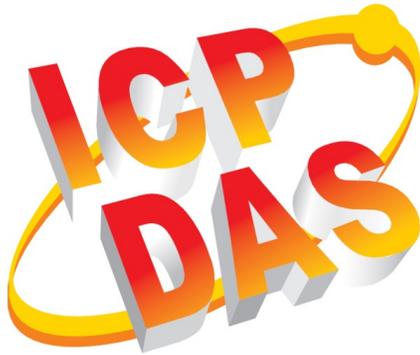


# **Local IO Wizard**

for Win-GRAF

## **User Manual**

(Version 1.0)



### **WARRANTY**

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

### **WARNING**

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

### **COPYRIGHT**

Copyright © 2022 by ICP DAS. All rights are reserved.

### **TRADEMARK**

Names are used for identification only and may be registered trademarks of their respective companies.

### **CONTACT US**

If you have any questions, please feel free to contact us via email at:

[service@icpdas.com](mailto:service@icpdas.com)

[service.icpdas@gmail.com](mailto:service.icpdas@gmail.com)

## Revision

Revision	Date	Description	Author
1.0	30.06.2022	Initial version	M. K.

# Contents

<b>1</b>	<b>OVERVIEW.....</b>	<b>5</b>
<b>2</b>	<b>SOFTWARE INSTALLATION.....</b>	<b>5</b>
2.1	WORKBENCH.....	6
2.2	RUNTIME.....	6
<b>3</b>	<b>IO CONFIGURATION.....</b>	<b>7</b>
3.1	OPEN THE IO WIZARD.....	7
3.2	IO SELECTION AND CONFIGURATION PROCEDURE.....	9
<b>4</b>	<b>APPENDIX.....</b>	<b>17</b>
4.1.1	Extension Module Error.....	17

---

# 1 Overview

---

The PAC series from ICPDAS are equipped with slots which supports the extension of local IO and communication interfaces (serial, Ethernet, USB). ICPDAS provides a large variety of IO extension modules starting from simple digital IO, to analogue IO for current, voltage and temperature measurement.

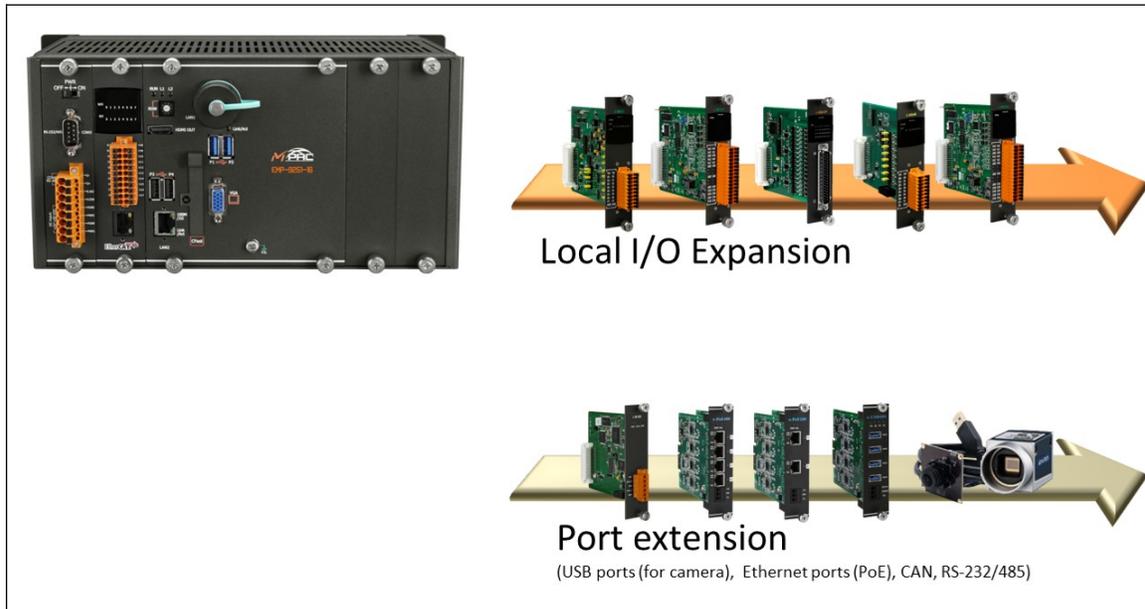


Figure 1: Local expansion modules for PAC series

The purpose of this manual is to describe how to use the Win-GRAF IO Wizard to configure and integrate each module into the PLC programming environment. The IO Wizard provides a graphical user interface to add a module to the PAC IO slot via drag and drop mechanism. The IO wizard can only be used in conjunction with the IO modules supported by the PAC series from ICPDAS.

---

## 2 Software Installation

---

### 2.1 Workbench

The Win-GRAF workbench setup program "Win-GRAF\_Workbench\_xxxx\_Setup" automatically installs all the required IO Wizard plugin libraries for the IO extension module integration. The following files are installed:

*C:\Program Files (x86)\Win-GRAF Workbench\Win-GRAF Wb xx.xx\IOD\K5BusPacIo.dll*  
*C:\Program Files (x86)\Win-GRAF Workbench\Win-GRAF Wb xx.xx\IOD\PacIoProfile\..*  
*C:\Program Files (x86)\Win-GRAF Workbench\Win-GRAF Wb xx.xx\IOD\PacIoBitMap\..*

## 2.2 Runtime

By default the IO runtime DLL ("*ddkc\_pacio.dll*") is being installed by the runtime setup program in the directory of the runtime execution file "*WinGrafRuntime.exe*". If you manually move the runtime execution file to a different directory make always sure that the IO runtime DLL exists in the runtime directory.

## 3 IO Configuration

ICPDAS offers two type of PAC series:

- 8000 series: XP-8000-WES7, LP-8000, WP-8000-CE7/WP-8000
  - Supported IO modules: I-8K and I-87K
- 9000 series: WP-9000-CE7, XP-9000-WES7, XP-9000-IoT, EMP-9000
  - Supported IO modules: I-9K and I-97K

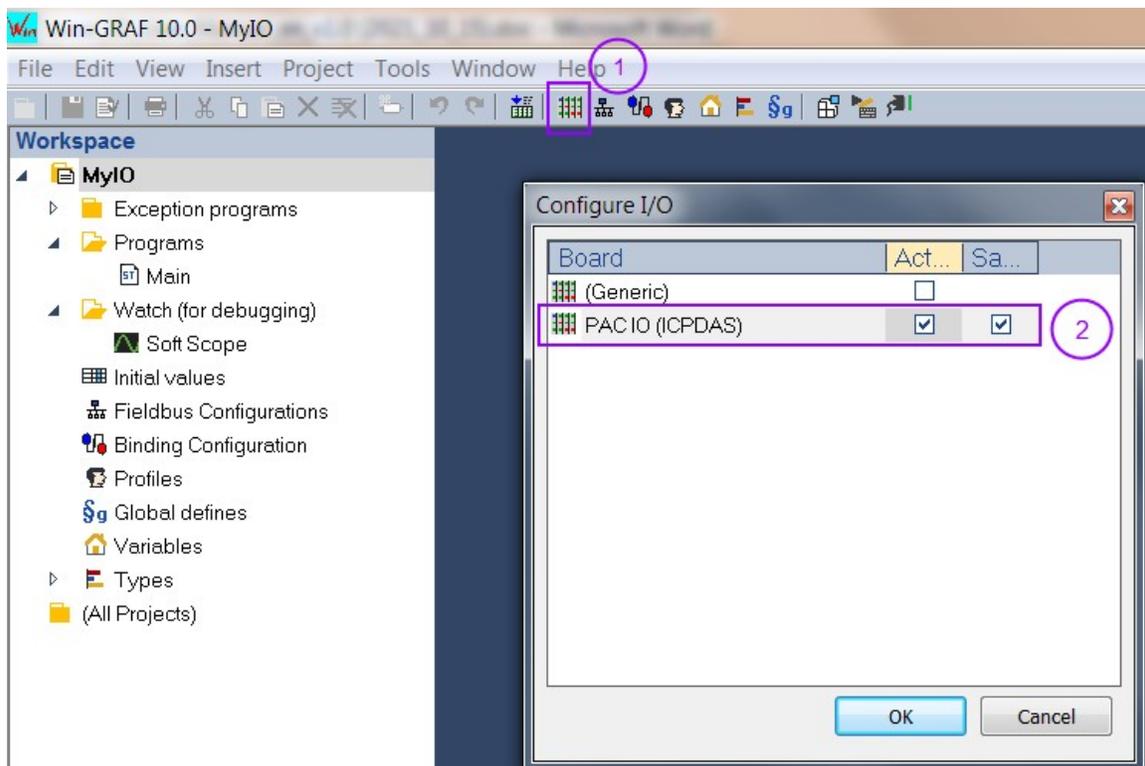
The form factor of both the main control unit and the IO extension modules differs between both series. It is therefore important to select the correct IO module type for respective controller type.

### 3.1 Open the IO Wizard

In the following the it is shown how to open the IO wizard in the Win-GRAF workbench.

**Step 1:** Click the 'Open I/Os' button in the tool bar

**Step 2:** Select 'PAC IO (ICPDAS)' and click 'OK'



The user interface of the IO wizard will be displayed in the workbench (Figure 2).

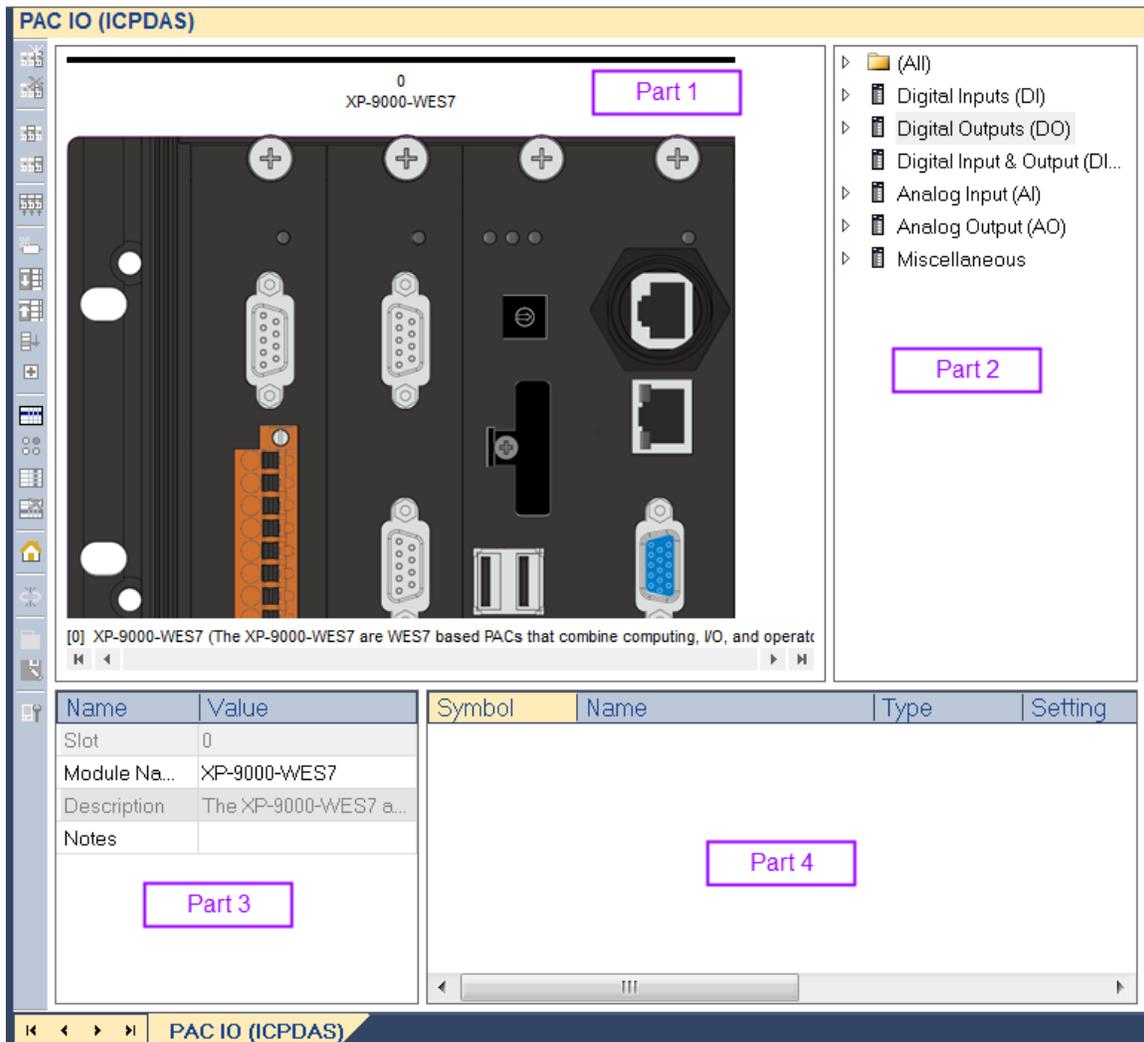


Figure 2: IO Wizard

The user interface is divided into four section:

- Part 1: Displays the main control unit (PAC) with the selected IO modules.
- Part 2: List all the IO modules supported by the PAC series. The modules can be added to the PAC slot via drag & drop procedure: Select a IO module from the list and drag it to the PAC device displayed in the first window (Part 1).
- Part 3: Displays information of the selected module. Click on either the PAC main control unit or one of its IO module in the slot to display the name and description. In addition it allows to select the PAC type to be used and add some notes to describe the main purpose of the device in the application.
- Part 4: Displays the mapped variables to the in- or output channels of the IO module

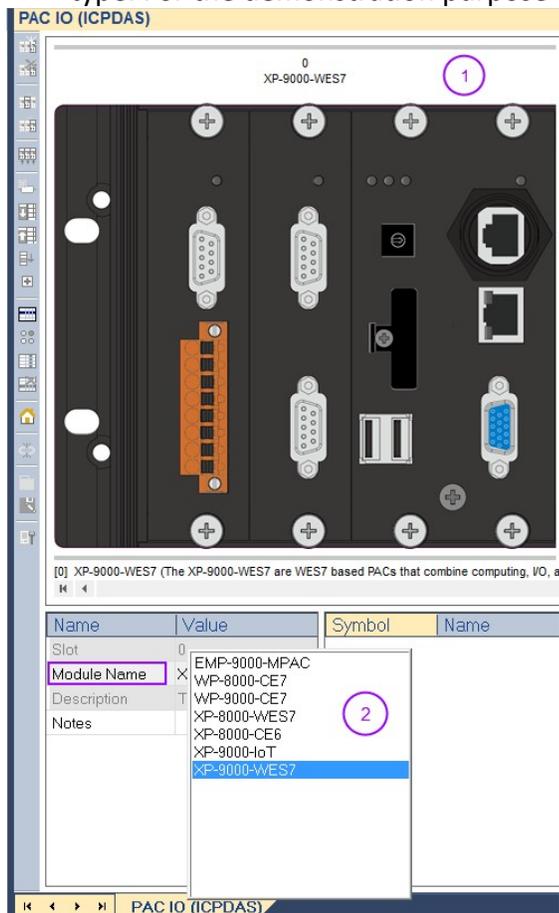
## 3.2 IO Selection and Configuration Procedure

This section provides a quick overview how to add and configure a IO slot module with the Win-GRAF workbench. The IO wizard can not scan the target platform and load the current IO module setup and configuration. The user has to use the IO wizard to manually add the IO module to the PAC slot via drag & drop according to the IO setup in the target platform.

In the following the steps are shown for IO selection and configuration.

**Step 1:** Select the PAC type.

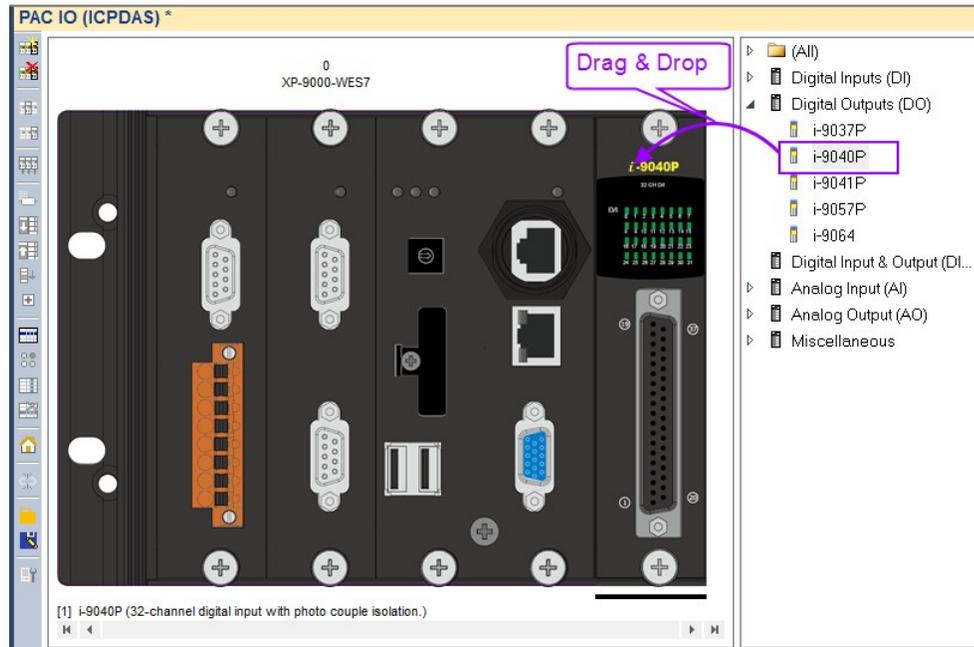
1. Click on the main control unit image (Part 1) on the top left window
2. Click on the drop box next to 'Module Name' to select a PAC type. The image of the main control unit will change according to the selected PAC type. For the demonstration purpose 'XP-9000-WES7' is selected.



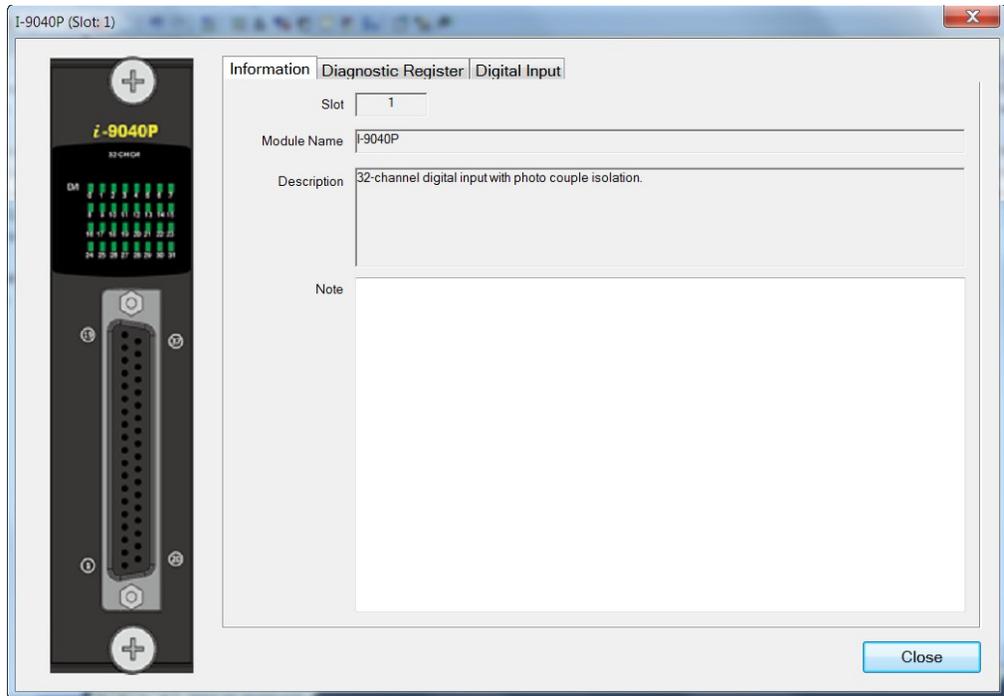
**Step 2:** Add the IO module to the PAC extension slots.

Expand the IO module tree on the right window; select a module item and

drag it over the PAC image. In this example the 32 DI channel i-9040P is added to the first slot.

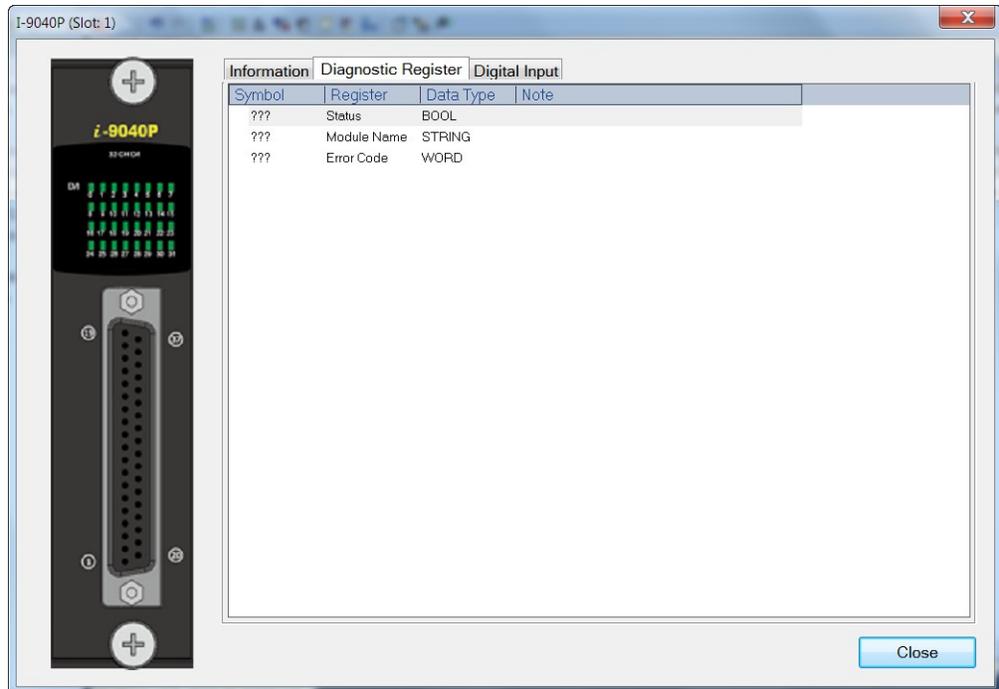


- Step 3:** Parameter mapping and configuration. In this step PLC variables are defined and mapped to the individual channels of the inserted slot modules.
1. Double click the slot module image on the top window section. A window with different tabs pops up. The first tab gives more detailed information of the slot module, e.g. the slot number in which the module has been inserted, its name and description. In the 'Note' section the user can describe the purpose of the module and its channels in the application.



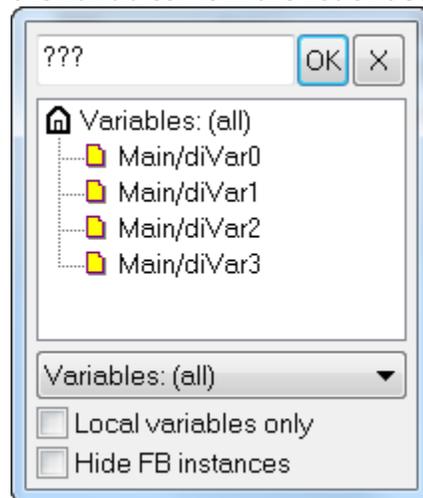
- The purpose of the 'Diagnostic Register' tab is to allow the PLC program to directly access the modules name, communication status and the error type for failed module access.

Diagnostic Register	Description
Status	<ul style="list-style-type: none"> <li>The 'Status' indicates whether the real module plugged into the slot is identical to the module configured via the IO wizard.</li> <li>If both modules matches the 'Status' variable will be set to TRUE. The IO module checking will only be done during the initialization phase.</li> </ul>
Module Name	<ul style="list-style-type: none"> <li>Displays the module name plugged into the slot</li> <li>For performance reason the module name will only be read during the initialization phase.</li> <li>By reading the 'Module Name' the PLC application can ensure that the correct module has been inserted before communication starts</li> </ul>
Error Code	<ul style="list-style-type: none"> <li>In each PLC cycle the communication state of the each slot module will be checked to ensure that the data exchanges was successful. Communication errors for example may occur if the extension module has not been correctly plugged into the slot.</li> <li>See error table for more information</li> </ul>

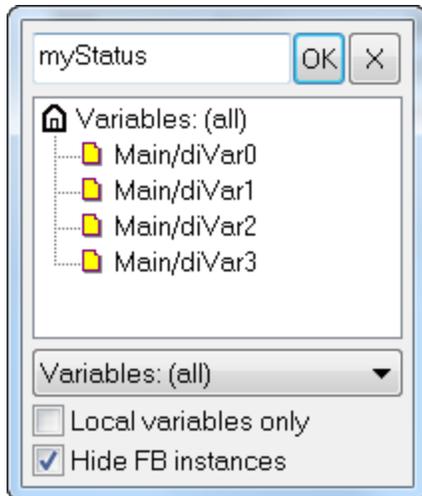


**3. Diagnostic register variable mapping: Assign PLC variables to the diagnostic registers**

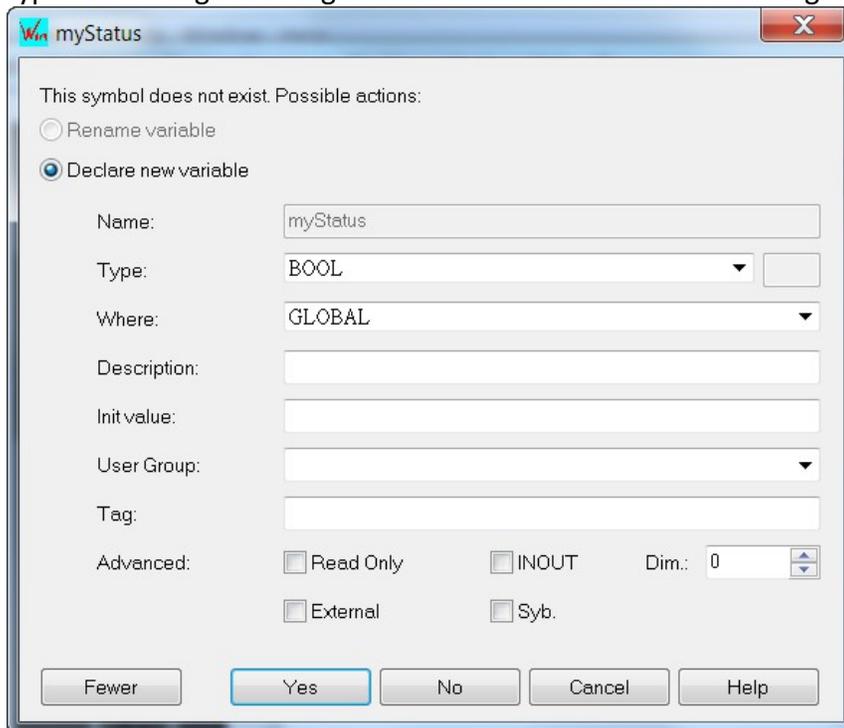
- Double click '???' to declare a PLC variable or select an existing PLC variable to hold the value of the mapped register. A variable editor window appears which list all the declared variables. Select one of the variables from the list or declare a new variable.



- Declaring a new variable: Enter a variable name in the edit box at the top of the variable editor which displays the '???' symbols. In this example the name '*myStatus*' is selected. Enter the 'OK' button.



Another window pops up which allows you to select the data type, initial value etc.. Make sure that the selected data type matches the type of the diagnostic register. Click 'Yes' to confirm the setting.



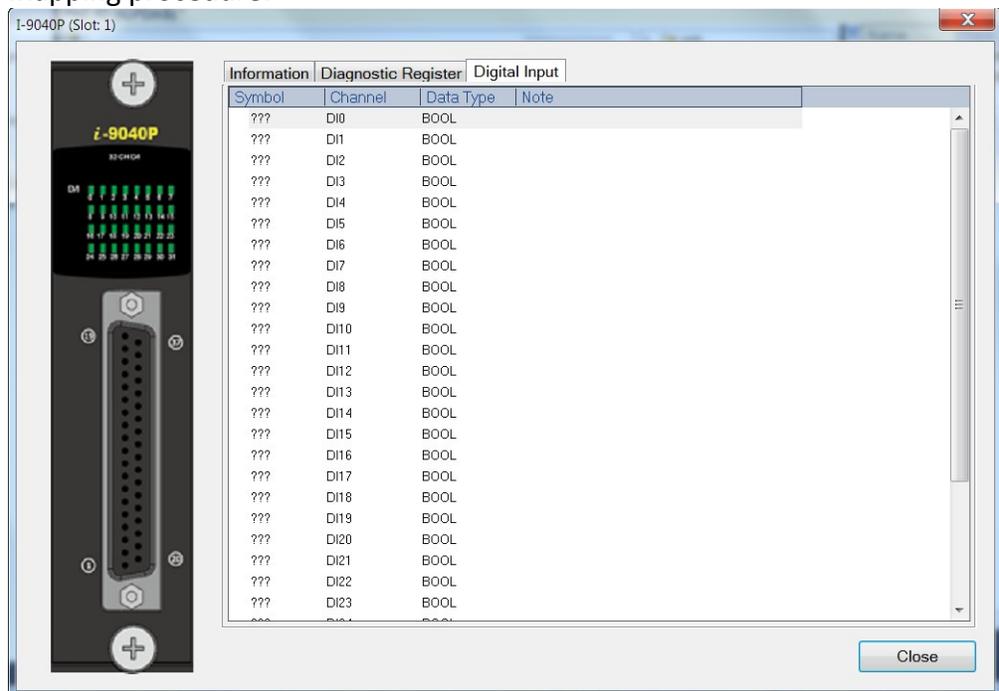
The newly declared variable will be shown next to the mapped register. The 'myStatus' variable will be updated with the 'Status' register in each PLC cycle.

Information	Diagnostic Register	Digital Input	
Symbol	Register	Data Type	Note
myStatus	Status	BOOL	
???	Module Name	STRING	
???	Error Code	WORD	

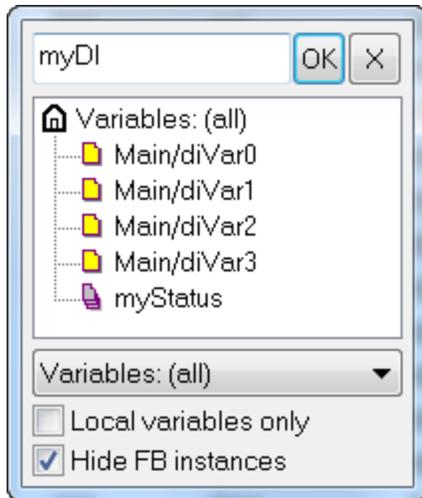
4. Channel register variable mapping: Each I/O channel has to be mapped to a separate PLC variable.

The I/O channel variable mapping procedure is identical to the diagnostic register mapping procedure described in the previous step.

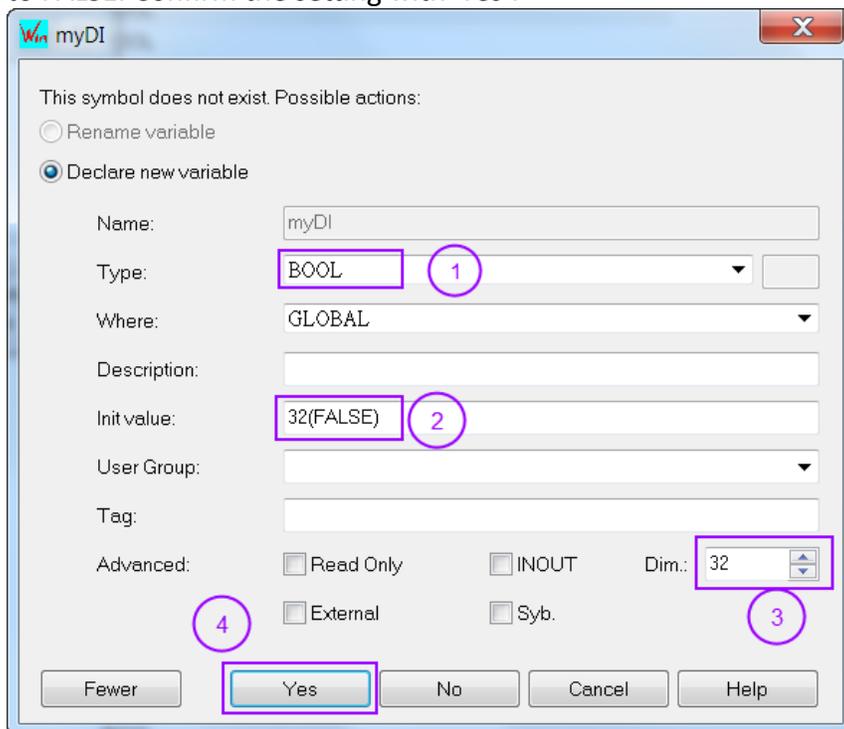
Using the i-9040P as an example we will show how to simplify the mapping procedure:



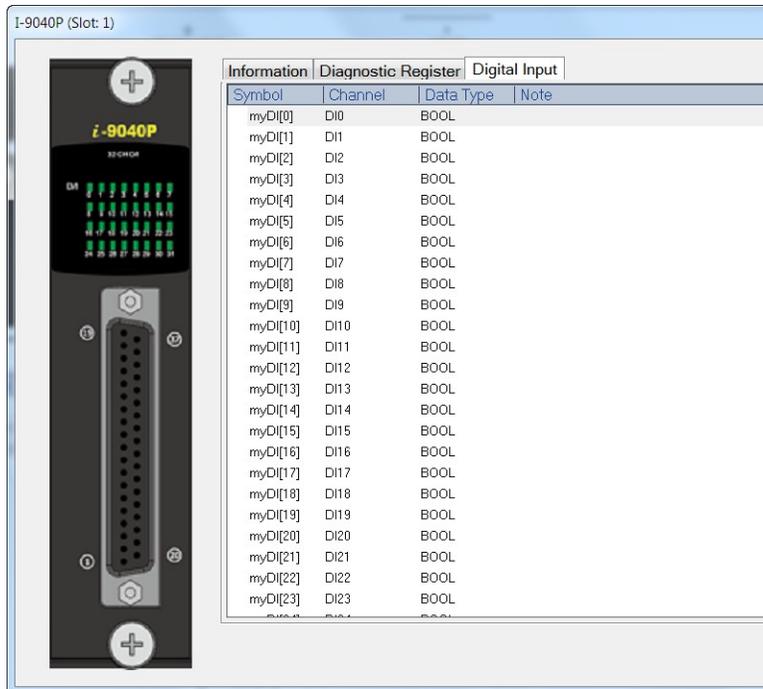
The i-9040P module has got 32 DI channel. Each entry in the 'Digital Input' table represent one DI channel. Declare a BOOL array with 32 elements by double click on any 'Symbol' cell displaying '???'. In the popup variable editor enter a variable name (for example 'myDI') and click 'OK'.



- In the popup declaration editor select the BOOL data type and set the dimension to 32 elements. Initialize all 32 element in the array to FALSE. Confirm the setting with 'Yes'.

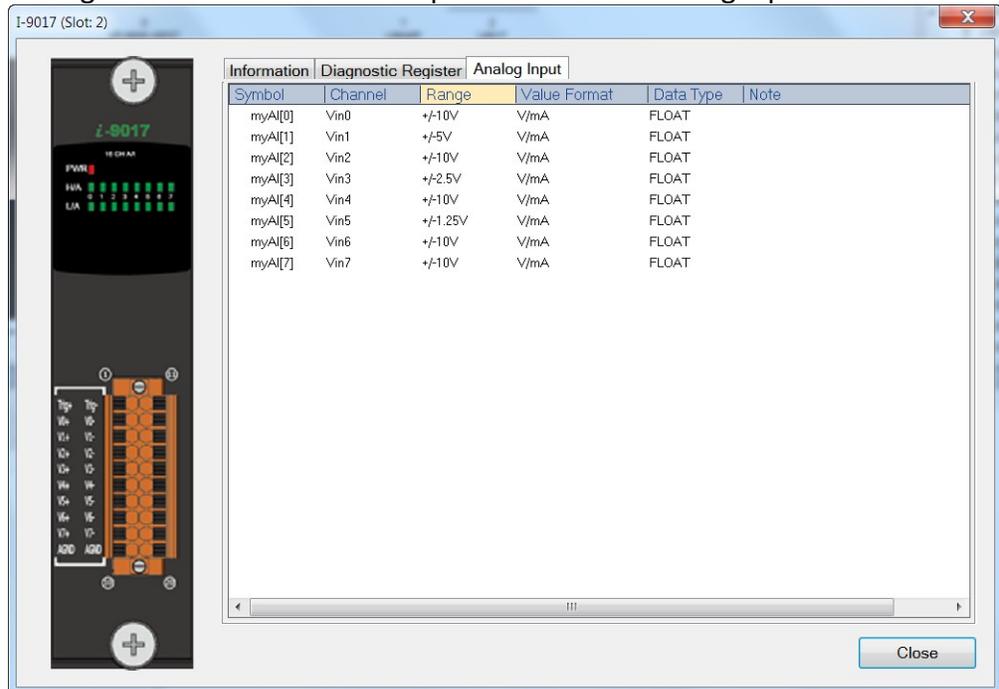


- Enter the array name with index to the 'Symbol' column.



- Click 'Close'. Now the PLC application can access the DI channel status. The mapped variables are updated in each cycle.
5. The variable mapping procedure for analog and digital expansion module are the same except that some additional configurations are necessary such as setting the measured analog value range, its physical unit and the value format.

The figure below shows an example of the i-9017 analog input module.



---

## 4 Appendix

---

### 4.1.1 Extension Module Error

Each time the PLC accesses an extension module the '*Error Code*' register will indicate whether the access was successful:

Error	Description
0x0000	<ul style="list-style-type: none"><li>No error occurred</li></ul>
0x0001	<ul style="list-style-type: none"><li>Module type mismatch. The module plugged into the PAC slot does not match with the module setup via the IO wizard</li></ul>
0x0002	<ul style="list-style-type: none"><li>Initializing the extension module failed during the start-up phase</li></ul>
0x0100	<ul style="list-style-type: none"><li>Writing the digital output channel failed</li></ul>
0x0101	<ul style="list-style-type: none"><li>Reading the digital input channel failed</li></ul>
0x0102	<ul style="list-style-type: none"><li>Writing the analog output channel failed</li></ul>
0x0103	<ul style="list-style-type: none"><li>Reading the analog input channel failed</li></ul>

**Table 1:** '*Error Code*' definitions