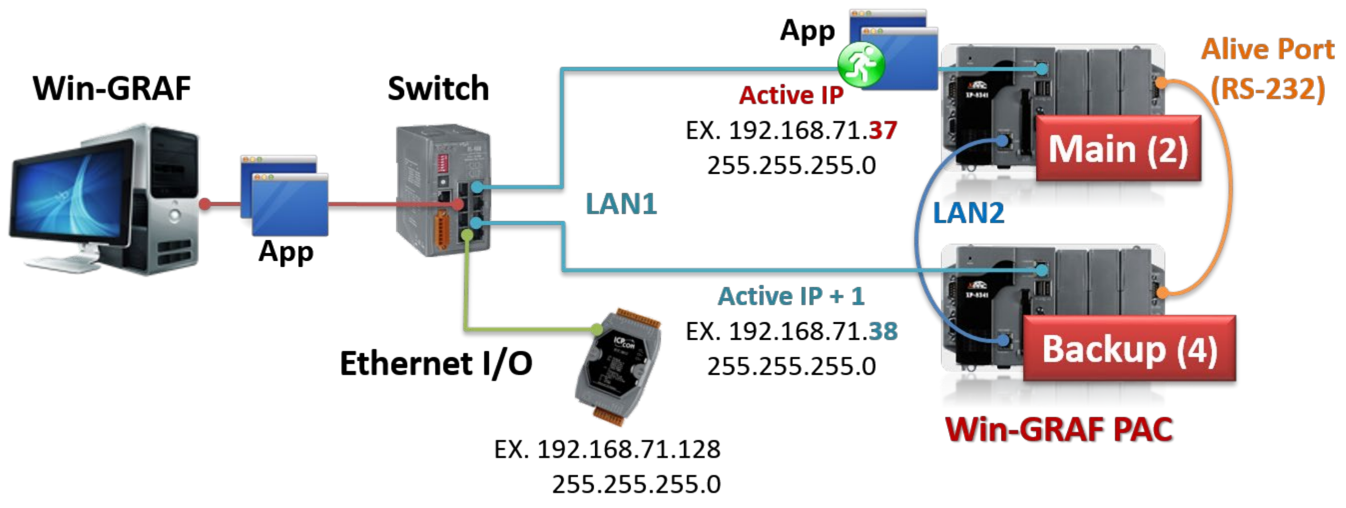
4. **Simplifying the programming process:**
   Users do not need to specify what files or data should be sent to spare PAC because the Win-GRAF redundant system will automatically handle these tasks.

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 iDCS-8000 system to achieve I/O redundancy.

7.1 Hardware Connection

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.

Hardware		
Two XPAC	Rotary Switch	One is set to "**2**" (called Main-PAC) One is set to "**4**" (called Backup-PAC)
	LAN1	2 x Ethernet cable
	LAN2	1 x Ethernet crossover cable
	COM5 (RS-232)	1 x RS-232 crossover cable

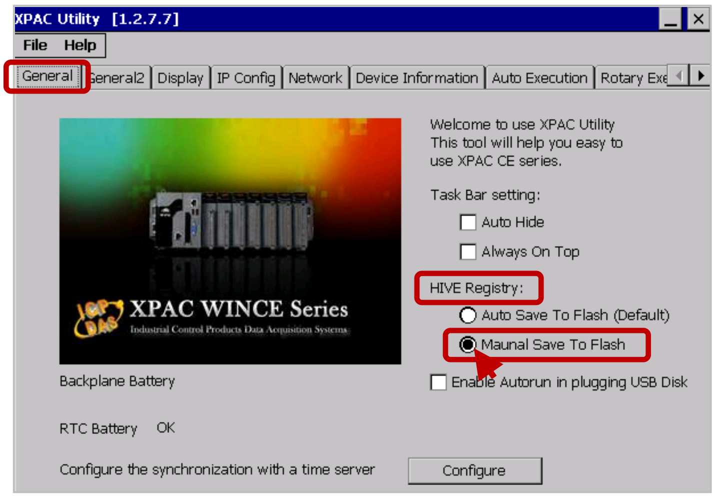
7.2 Software Settings

7.2.1 XPAC Utility

Set IP addresses for LAN1 and LAN2:

☒ The factory default setting of LAN1 and LAN2 is DHCP that must be set as static IP by using XPAC_Utility.

Set the rotary switch to "0" for redundant system.
Excute XPAC Utility and select the Manual Save To Flash option under HIVE Registery on the General page.



XPAC Utility [1.2.7.7]

File Help

General | General2 | Display | IP Config | Network | Device Information | Auto Execution | Rotary Exe

Welcome to use XPAC Utility
This tool will help you easy to use XPAC CE series.

Task Bar setting:
☐ Auto Hide
☐ Always On Top

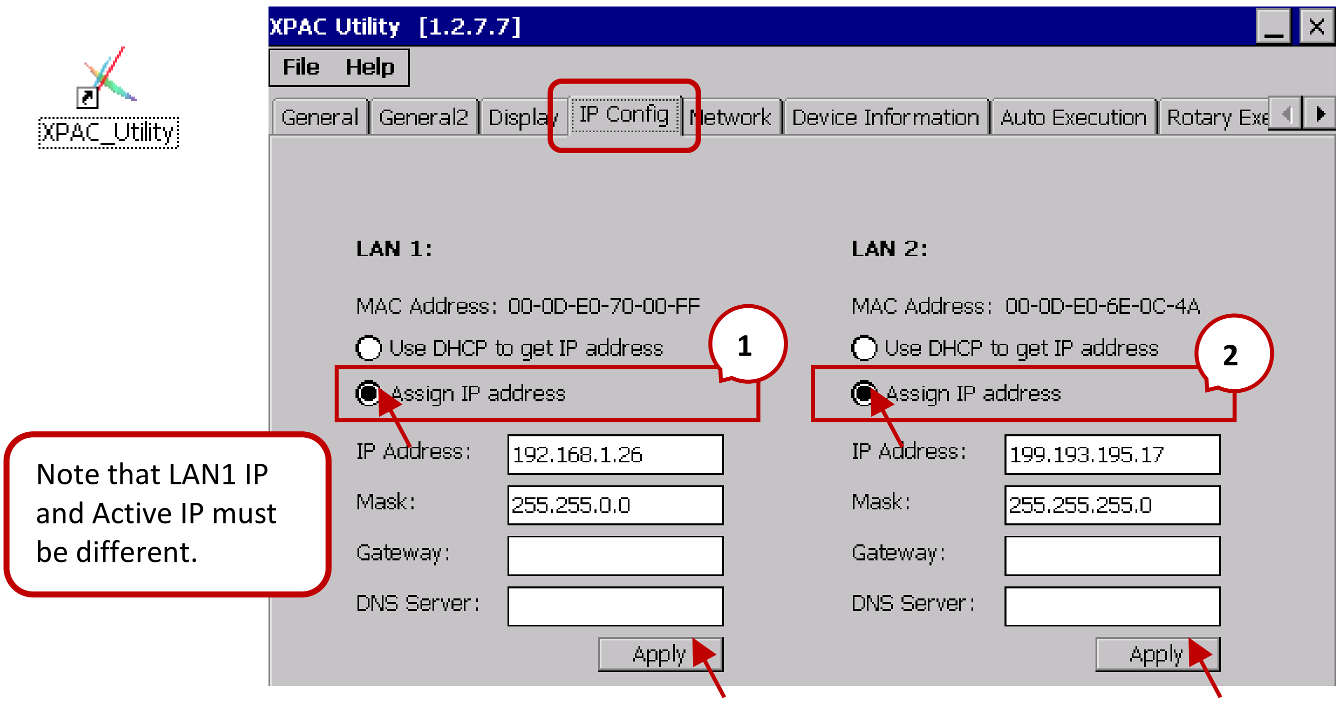
HIVE Registry:
○ Auto Save To Flash (Default)
● Maunal Save To Flash
☐ Enable Autorun in plugging USB Disk

Backplane Battery

RTC Battery OK

Configure the synchronization with a time server [Configure]

Click on IP Config tab and configure LAN1 IP for redundant system. Note that,
1) Set LAN1 IP and Active_IP to be different to avoid IP address conflicts after rebooting PACs.
2) For Win-GRAF and the PAC can communicate properly, both the IP addresses of them must on the same network segment.



XPAC_Utility

XPAC Utility [1.2.7.7]

File Help

General | General2 | Display | IP Config | Network | Device Information | Auto Execution | Rotary Exe

LAN 1:		LAN 2:	
MAC Address: 00-0D-E0-70-00-FF		MAC Address: 00-0D-E0-6E-0C-4A	
○ Use DHCP to get IP address		○ Use DHCP to get IP address	
● Assign IP address		● Assign IP address	
IP Address:	192.168.1.26	IP Address:	199.193.195.17
Mask:	255.255.0.0	Mask:	255.255.255.0
Gateway:		Gateway:	
DNS Server:		DNS Server:	
	Apply		Apply

Note that LAN1 IP and Active IP must be different.

Check "Assign IP address" under the LAN2 setting for redundant system.
Click both two "Apply" buttons to save the settings.

Set up communication cables and redundant mode (Rotary switch: 2 & 4)

A. Conncet 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 COM5 ports of PACs must connect to a RS-232 crossover cable.
D. Set the rotary switch of PAC to "2" (called Main-PAC) and the other one to "4" (called Backup-PAC).
E. Rebooting two PACs.

After rebooting, the LAN2 IP address and Mask address will automatically be set as follows:

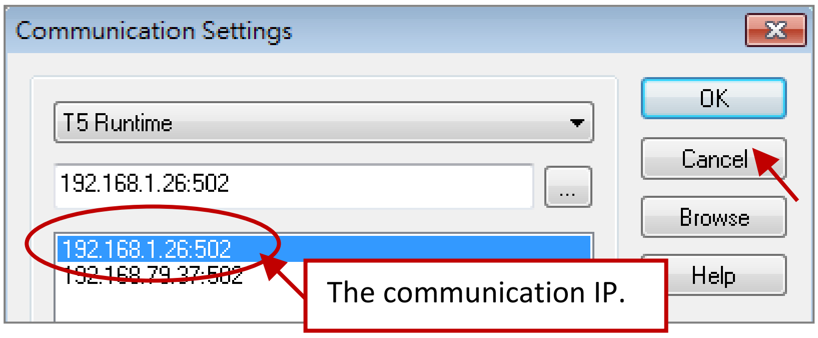
Main-PAC (2) : 199.193.195.**17** / 255.255.255.0 ;

Backup-PAC (4) : 199.193.195.**9** / 255.255.255.0

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 (2).

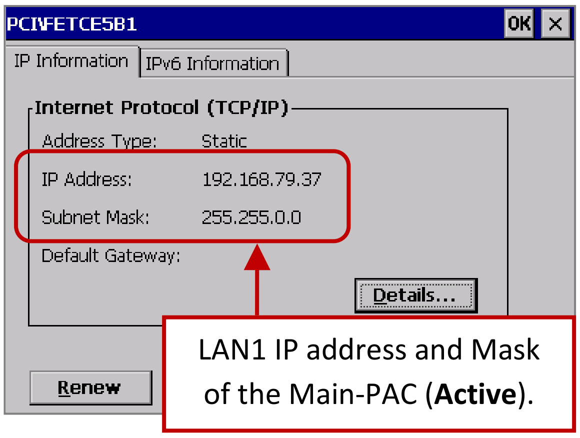

Communication Settings
T5 Runtime
192.168.1.26:502
192.168.1.26:502
192.168.79.37:502

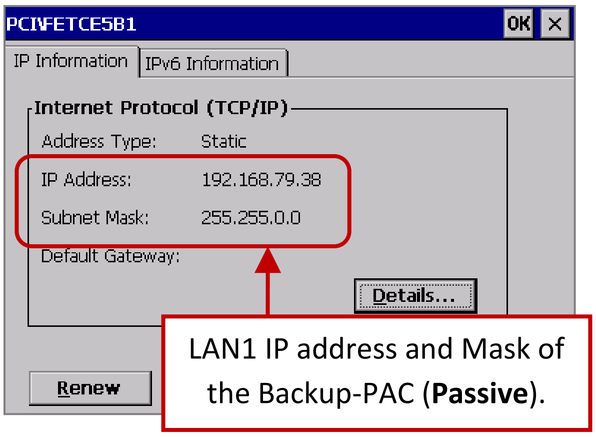
2. Set the Active_IP address

Set the Active_IP and Mask address for the " i_redundancy" function depends on the network environment.

3. Download the Win-GRAF project

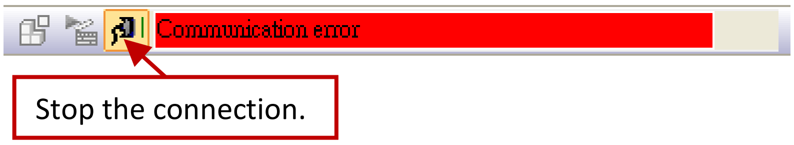
Click the "On Line" button ( ) to connect, and download the redundant project to the Main-PAC. After it's downloaded, LAN1 IP address of the Main-PAC(2) will automatically set to the Active_IP address and LAN1 IP address of the Backup-PAC(4) will set to Active IP + 1.


PCIVFETCE5B1
Internet Protocol (TCP/IP)
Address Type: Static
IP Address: 192.168.79.37
Subnet Mask: 255.255.0.0
Default Gateway:

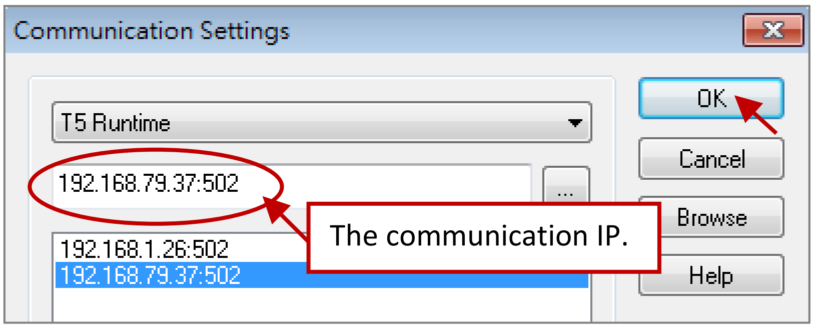

PCIVFETCE5B1
Internet Protocol (TCP/IP)
Address Type: Static
IP Address: 192.168.79.38
Subnet Mask: 255.255.0.0
Default Gateway:

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.

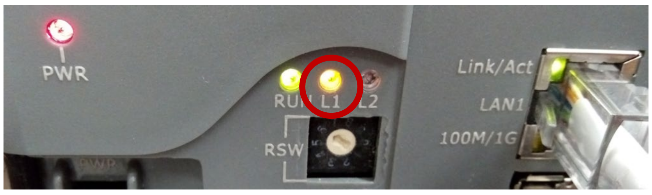


Next, change the Win-GRAF communication IP to the Active_IP address so that this project will always download to the Active-PAC whenever the user wants to debug or change it.



Note: Since Win-GRAF Driver v1.10, users can know which one is Active-PAC by checking the L1 LED indicator on PAC.

L1 LED indicator	
ON	This PAC is Active PAC
Blinking	This PAC is Passive PAC



7.3 Switchover Mechanism

In this chapter we will introduce the mechanism of how XPACs have be back ups for each other. Following figure shows the hardware cables:

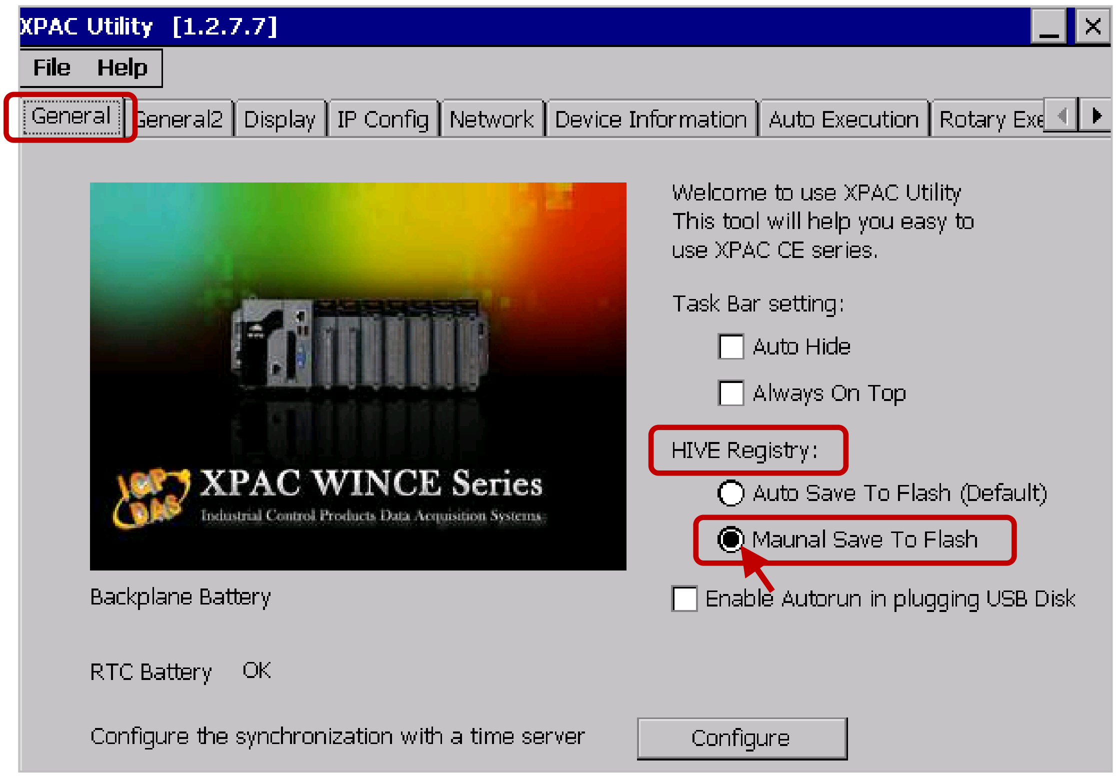
Hardware		
Two XPAC	Rotary Switch	One is set to "**2**" (called Main-PAC) One is set to "**4**" (called Backup-PAC)
	LAN1	2 x Ethernet cable
	LAN2	2 x Ethernet cable
	COM5 (RS-232)	1 x RS-232 crossover cable

7.3.1 Main PAC and Backup PAC Switchover

The factory default setting of LAN1 and LAN2 is DHCP that must be set as static IP by using XPAC_Utility.

A. Set the rotary switch to “0” for redundant system.

B. Excute XPAC Utility and select the Manual Save To Flash option under HIVE Registery on the General page.



Welcome to use XPAC Utility
This tool will help you easy to use XPAC CE series.

Task Bar setting:
☐ Auto Hide
☐ Always On Top

HIVE Registry:
○ Auto Save To Flash (Default)
● Maunal Save To Flash

☐ Enable Autorun in plugging USB Disk

Backplane Battery

RTC Battery OK

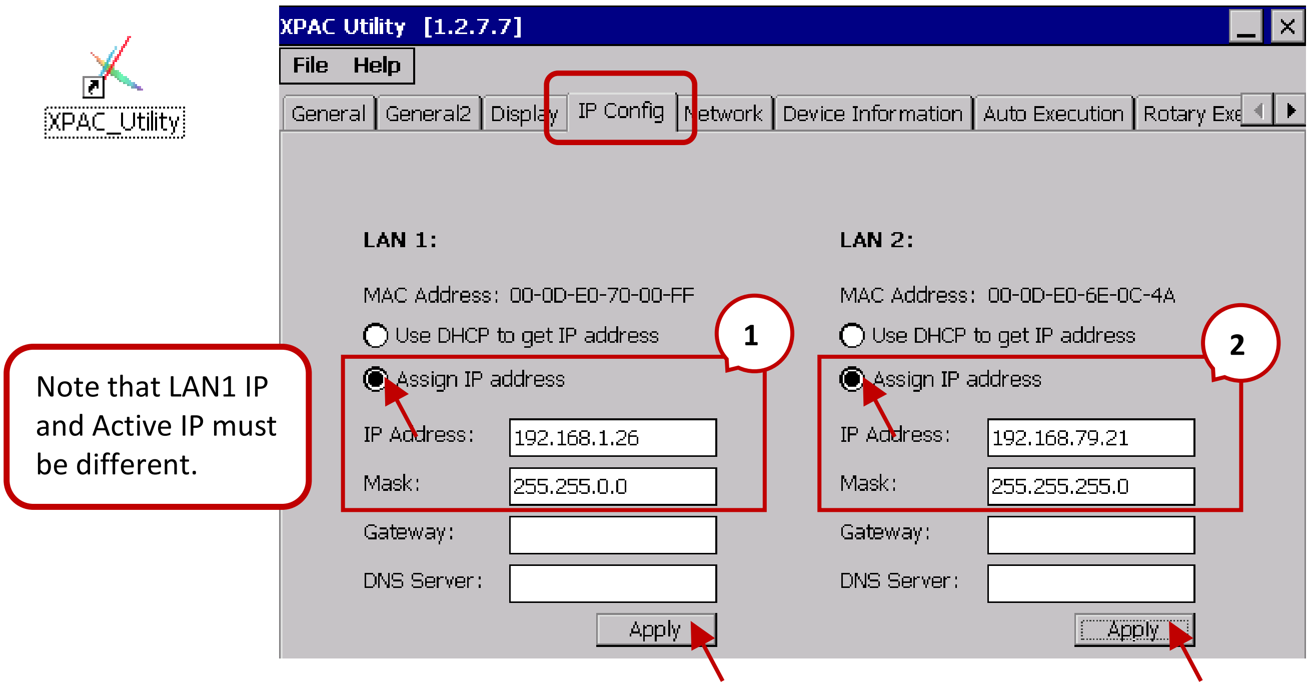
Configure the synchronization with a time server

C. Click on IP Config tab and configure LAN1 IP for redundant system.

Note:

1) Set LAN1 IP and Active_IP to be different to avoid IP address conflicts after rebooting PACs.

2) For Win-GRAF and the PAC can communicate properly, both the IP addresses of them must on the same network segment.


XPAC_Utility
XPAC Utility [1.2.7.7]
File Help
General General2 Display IP Config Network Device Information Auto Execution Rotary Exe
1
2
Apply
Apply

Note that LAN1 IP and Active IP must be different.

LAN 1:

MAC Address: 00-0D-E0-70-00-FF

Use DHCP to get IP address [✓ Assign IP address]

IP Address:	192.168.1.26
Mask:	255.255.0.0
Gateway:	
DNS Server:	

LAN 2:

MAC Address: 00-0D-E0-6E-0C-4A

Use DHCP to get IP address [✓ Assign IP address]

IP Address:	192.168.79.21
Mask:	255.255.255.0
Gateway:	
DNS Server:	

D. Configure the LAN2 settings for two PACs.

Note:

When using far-field redundant system, user needs to configure LAN2 IP for each PAC.

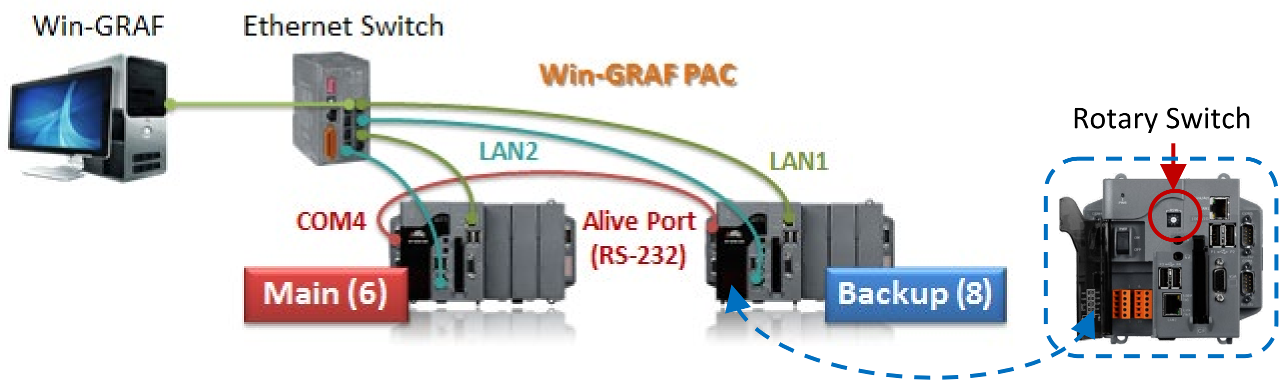
1) If the LAN2 IP of Main-PAC is x.x.x.**n**, the LAN2 IP of Backup-PAC must be x.x.x.**n+1**.

2) The first three octet of LAN2 IP for two PACs must be the same, and the Mask address for LAN2 and LAN1 must be different.

Rotary Switch	**LAN2**	**Description**
Main PAC (2)	**IP** (E.g., 192.168.79.**21**) **Mask** (E.g., 255.255.255.0)	Download Win-GRAF project to this PAC. **It is Active PAC by default** (L1 LED is ON).
Backup PAC (4)	**IP+1** (E.g., 192.168.79.**22**) **Mask** (E.g., 255.255.255.0)	**It is Passive PAC by default** (L1 LED is OFF)

E. Click both two "Apply" buttons to save the settings.

7.3.2 Normal Working Sync. Condition



Step 1: Turn on two PAC with normal mode (i.e., Rotary Switch = 0 & 0) and set IP addresses for LAN1 and LAN2.

Note: To avoid IP address conflicts after rebooting PACs, LAN1 and Win-GRAF Active_IP must be set to a different IP address.

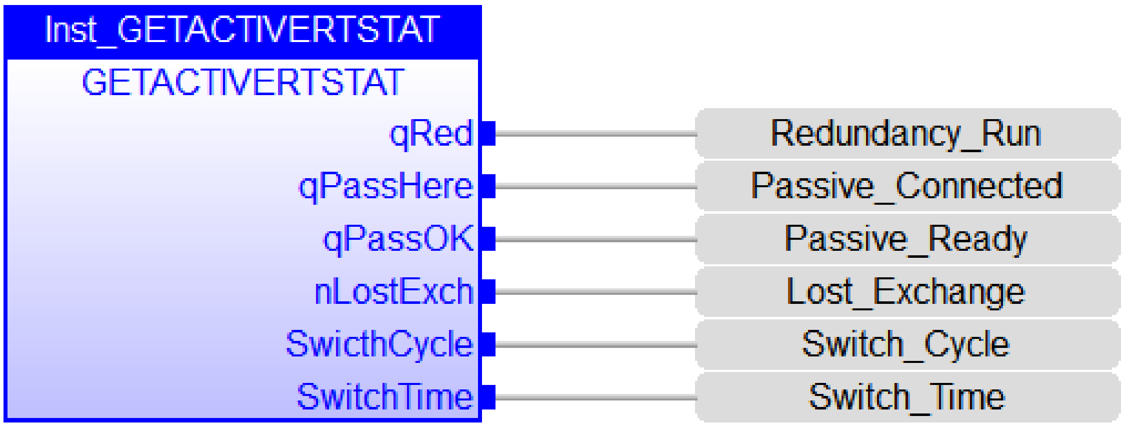
Step 2: Wiring three communication cables for each PAC and power on with redundant mode (i.e., Rotary Switch = 2 & 4).

Step 3: Modify both communication IP and Active IP in the Win-GRAF project, and download it to Main PAC (2). After that, the project will automatically transfer to Backup PAC (4) via LAN2.

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.

7.3.3 Concept of Syncronyzation

1. Switch_Time: Last synchronized. To switch Time Range. Not the time from Active to Passive.
2. Switch_Cycle: First cycle after switch
3. Lost Exchange: Always synchronize Data.



7.4 Function Blocks

7.4.1 GetActiveRTStat

GetActiveRTStat		
This function shows the status of active runtime		
NAME	TYPE	DESCRIPTION
Output		
qRed	BOOL	This parameter is TRUE if redundancy is possible. It is FALSE if the runtime does not support redundancy.
qPassHere	BOOL	This parameter indicates if a passive runtime is connected to the running active runtime.
qPassOK	BOOL	This parameter indicates if the connected passive runtime has finished its initialization operations and is now synchronized for data replication.
nLostExch	DINT	This parameter indicates the number of replication exchanges failed on timeout.
SwitchCycle	BOOL	This parameter is TRUE if the running runtime has become active on a switch and is now running its first cycle.
SwitchTime	TIME	This parameter gives an indication of the last switch duration.

7.4.2 GetActiveExStat

GetActiveExStat		
This function shows the extra status of active runtime		
NAME	TYPE	DESCRIPTION
Output		

PAC_Mode	INT	This parameter indicates PAC mode.
		0: Standard PAC;
		1: Redundancy Main PAC;
		2: Redundancy Backup PAC

Sync_LAN	INT	This parameter indicates current synchronization LAN port. 1: LAN1; 2: LAN2.
LAN1_State	BOOL	This parameter is TRUE if LAN1 is connected. It is FALSE if LAN1 is disconnected.
LAN2_State	BOOL	This parameter is TRUE if LAN2 is connected. It is FALSE if LAN2 is disconnected.

7.4.3 ActiveRTSwitch

ActiveRTSwitch		
This function is used to switch active PAC and passive PAC		
NAME	TYPE	DESCRIPTION
Input		
bEnable	BOOL	Set this parameter as TRUE to enable active and passive PAC switch.
Output		
bOK	BOOL	This parameter is TRUE if switch successfully.

7.4.4 ActiveSYNSwitch

ActiveSYNSwitch		
This function is used to switch synchronization LAN port		
NAME	TYPE	DESCRIPTION
Input		
bEnable	BOOL	Set this parameter as TRUE to enable synchronization switch.
Output		
bOK	BOOL	This parameter is TRUE if switch successfully.

8 OPC UA Server

OPC Unified Architecture (UA) is an open standard created by the OPC Foundation and defines a platform independent interoperability standard. OPC UA offers a secure method of client-to-server connectivity and has the ability to connect securely through firewalls and over VPN connections.
For the majority of user applications, the most relevant components of the UA standard are as follows:

- Secure connections through trusted certificates for client and server endpoints.
- Robust item subscription model to provide efficient data updates between clients and servers.
- An enhanced method of discovering available information from participating UA servers.

The purpose of this manual is to introduce the main functions and configuration supported by the Win-GRAF OPC UA server. In addition configuration and testing procedure are given to familiarize yourselves with the features, functions, limitations and operating characteristics of specific settings.

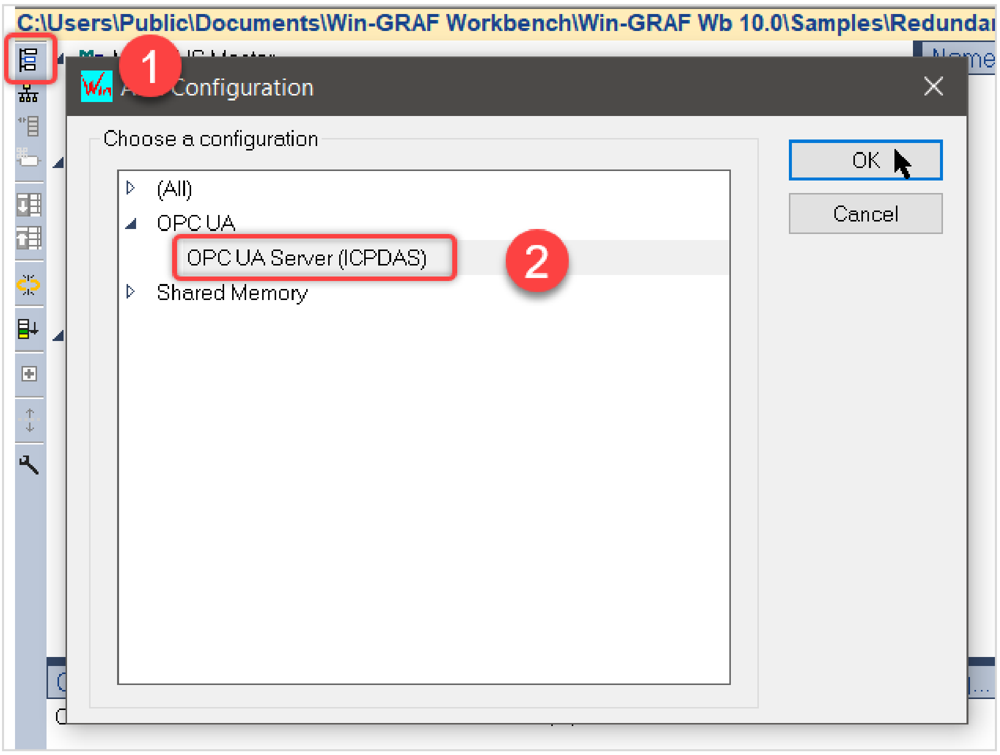
8.1 OPC UA Server Configuration

You should be familiar with the OPC UA specification and the communication methods used for the data exchange between OPC UA servers and clients. If security plays a vital role in your application a deeper understanding of how OPC UA certificate are used by the servers and clients to securely identify and communicate with each other is being required.
This chapter gives a quick overview of the OPC UA server configuration procedure using the Win-GRAF workbench and describes the supported configuration parameters.

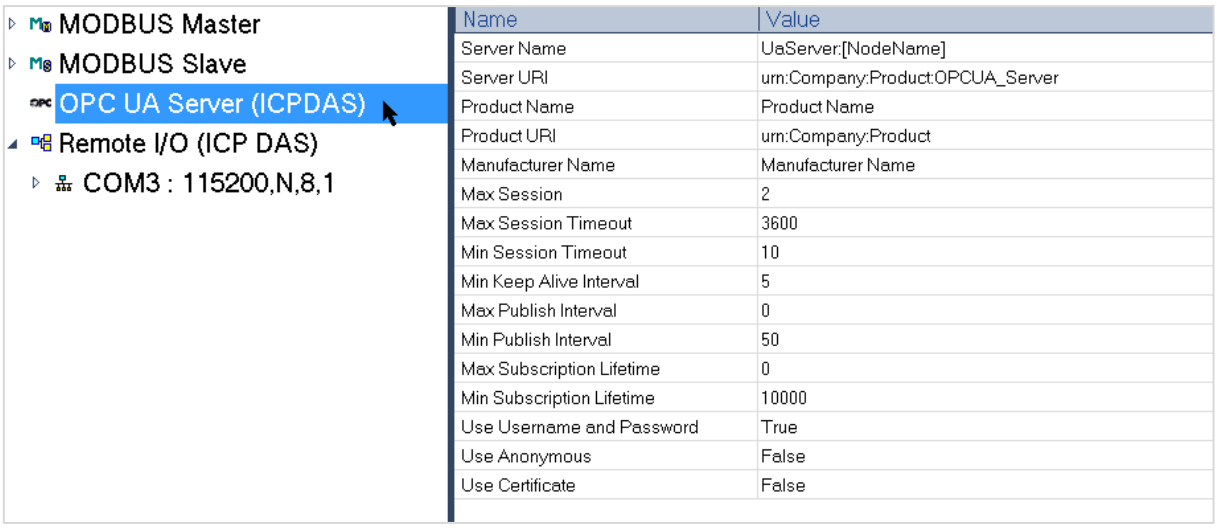
8.1.1 Server Configuration Procedure

This section provides a quick overview how to add and configure a OPC UA server with the Win-GRAF workbench. The next chapter describes the server parameters in more details.

- **Step 1**: Start the Win-GRAF workbench and create a new project
- **Step 2**: Open the Fieldbus Configurations window by clicking on the 'Fieldbus Configuration' button in the toolbar or double clicking the 'Fieldbus Configuration' node in the workspace.
- **Step 3**: Open the 'OPC UA Server' plug-in: Click 'Insert Configuration' button on the left toolbar and select 'OPC UA Server (ICPDAS)' from the 'Add Configuration' dialog.

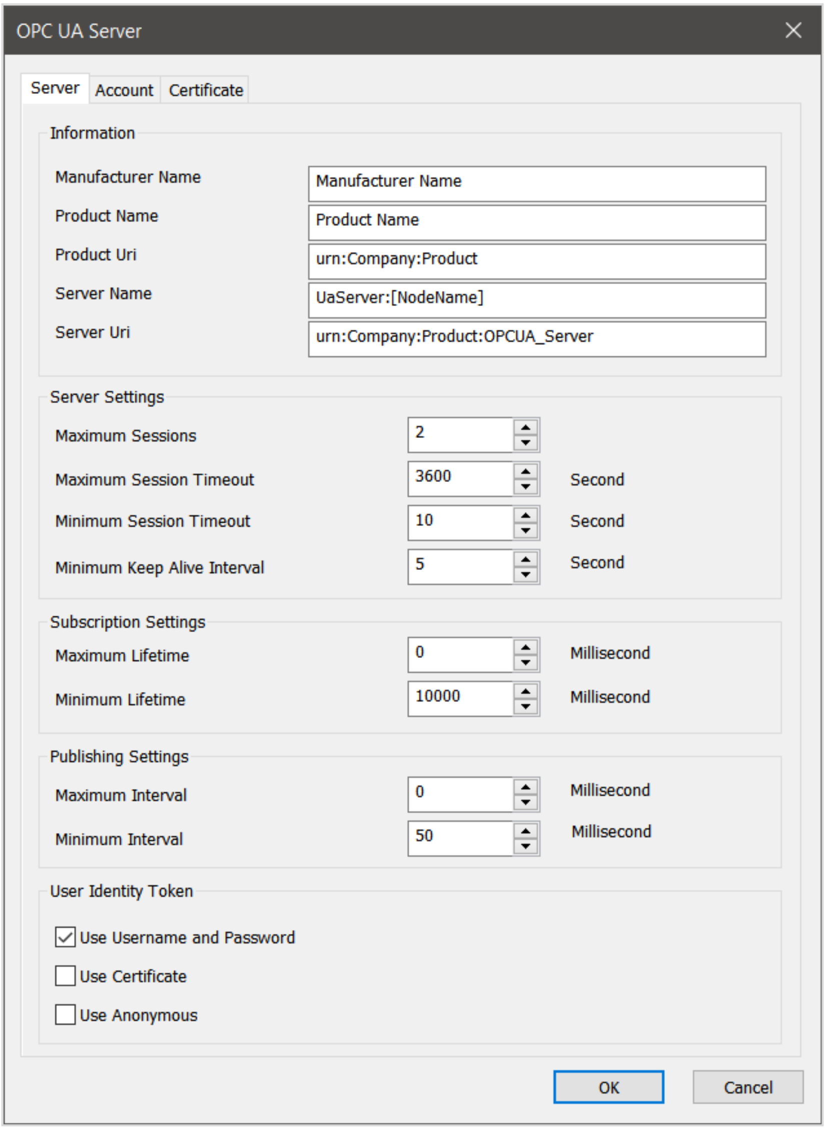


- **Step 4**: Basic configuration parameters setting:
  - Click on the "OPC UA Server (ICPDAS)" node to show a list of basic configuration parameters with their default values. Double click a parameter entry in the 'Value' column to modify the setting.



Name	Value
Server Name	UaServer:[NodeName]
Server URI	urn:Company:Product:OPCUA_Server
Product Name	Product Name
Product URI	urn:Company:Product
Manufacturer Name	Manufacturer Name
Max Session	2
Max Session Timeout	3600
Min Session Timeout	10
Min Keep Alive Interval	5
Max Publish Interval	0
Min Publish Interval	50
Max Subscription Lifetime	0
Min Subscription Lifetime	10000
Use Username and Password	True
Use Anonymous	False
Use Certificate	False

-Double click the 'OPC UA Server (ICPDAS)' node (left hand side) to open a separate dialog which beside the basic parameter allows you to do additional setting (e.g. user identity token, user login account).



Information

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

Server Settings

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

Subscription Settings

Maximum Lifetime	0	Millisecond
Minimum Lifetime	10000	Millisecond

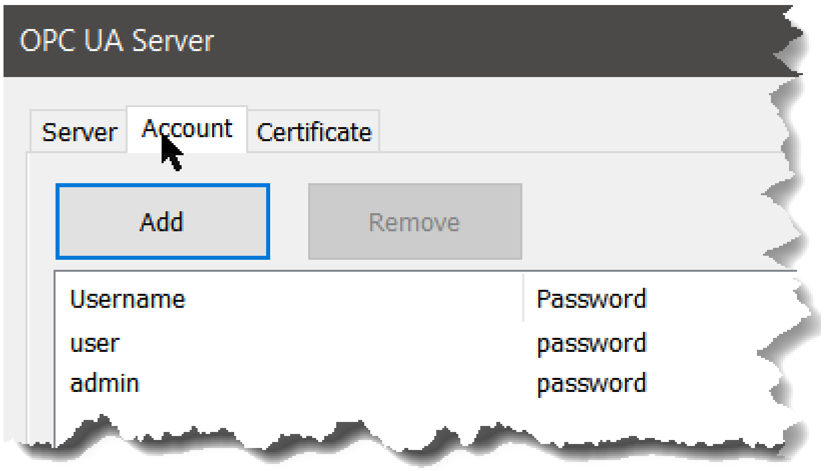
Publishing Settings

Maximum Interval	0	Millisecond
Minimum Interval	50	Millisecond

User Identity Token

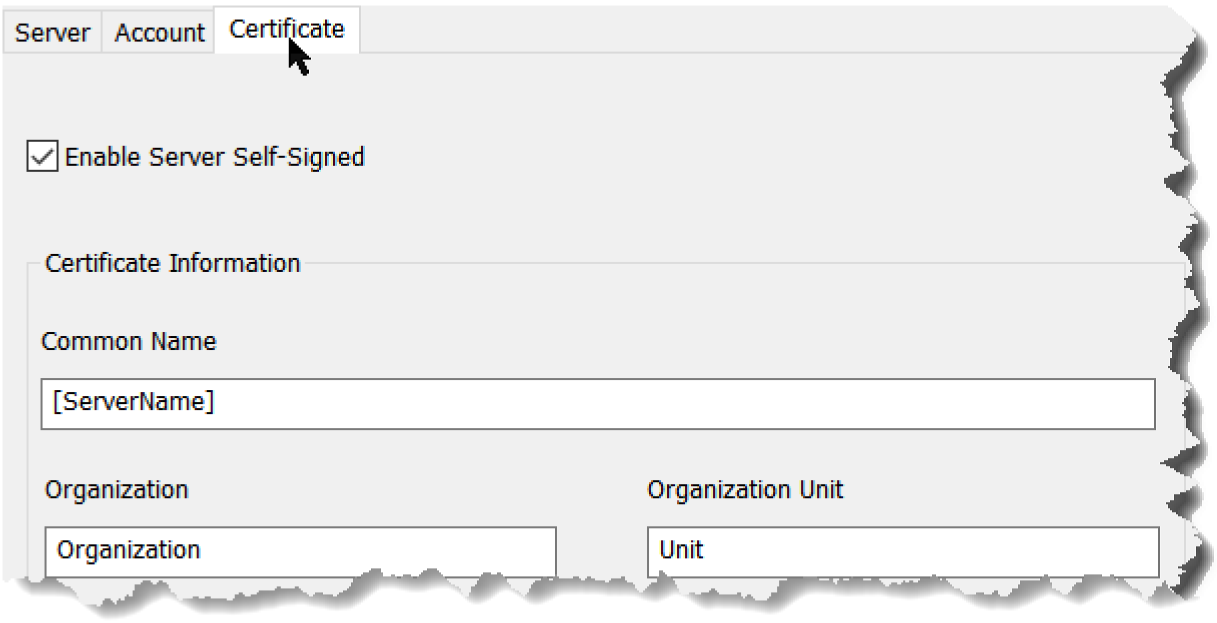
- [✓] Use Username and Password
- [ ] Use Certificate
- [ ] Use Anonymous

- If 'Use Username and Password' is selected, add the usernames with their passwords to the account list in the 'Account' tab.



Username	Password
user	password
admin	password

- Under the 'Certificate' tab make sure that the 'Enable Server Self-Signed' option is enabled and the other detailed as shown below a filled in. The server will use this information to generate a certificate with its private key and store it in the directories 'pkiserver\own\certs\' and 'pkiserver\own\private\'. The 'Enable Server Self-Signed' option only needs to be selected if no server certificate with its key already exist in the directories, because the server can not start without having a certificate.



☑ Enable Server Self-Signed

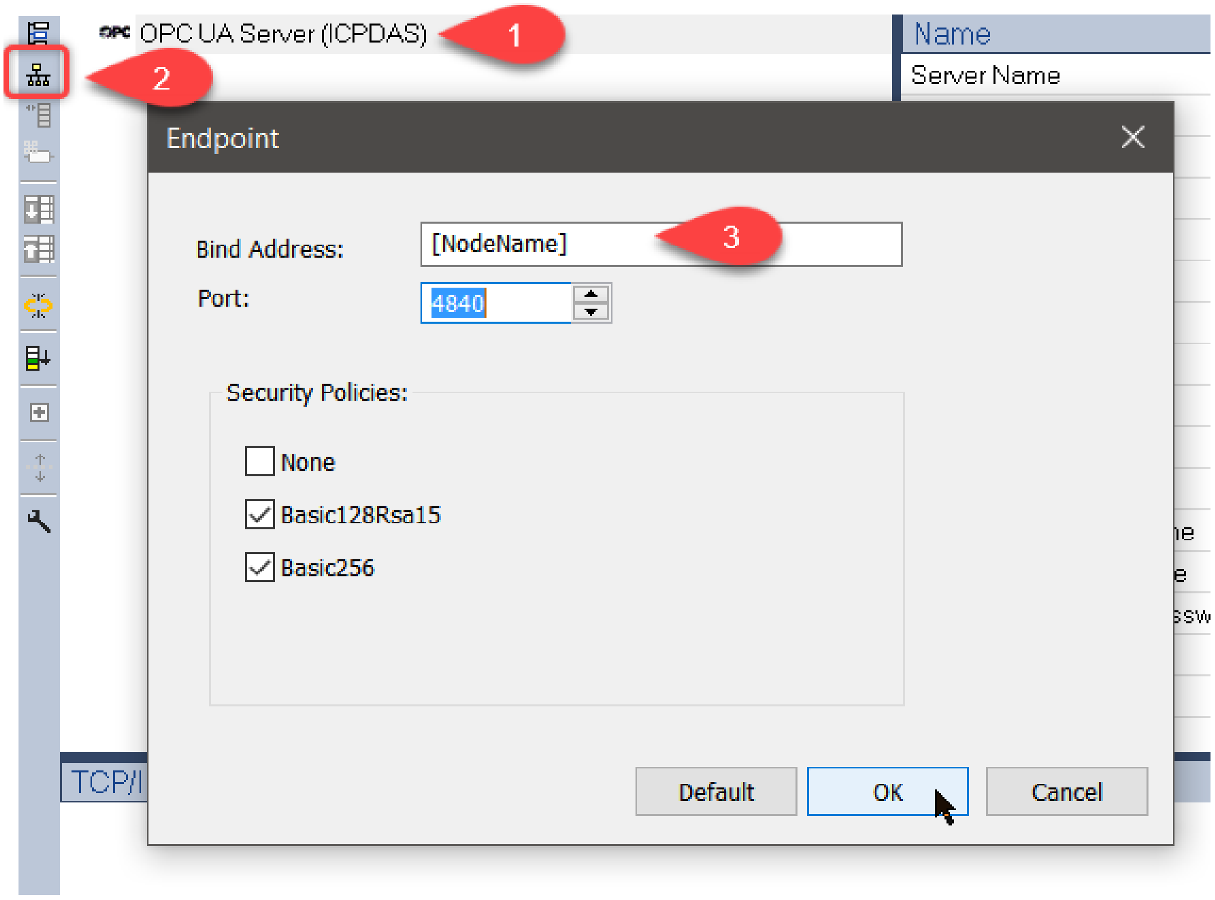
Certificate Information

Common Name: [ServerName]

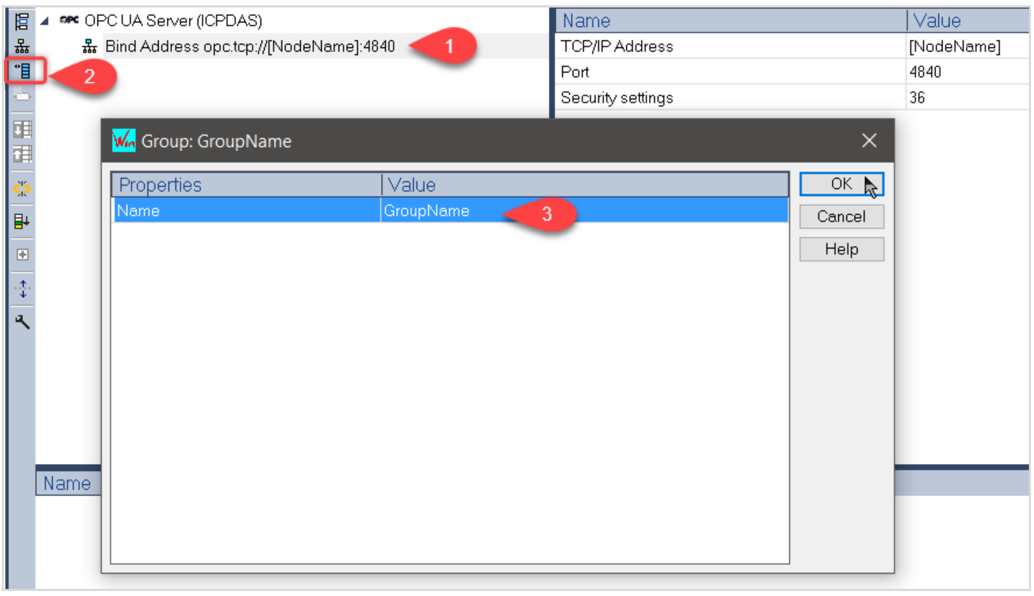
Organization: Organization

Organization Unit: Unit

- **Step 5**: Endpoint setting:
  - Add an endpoint node to the server:
  1. Activate the 'OPC UA Server (ICPDAS)' node by clicking on the node
  2. Click on the 'Insert Master/Port' button on the left toolbar. A 'Endpoint' dialog pops up.
  3. Set the endpoint parameters. Only one endpoint is supported by the Win-GRAF server.
  Confirm the setting with 'OK'.



- **Step 6**: Create a monitoring group node:
  1. Activate endpoint node (*'Bind IP Address: opc.tcp://...'*)
  2. Click 'Insert Slave/Data Block' command on the left toolbar. A 'Group' dialog will be displayed
  3. Enter a group name in the dialog by double clicking the 'Value' column. Click 'OK'
  The new node with the group name will added to the tree view.
-

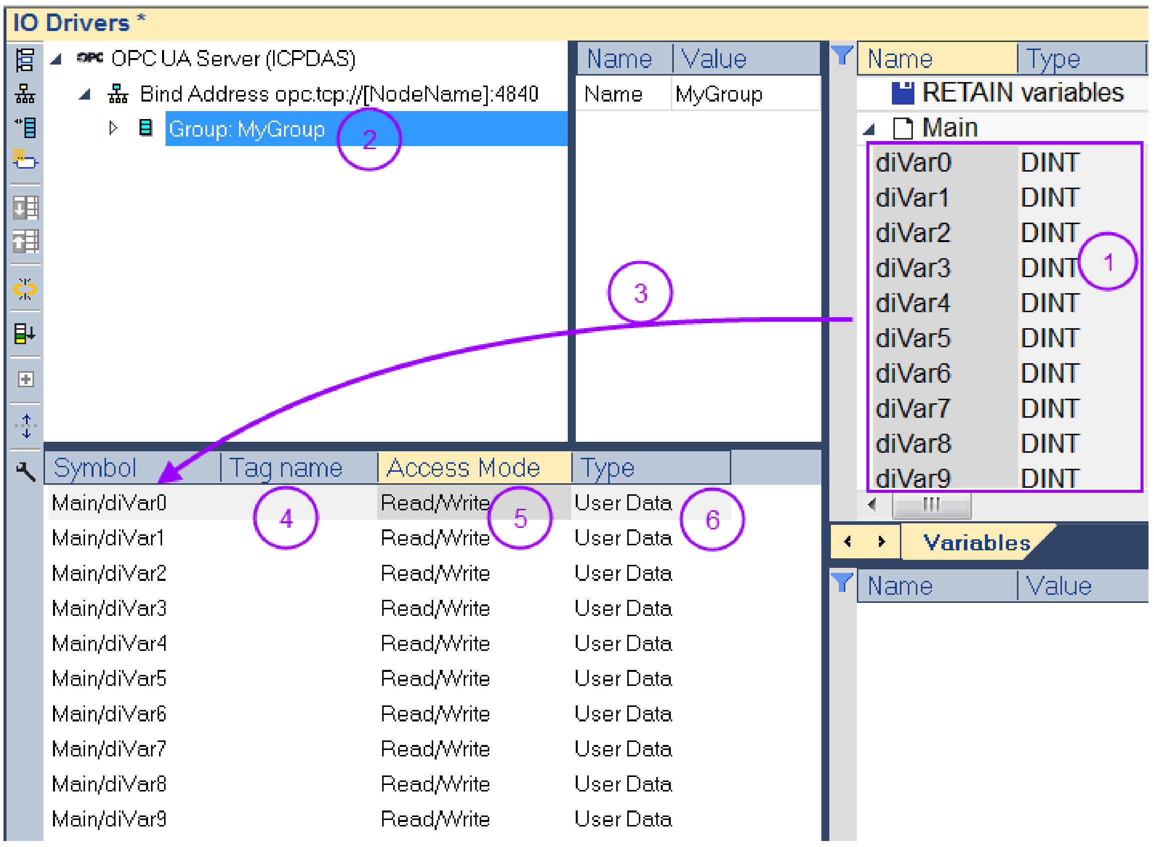

OPC UA Server (ICPDAS)
Bind Address opc.tcp://[NodeName]:4840

Name	Value
TCP/IP Address	[NodeName]
Port	4840
Security settings	36

Group: GroupName

Properties	Value
Name	GroupName

- **Step 7**: Assign PLC variable to the group. Here the PLC variables are assigned to the UA server for the client to access.
  1. Declare variables to be accessed by the client
  2. Select the group node by clicking on it
  3. Drag and drop the variables to the mapping area
  4. Enter a 'Tag name' for each variable. The tag name will be shown to the client. If the tag name is empty, then the 'Symbol' name will use for the node name.
  5. Select the client access mode: 'Read only', 'Write only' or 'Read/Write', 'No Access'. If 'No Access' has been selected, then the variable represents the OPC UA server status. The type of status to be shown has to be set in the 'Type' column.
  6. In the 'Type' column the OPC UA server status type represented by the variable is set. The 'Type' column is only valid if the access mode 'No Access' has been selected. Available server status type:
     - - Server Status
     - - Used Sessions
     - - Used Subscription
     - - Used Monitored Items



Name	Value
Name	MyGroup

Name	Type
diVar0	DINT
diVar1	DINT
diVar2	DINT
diVar3	DINT
diVar4	DINT
diVar5	DINT
diVar6	DINT
diVar7	DINT
diVar8	DINT
diVar9	DINT

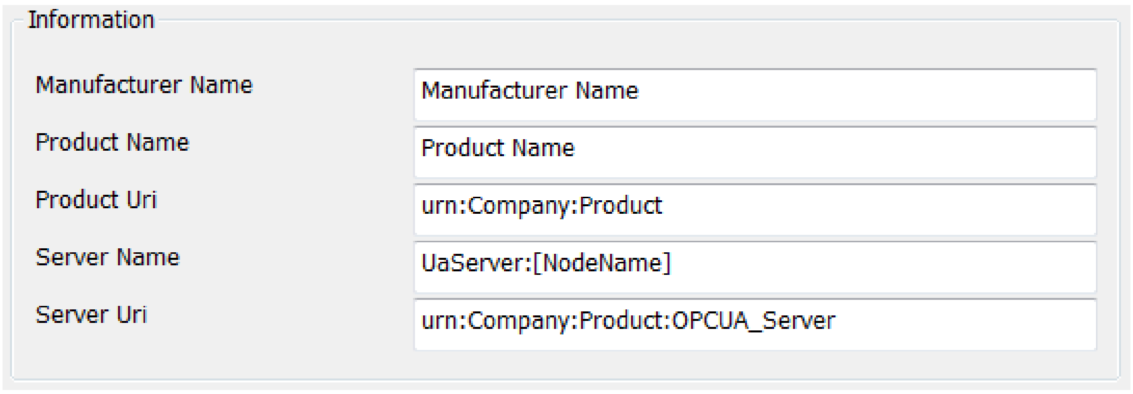
Symbol	Tag name	Access Mode	Type
Main/diVar0		Read/Write	User Data
Main/diVar1		Read/Write	User Data
Main/diVar2		Read/Write	User Data
Main/diVar3		Read/Write	User Data
Main/diVar4		Read/Write	User Data
Main/diVar5		Read/Write	User Data
Main/diVar6		Read/Write	User Data
Main/diVar7		Read/Write	User Data
Main/diVar8		Read/Write	User Data
Main/diVar9		Read/Write	User Data

- **Step 8**: Built the program, download it to the runtime and start the application.
  If no certificate exist in the folder *'.\pkiserver\own\certs'* of the runtime execution file directory then a new certificate with its private key will be automatically created.
- **Step 9**: Use a OPC UA client to connect to the server.

8.2 Server Configuration Parameters

The communication settings for the OPC UA Server module determine how the server will appear on the network, as well as how OPC UA clients may communicate with it.

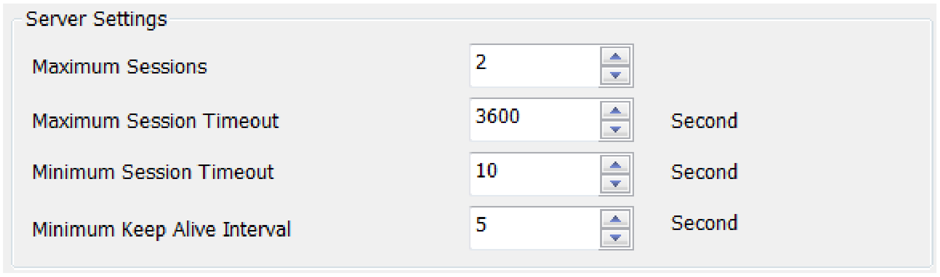
8.2.1 Server Information



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

Server Setting	Description
Manufacturer Name	• The name of the manufacturer. This name will be used for the PLC application program build information and is shown in the trace log file.
Product Name	• The name of this application. Same as the manufacturer name it is being used as internal data to identify the build program and the trace log file. • Appears as a header in the trace log file.
Product Uri	• A globally unique identifier for the product the server belongs to. • Default name: 'urn:Company:Product' • The following the key strings are supported: - '[NodeName]' - gets the host name of the actual computer. - '[ServerName]' - replaces the key string with the server name • example: - 'urn:MyCompany:MyProduct' - 'urn:[NodeName]:MyProduct'
Server Name	• The name of the server. The default common name is 'Server'. • This name will be stored under the keyword '[ServerName]' which means strings containing the keyword are modified by the server by replacing the keyword with the server name. For example by default the keyword '[ServerName]' is assigned to the 'Common Name' of the self-signed certificate, which means the server name will be used as a 'Common Name'. • Use the key string '[NodeName]' to automatically gets the host name of the actual computer. - For example: If the computer name is 'DESKTOP123' then the string 'MyName@[NodeName]' will generate the common name 'MyName@DESKTOP123'
Server Uri	• Every Server shall have a globally unique identifier called the ServerUri. • If a UA server is selected, the server URI will be displayed. The server URI can be used as a reference to filter the discovery result. • The default name is 'urn:Company:Product:OPCUA_Server' • The following the key strings are supported: - '[NodeName]' - gets the host name of the actual computer. - '[ServerName]' - replaces the key string with the server name • example: - 'urn:MyCompany:MyProduct:MyServer' - 'urn:[NodeName]:MyProduct:[ServerName]'

8.2.2 Session Settings



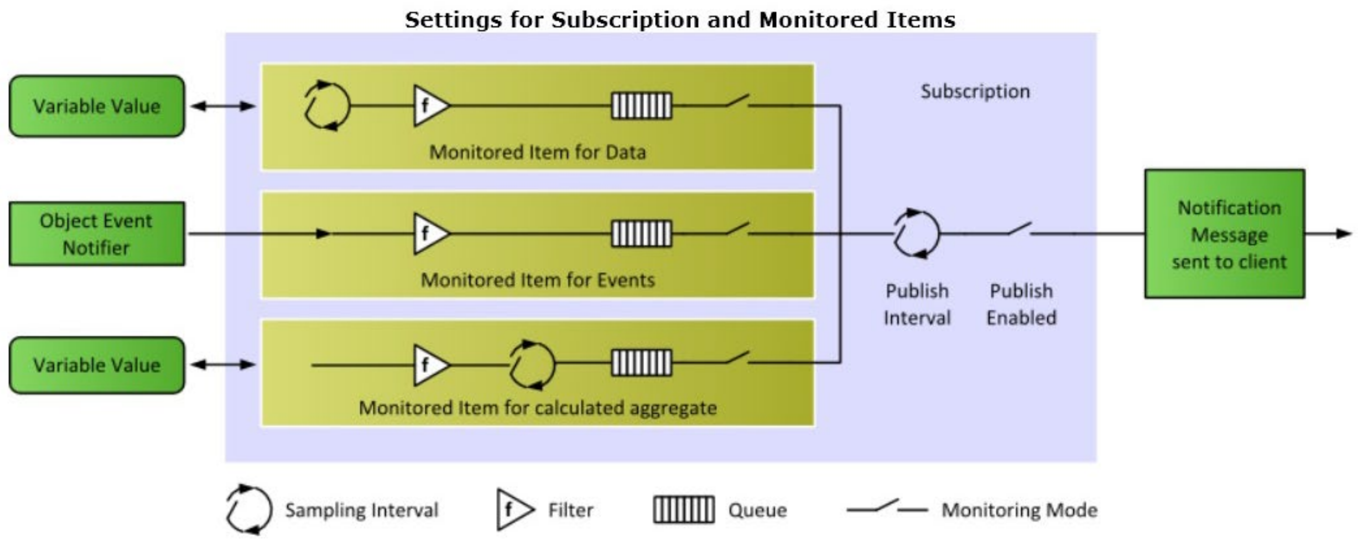
Server Settings

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

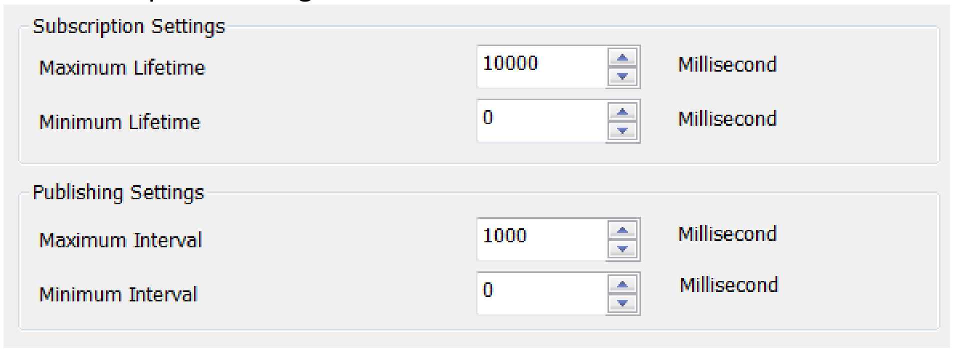
Server Setting	Description
Maximum Sessions	• This parameter specifies the limit of supported connections. The valid range is 1 to 128. The default setting is 2.
Session Timeout	• This parameter specifies the UA client's timeout limit for establishing a session. Values may be changed depending on the needs of the application. The default values are 10 to 3600. • **Minimum**: This parameter specifies the UA client's minimum timeout limit. The default setting is 10 seconds. • **Maximum**: This parameter specifies the UA client's maximum timeout limit. The default setting is 3600 seconds.
Keep Alive Interval	• Subscriptions have a keep-alive counter that counts the time in which there have been no Notifications to report to the Client. When the Keep Alive timeout is reached, a Keep Alive Message informs the Client that the Subscription is still active. • In case that there is nothing to report (e.g. no values have changed) the server will send a Keep Alive notification to the Client, which is an empty Publish, to indicate that the server is still alive.

8.2.3 Server Subscription Setting

OPC UA supports polling and subscription mode.
- In polling mode the client is continuously updating the objects at a defined interval. This creates a higher communication loading and is therefore only recommended for a few symbols.
- In subscription mode the server monitors a selected number of nodes. Only when the nodes value changes will the server notify the client about the changes. This mechanism reduces the amount of transferred data immensely and is therefore the recommended mode for reading information from a OPC UA server.



In the following the subscription setting is described.


Subscription Settings

Maximum Lifetime	10000	Millisecond
Minimum Lifetime	0	Millisecond

Publishing Settings

Maximum Interval	1000	Millisecond
Minimum Interval	0	Millisecond

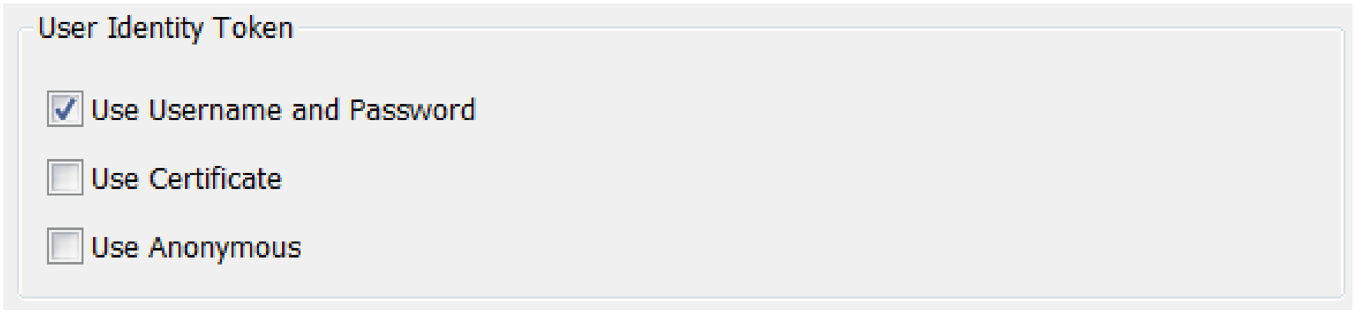
Server Setting	Description
Lifetime	• Interval, in which server publish data to Client • The server will modify the 'Minimum Interval' interval to protect the source from extensive load that may be caused by a high sampling rate. • **Maximum**: The maximum Subscription lifetime (in milliseconds) the server allows. (Default: 10000) • **Minimum**: The minimum subscription lifetime (in milliseconds) the server allows; 0 is no limitation. (Default: 0)
Interval	Interval for sampling (and storing) data at server and send in each publishing interval. The sampling interval defines is the time interval at which the server checks the Monitored Item for data changes. The interval can be set to a faster than the notification interval (Lifetime) to the Client, in which case the server may queue the sampled data and publish the complete queue.

8.3 User Authentication Configuration

When a user is trying to connect from an OPC UA Client to an OPC UA Server, the OPC UA server needs to confirm the identity of the user before allowing the connection from the OPC UA client. The Win-GRAF server currently supports three different ways to authenticate a user during the activation of a session:

- Anonymous Identity Token
- User Name Identity Token
- X509 Identity Token

The '*User Identity Token*' area determines the identity types of the client that will be allowed to log on to the OPC UA server.


User Identity Token

[✓] Use Username and Password

[ ] Use Certificate

[ ] Use Anonymous

Identity Token Option	Description
Use Username and Password	User Name Identity Token • Require clients to enter a username and password that a match a user in your project's security system. This option is available only if the security system has been enabled. • If this option has been selected it is necessary to enter the username and password. The 'Account' tab (Figure 6) provides a list in which the user

	with the password can be entered. • Default: Enabled
Use Certificate	X509 Identity Token • Instead of using a username and password to login into the server a separate X509 certificate is being used for each user. • The common name of the X509 certificate acts as the user name and its private key functions as a password. • Default: Disabled
Use Anonymous	• Allow clients to log on to the server without entering a username or password. • Default: Disabled

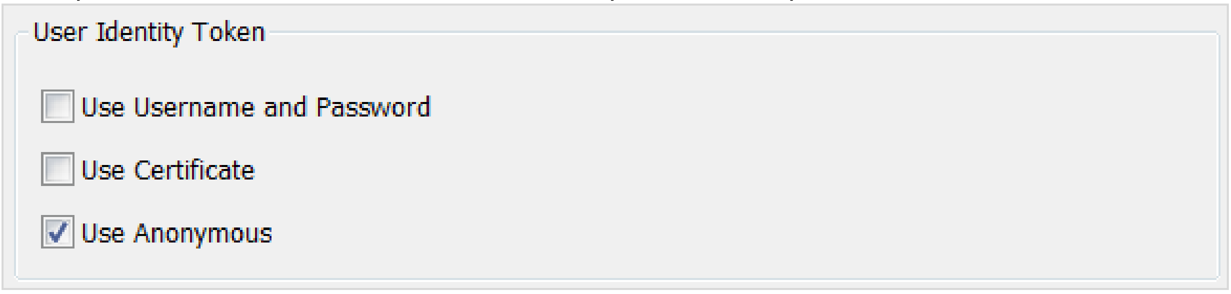
Note:
- At least one of the available options have to be set. If none of the identity options have been selected then the client will not be able to connect to the server.
- The options are not exclusive; more than one option can be selected .

In the following the three application based security configuration is explained.

8.3.1 Anonymous Identity Token

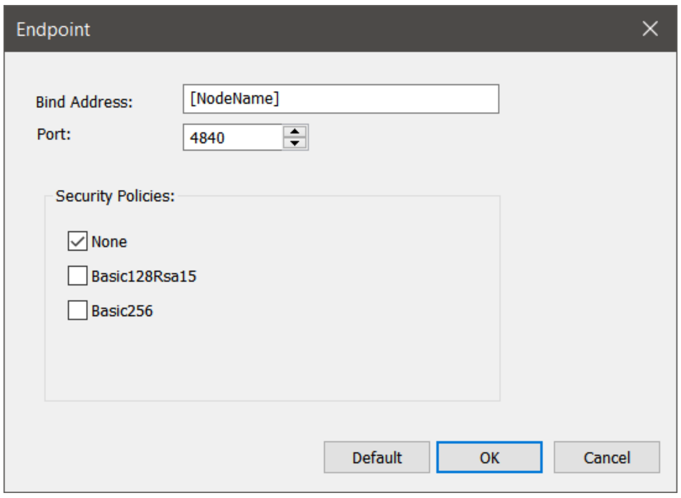
The anonymous identity token does not perform any authentication (Figure 8.3.1.A). That means
- Every client which certificate has been added to the '*pkiserver/trusted/certs*' directory can establish a connection with the server without having to enter a user name and password.
- If the '*Security Policy*' of the server is set to '*None*' (Figure 8.3.1.B) then all clients regardless whether they are trusted by the server or not can create a session with the server.

For security reasons it is recommended to use anonymous identity token.


User Identity Token

- [ ] Use Username and Password
- [ ] Use Certificate
- [✓] Use Anonymous

Figure 8.3.1.A


Endpoint

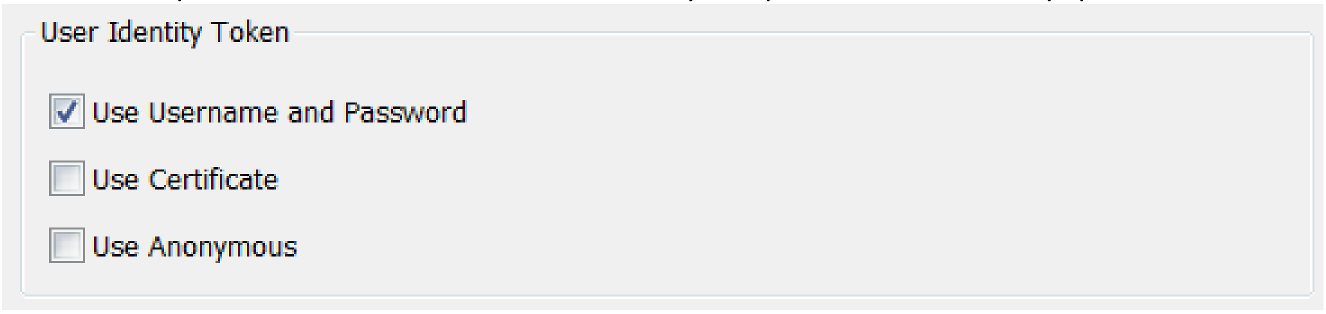
Bind Address: [NodeName]
Port: 4840

Security Policies:
- [✓] None
- [ ] Basic128Rsa15
- [ ] Basic256

Figure 8.3.1.B

8.3.2 User Name Identity Token

This token type configuration (Figure 8.3.2.A) allows to authenticate the user using a username and password. The password is passed to the UA Server in encrypted form, even when using a none secure channel. The password is never stored in the memory as a plain text so it is always protected.


User Identity Token

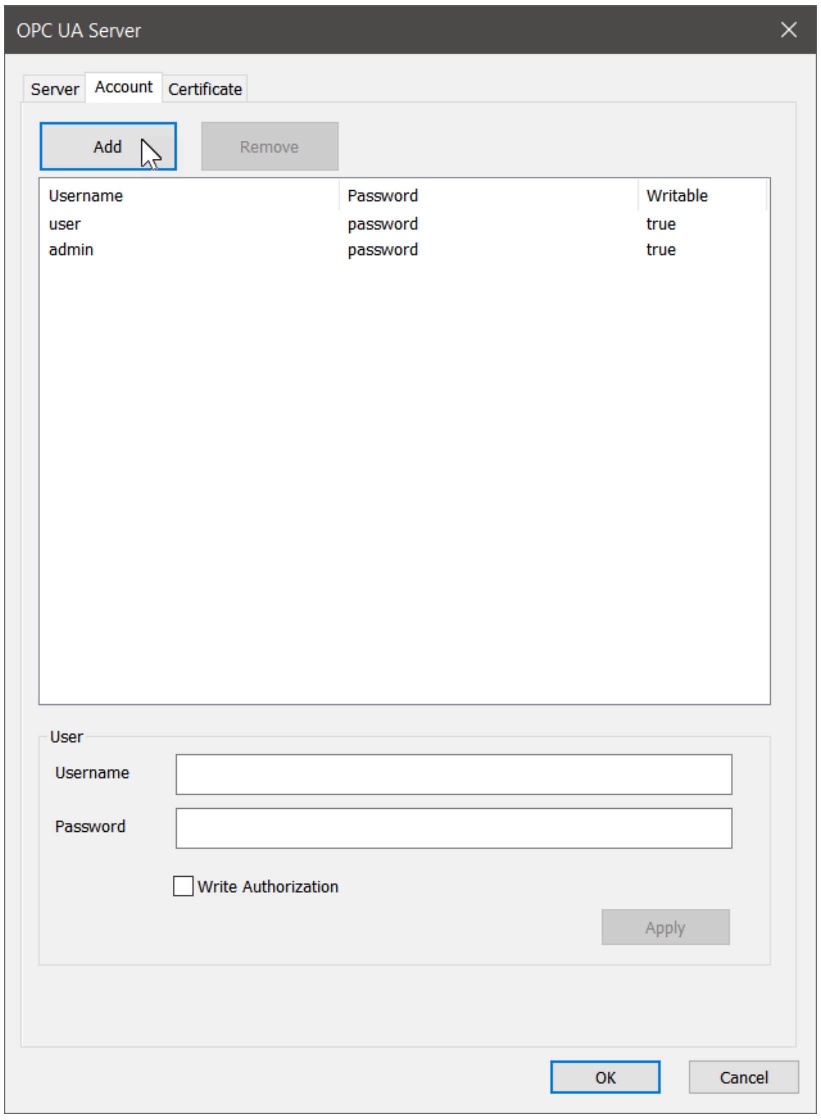
[✓] Use Username and Password

[ ] Use Certificate

[ ] Use Anonymous

Figure 8.3.2.A

The following user account list allows you enter a number of usernames with the corresponding password. In addition, the access right for each user can be limited to only read access by disabling the 'Write Authorization' checkbox. The default account entries are shown in following capture.



Username	Password	Writable
user	password	true
admin	password	true

User

Username

Password

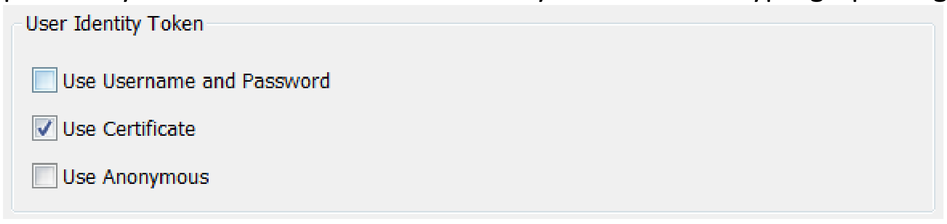
[ ] Write Authorization

Note:

It is important to remove the default entries from the account list and replace them with your user login account settings otherwise the security is impacted.

8.3.3 X509 Identity Token

This allows authenticating users using X509 certificates (Figure below). The system administrator has to provide for each user with a separate authentication certificate together with its private key which can either be stored on a smart card or on the client itself which is less secure. In addition, the system administrator has to add the authentication certificate file without its private key to the Win-GRAF server directory *'\pkiserver\trusted\certs\'* which contains all the trusted OPC UA client's certificates. The client generates a signature using the private key of the X509 certificate and the server uses the public key of the same certificate to verify the received cryptographic signature.



User Identity Token

☐ Use Username and Password

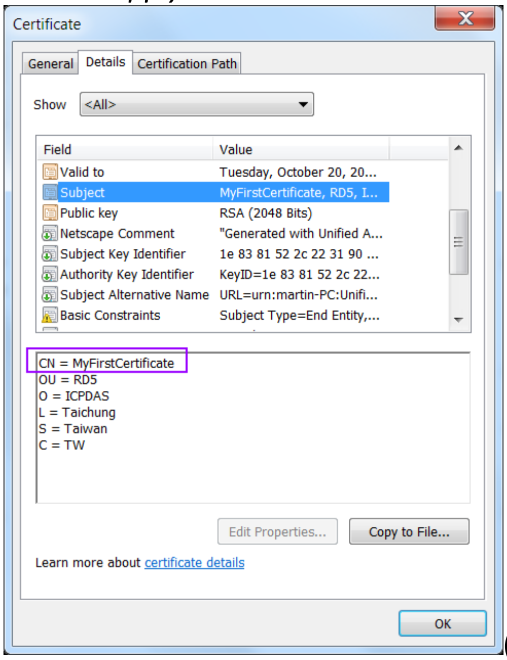
☑ Use Certificate

☐ Use Anonymous

This identity token authorization mechanism is similar to the password authentication where the *'Common Name'* (CN) field of the X509 certificate is mapped to a username and instead of a password the private key of the certificate is being used to authenticate the user. In the same way as for the password authentication the *'Common Name'* has to be added to the *'Username'* column of the user account list.

In the following the procedure for configuring the server to support X509 authentication is shown:

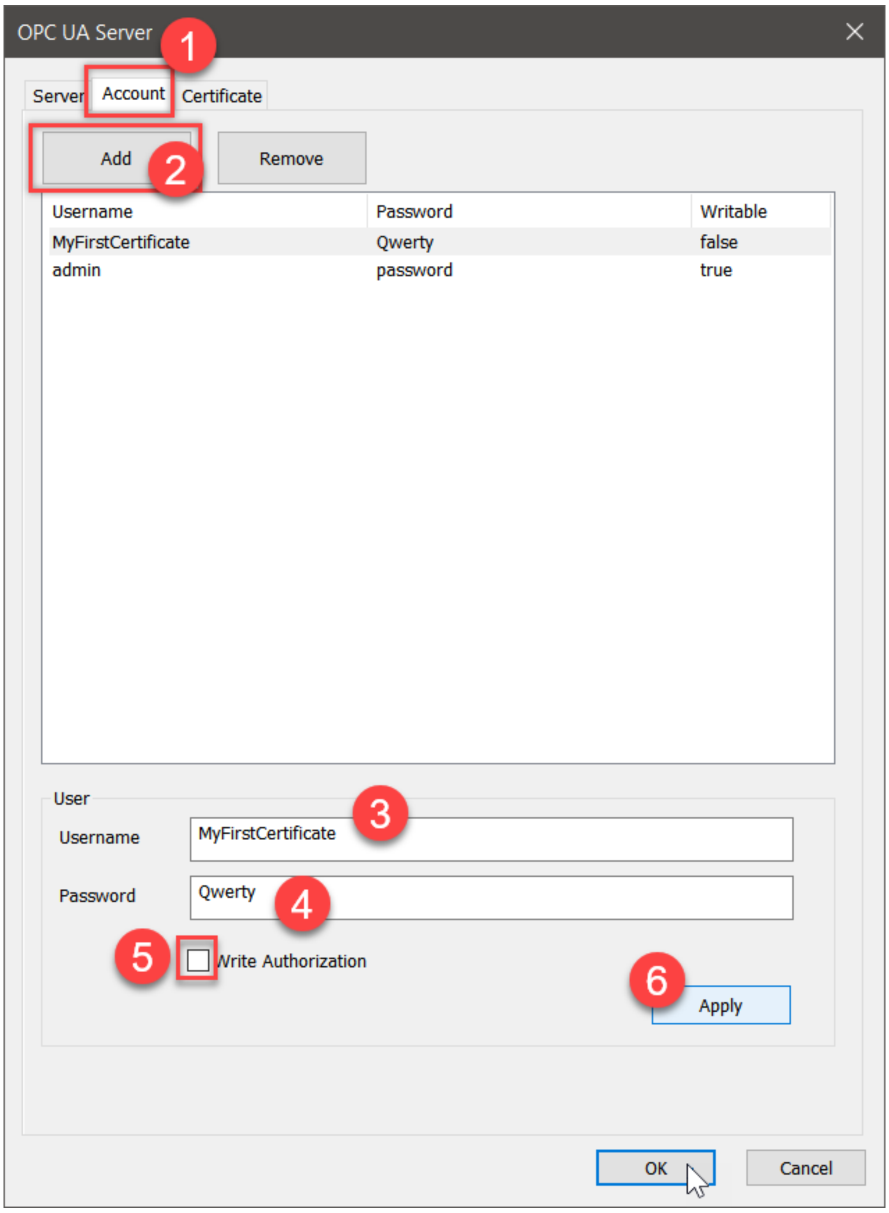
- **Step 1**: Set the identity token to X509.
  Double click the *'OPC UA Server (ICPDAS)'* node in the *'IO Driver'* window to open the main OPC UA configuration interface. Enable *'Use Certificate'* option as shown above.
- **Step 2**: Add the '*Common Name*' (CN) to the user account list.
  1. Click the 'Account' tab of the main configuration interface to display the account list.
  2. Click '*Add*' to add a new entry.
  3. Enter the '*Common Name*' (CN) of the X509 certificate (Figure 8.3.3.A) to the '*Username*' edit box (Figure 8.3.3.B).
  4. The password setting will be ignored if only the '*Use Certificate*' option is enabled (Figure 7). If the '*Use Username and Password*' option is selected as well then enter a secure password.
  5. Determine whether the user should have both read and write access rights by selecting the '*Write Authorization*' option.
  6. Click '*Apply*' and '*OK*' to confirm the setting.



Field	Value
Valid to	Tuesday, October 20, 20...
Subject	MyFirstCertificate, RD5, I...
Public key	RSA (2048 Bits)
Netscape Comment	"Generated with Unified A...
Subject Key Identifier	1e 83 81 52 2c 22 31 90 ...
Authority Key Identifier	KeyID=1e 83 81 52 2c 22...
Subject Alternative Name	URL=urn:martin-PC:Unifi...
Basic Constraints	Subject Type=End Entity,...

```
CN = MyFirstCertificate
OU = RD5
O = ICPDAS
L = Taichung
S = Taiwan
C = TW
```

(Figure 8.3.3.A)



Username	Password	Writable
MyFirstCertificate	Qwerty	false
admin	password	true

User

Username: MyFirstCertificate

Password: Qwerty

[ ] Write Authorization

Figure 8.3.3.B

9 Technical Tips

In this chapter we will introduces some applications and driver/ firmware updates that ICP DAS suggest users to implement or downloads.

9.1 How to Update Runtime on PAC?

1. Download the update pack from following linkage and unzip it.
https://www.icpdas.com/web/product/download/pac/win-graf/CE_PAC/driver/Win-GRAF_Installer_2.1.0.zip

2. Copy the file into pen drive or into SD card.

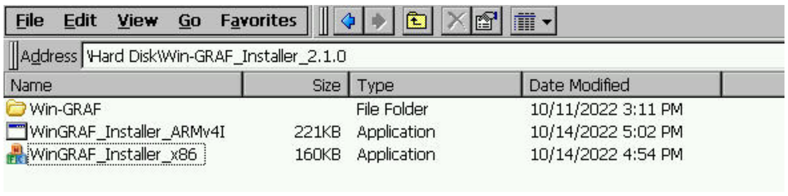
3. Insert pen drive or SD card into PAC.

4. Open „My Device“ on the PAC Desktop. Select „Hard Disk“.

5. There are two executable files:

① ......X86.exe is for XPAC series.

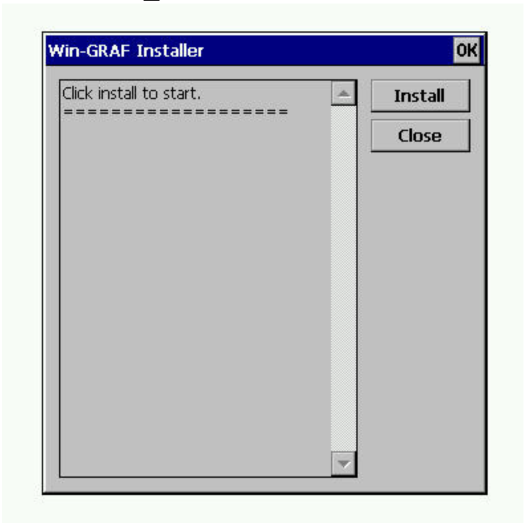
② ......ARMv4I.exe is for WinPAC-5K/8K/9K and ViewPAC.



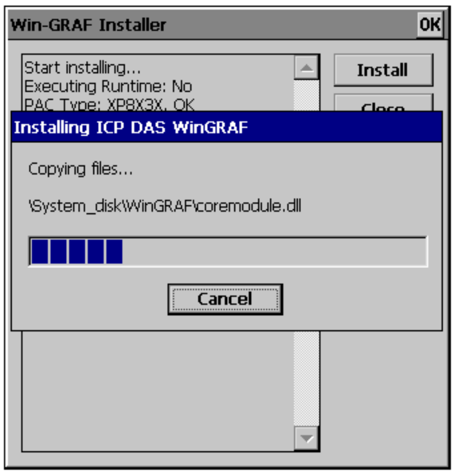
Address \Hard Disk\Win-GRAF_Installer_2.1.0

Name	Size	Type	Date Modified
Win-GRAF		File Folder	10/11/2022 3:11 PM
WinGRAF_Installer_ARMv4I	221KB	Application	10/14/2022 5:02 PM
WinGRAF_Installer_x86	160KB	Application	10/14/2022 4:54 PM

6. Execute _x86 installer. Click „Install“ to start.



Click install to start.
===================

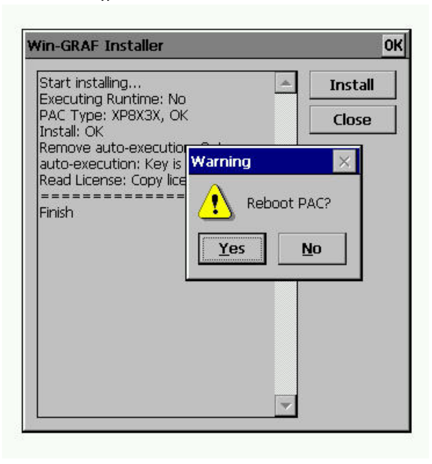


Start installing...
Executing Runtime: No
PAC Type: XP8X3X, OK

Copying files...

\System_disk\WinGRAF\coremodule.dll

7. Click „OK“ to reboot.



Start installing...
Executing Runtime: No
PAC Type: XP8X3X, OK
Install: OK
Remove auto-executio
auto-execution: Key is
Read License: Copy lic
=============
Finish

Reboot PAC?