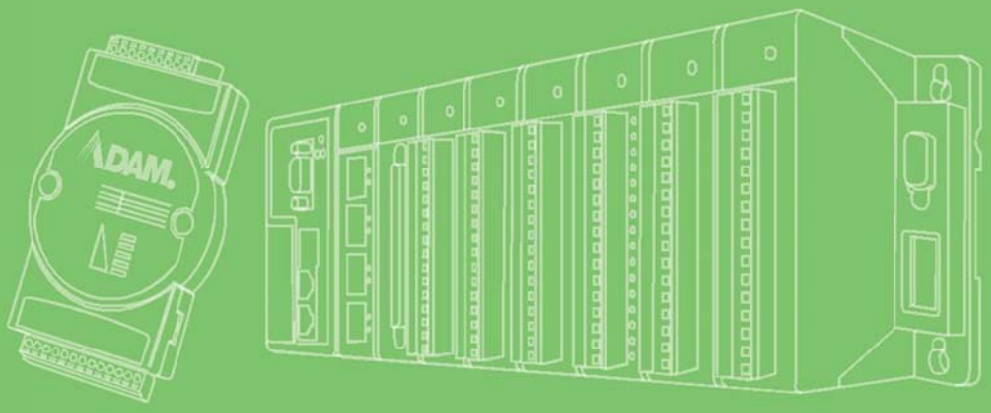


User Manual

Advantech CODESYS for ADAM-5560

CONTENTS

1. Introduction 4

1.1. About This Manual 4

1.2. Organization of This Manual 4

2. Installations 7

2.1. CODESYS Installation 7

2.2. Add-on Package Installation 10

2.2.1 First-time Installation 10

2.2.2 Updating the Package 13

3. Create and run a project 16

3.1. Start CoDeSys 16

3.2. Create a Project 17

3.3. Write a Program 19

3.4. Connect to the Target Device 21

3.5. Run the Application 23

4. Advantech I/O Modules 26

4.1. Insert I/O Modules into CODESYS 26

4.2. Map Variables to I/O Modules 27

4.3. Support List 29

4.4. Digital Input Modules 30

4.5. Digital Output Modules 32

4.6. Analog Input Modules 34

4.7. Analog Output Modules 35

4.8. Relay Output Modules 37

4.9. Counter/Frequency Modules 38

5. Diagnosis and Troubleshooting 44

5.1. Error Notification 44

5.2. Log Information 44

5.3. Error ID 45

Chapter 1

1. Introduction

1.1. About This Manual

This document describes the use of the CODESYS programming environment and the Wince runtime system for the Advantech ADAM-5560 series products.

Advantech provides add-on package for CoDeSys which allows developers and end users to connected I/O modules, perform configurations, and simple testing of the I/O.

This manual supplies information about how to apply CoDeSys to control Advantech ADAM-5560, including software installation, writing a new program in CoDeSys to testing ADAM I/O modules.

1.2. Organization of This Manual

This user manual is divided into the following sections:

- Introduction
- Installations
- Create and run a project
- Advantech I/O Modules
- Diagnosis and Troubleshooting

Introduction

This section gives the user a basic idea of this manual.

Installations

This section provides instructions on how to install CoDeSys and Advantech Add-on Package

Create and run a project

This section gives the new user a walk-through in creating a simple program.

Advantech I/O Modules

This section introduces the detail configuration and mapping variables of Advantech ADAM I/O modules

Diagnosis and Troubleshooting

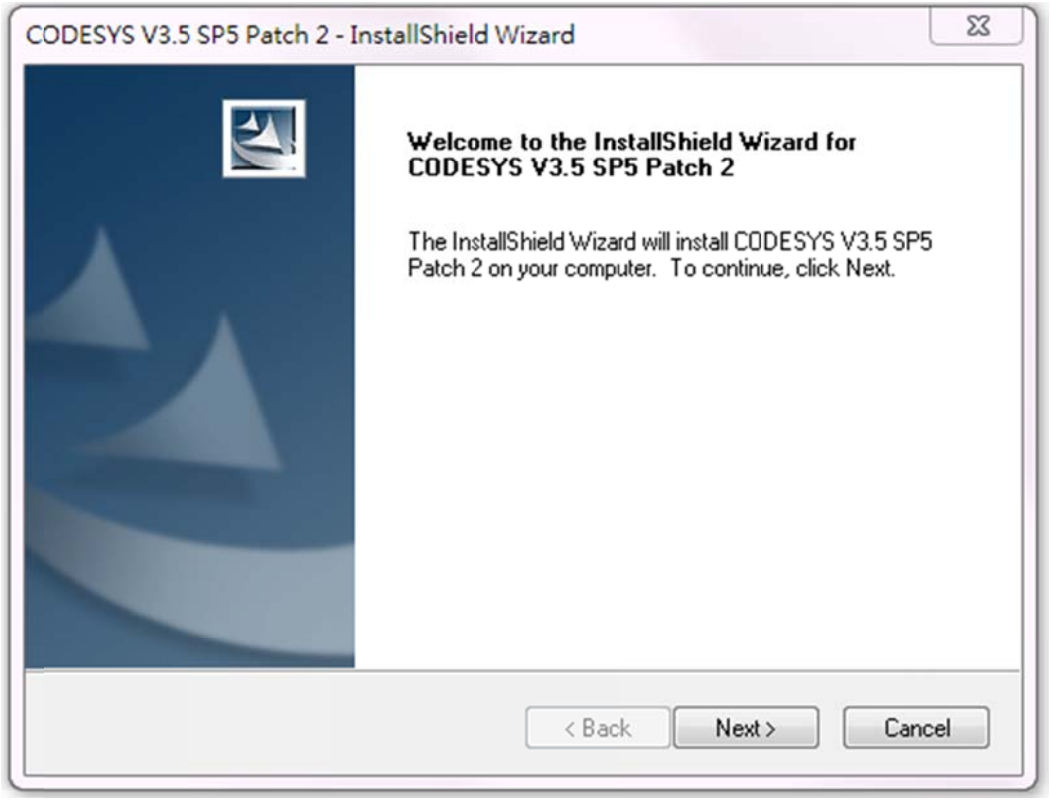
This section provides instructions on how to troubleshooting and diagnose operation mistakes or module errors.

Chapter 2

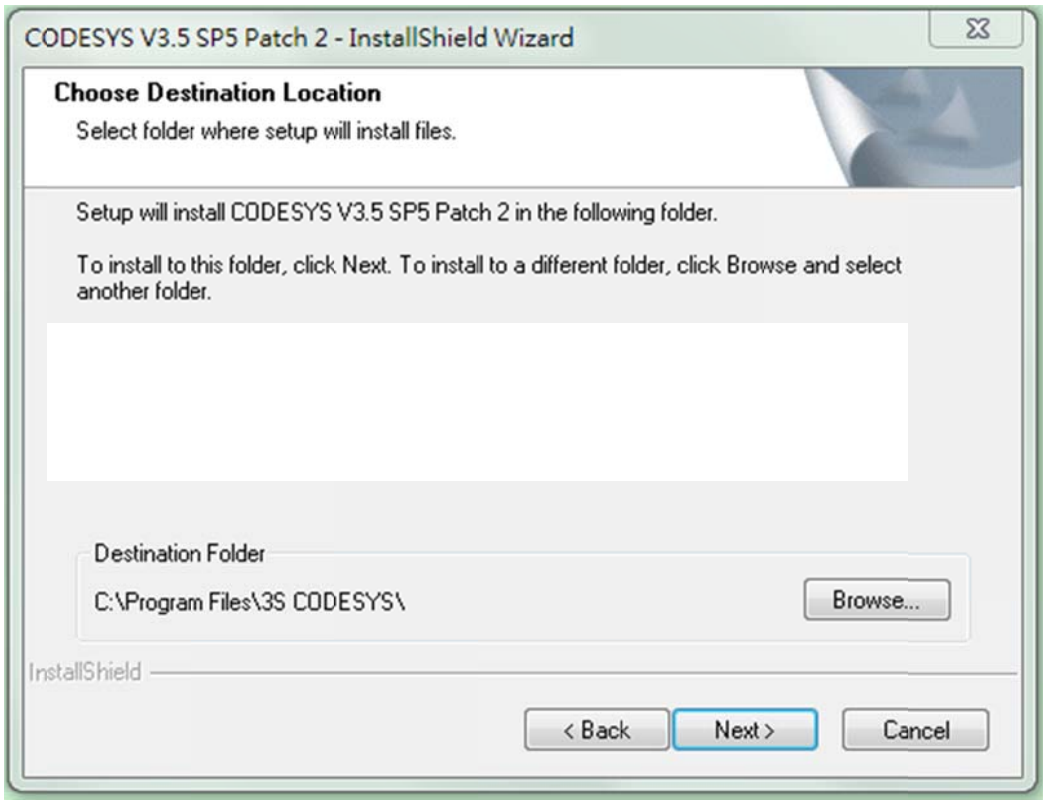
2. Installations

2.1. CODESYS Installation

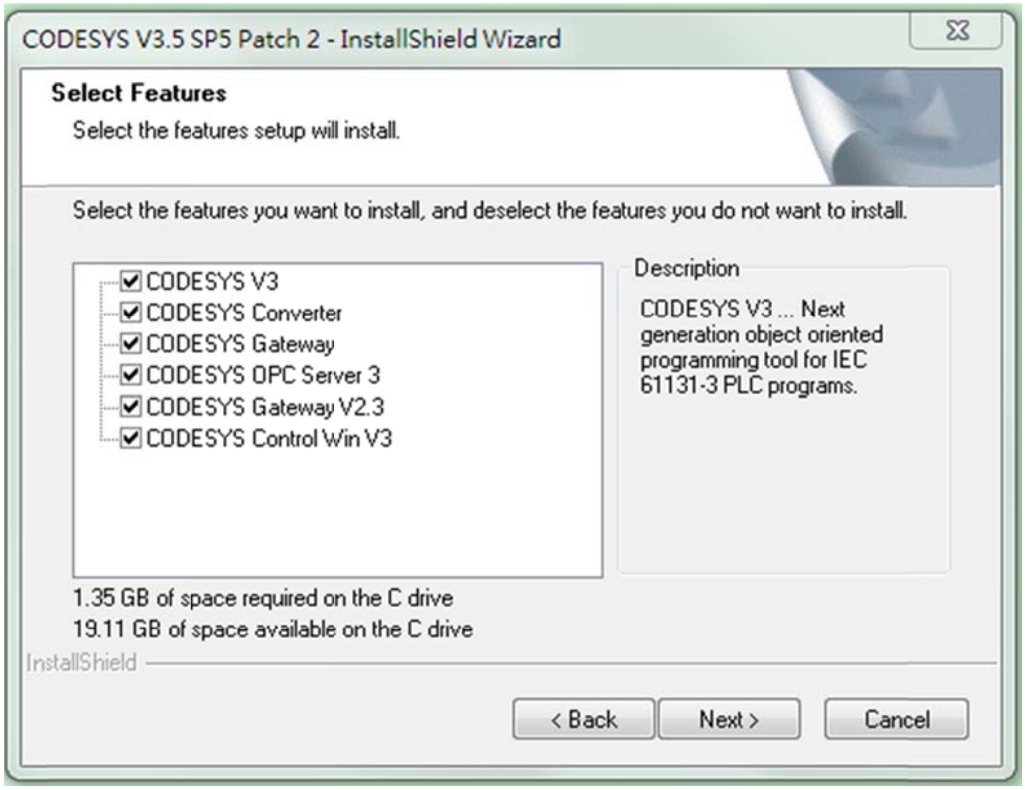
Step1: Double click and execute the "**Setup_CODESYSV<Version>.exe"** to start the installation assistant and then click Next to continue.



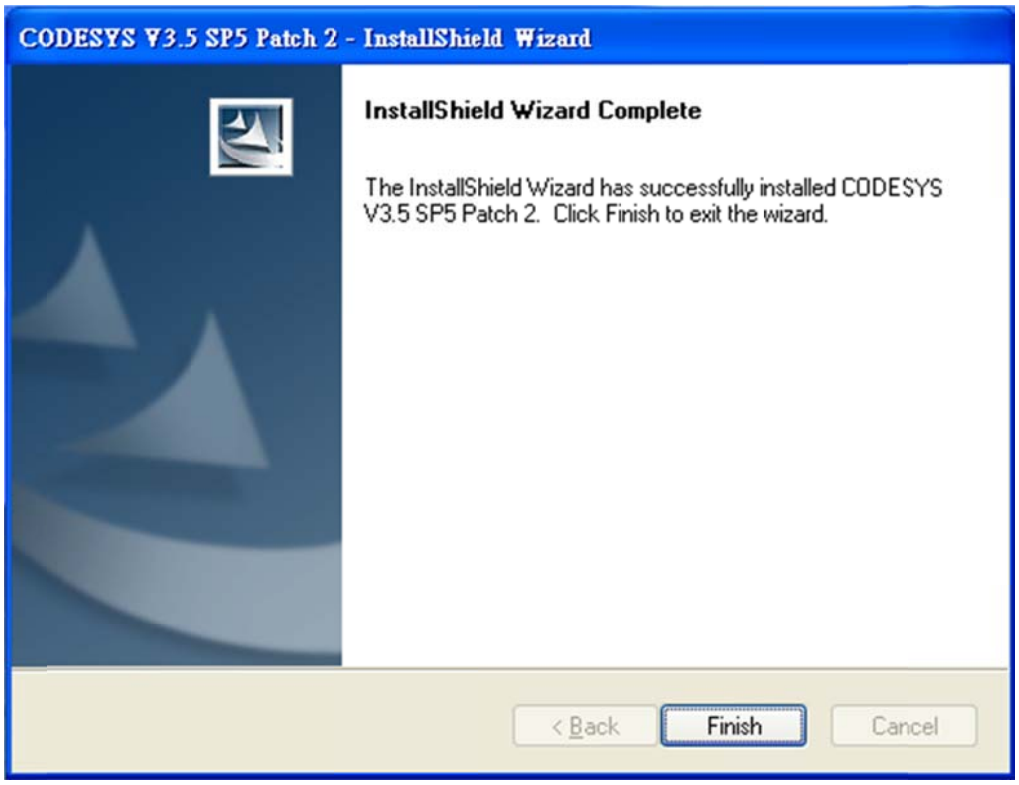
Step 2: You will then be prompted for the installation location. By default, CoDeSys will install to C:\Program Files\3S CODESYS, but you can specify the location or folder name of your choice. Click Next to proceed.



Step 3: Select all features and then click Next to proceed..



Step 4: Complete to install CoDeSys and you'll see CoDeSys icon  which is available on the desktop. Click "Finish" to close the installation wizard.



2.2. Add-on Package Installation

Now that you have CoDeSys on your system, you'll want to do a few steps to install add-on package on your environment.

2.2.1 First-time Installation

Step 1: Install the latest version of the add-on package by double-clicking the executable "**Advantech ADAM CODESYS ADD ON.package"**.

Note!

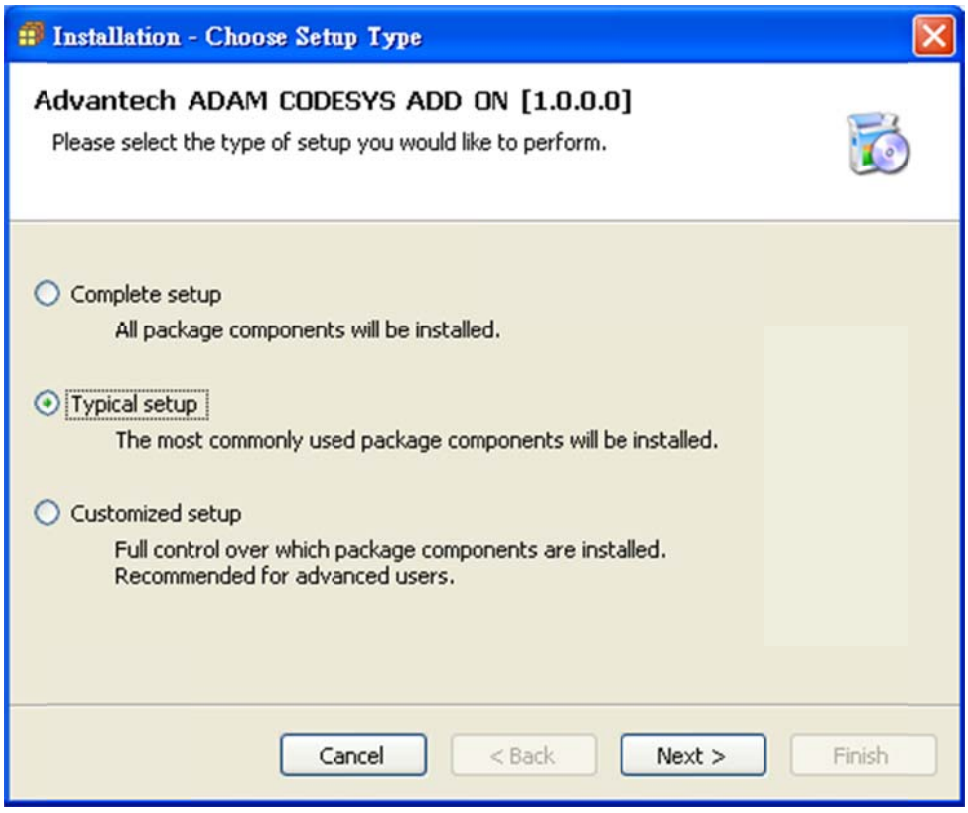
If you get an error stating "you must have administrative rights in order to install or uninstall a package" on Window 7, you should turn UAC off.

(a) Open User Account Control Settings by clicking the **Start** button , and then clicking **Control Panel**. In the search box, type **uac**, and then click **Change User Account Control settings**.

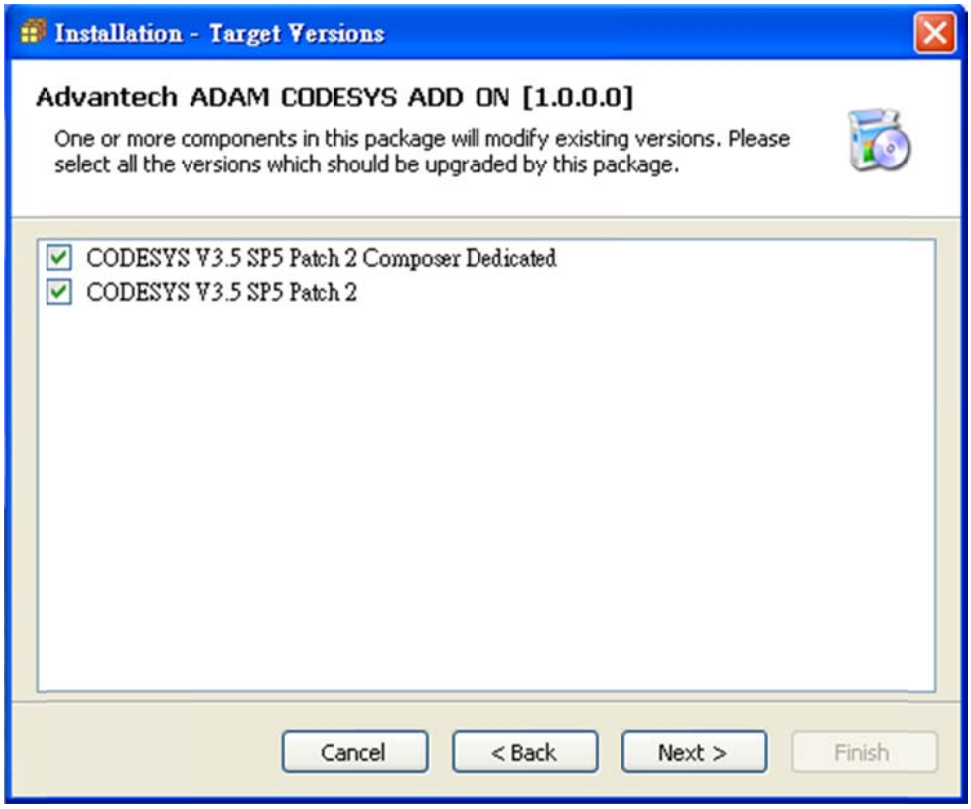
(b) Move the slider to the Never notify position, and then click **OK**. If you're prompted for an administrator password or confirmation, type the password or provide confirmation. You will need to restart your computer for UAC to be turned off.

For more information about notification options, please refer to Microsoft website.

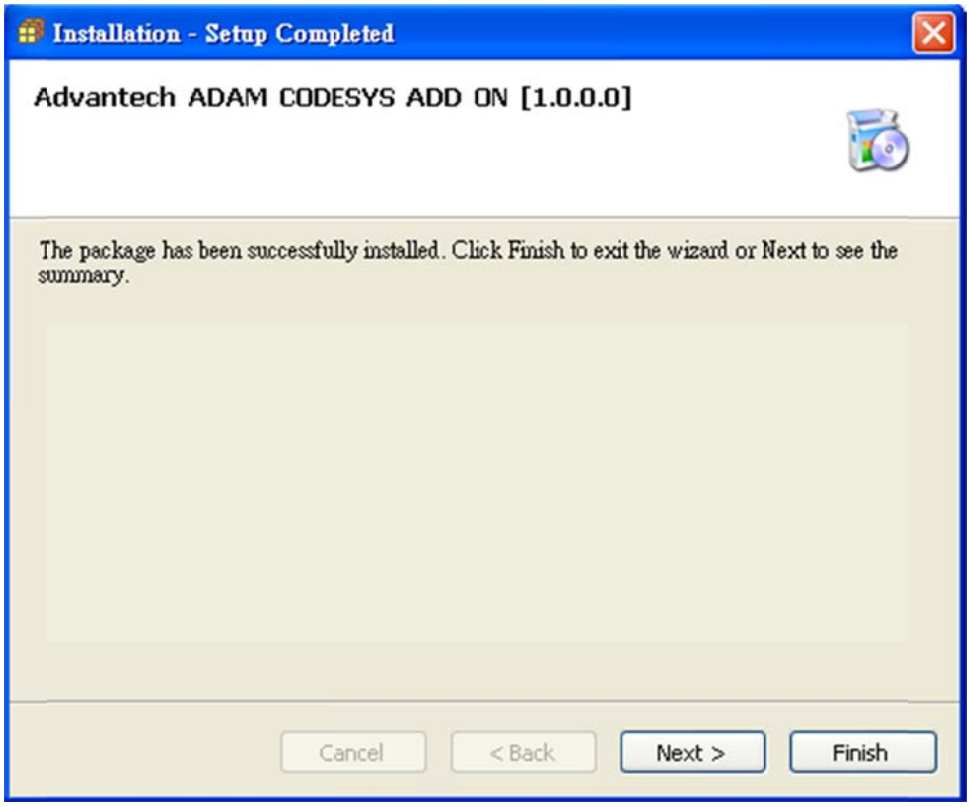
Step 2: When you are asked to choose setup type, choose the "**Typical setup**" and then click Next.



Step 3: Select all versions and then click next.



Step 4: After the files have installed, you will see the completion screen. Click "Finish" to close the installation wizard.

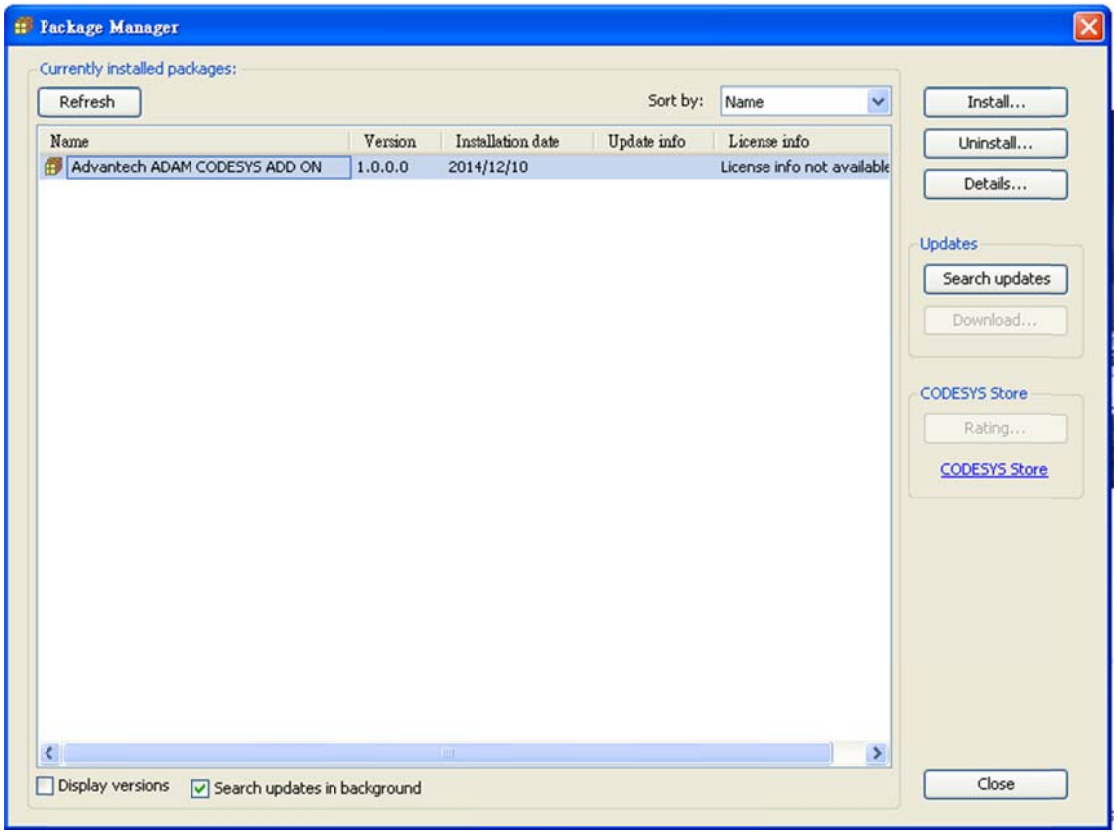


2.2.2 Updating the Package

It's highly recommended that you uninstall the previous version package before updating and installing new add-on package, so you can access to **Control Panel** and uninstall the previous version package manually.

Alternatively, we can start CoDeSys and perform command **Package Manager** from the menu (**Tools -> Package Manager**). Select the package you want to uninstall and then click “Uninstall”. Click "Close" to close the package manager.

After uninstalling the old package successfully, please refer to First-time Installation.


Package Manager
Currently installed packages:
Refresh
Sort by:
Name
Name
Version
Installation date
Update info
License info
Advantech ADAM CODESYS ADD ON
1.0.0.0
2014/12/10
License info not available
Install...
Uninstall...
Details...
Updates
Search updates
Download...
CODESYS Store
Rating...
CODESYS Store
Display versions
Search updates in background
Close

Chapter 3

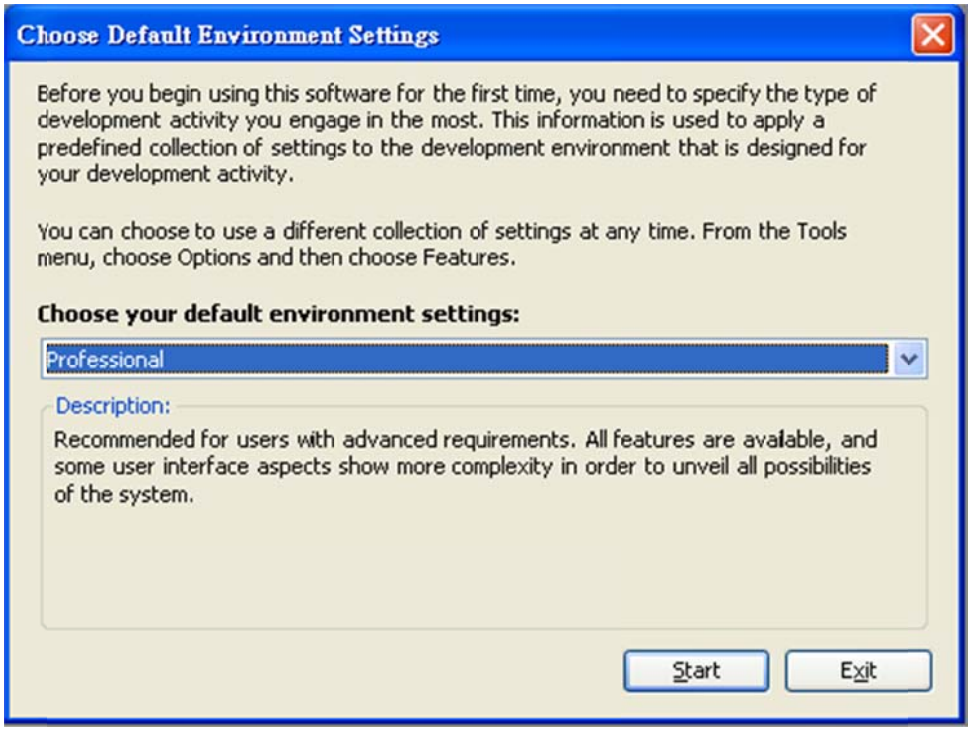
3. Create and run a project

3.1. Start CoDeSys

Start CoDeSys by double-clicking the CoDeSys icon  which is available on the desktop. Alternatively, you can start the CoDeSys programming system with

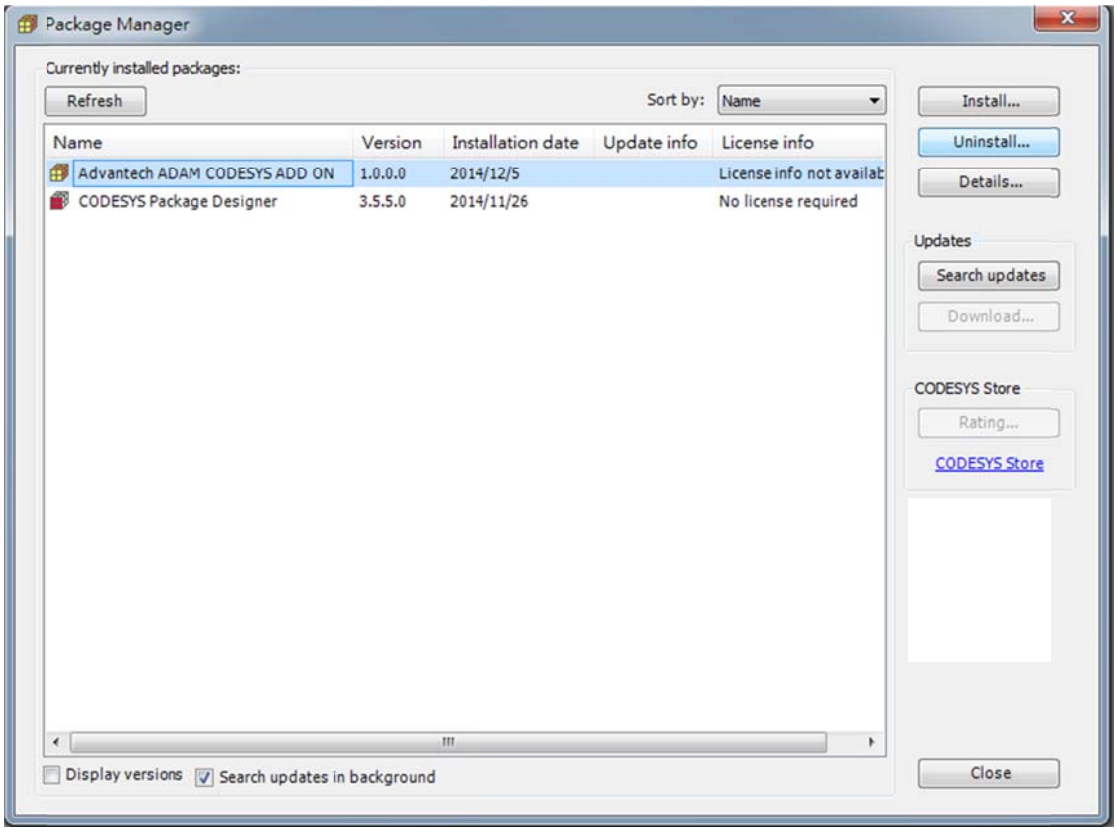
Start -> Programs -> 3S Software -> CoDeSys -> CODESYS V<version>

When you start the programming system the first time after first installation on the system, you will be asked to choose the default collection of settings and features. Choose the "**Professional**" and then click Start to proceed.



Before creating a project, make sure that Advantech ADAM add-on package is installed successfully. Choose **Package Manger** from the **Tools** menu:

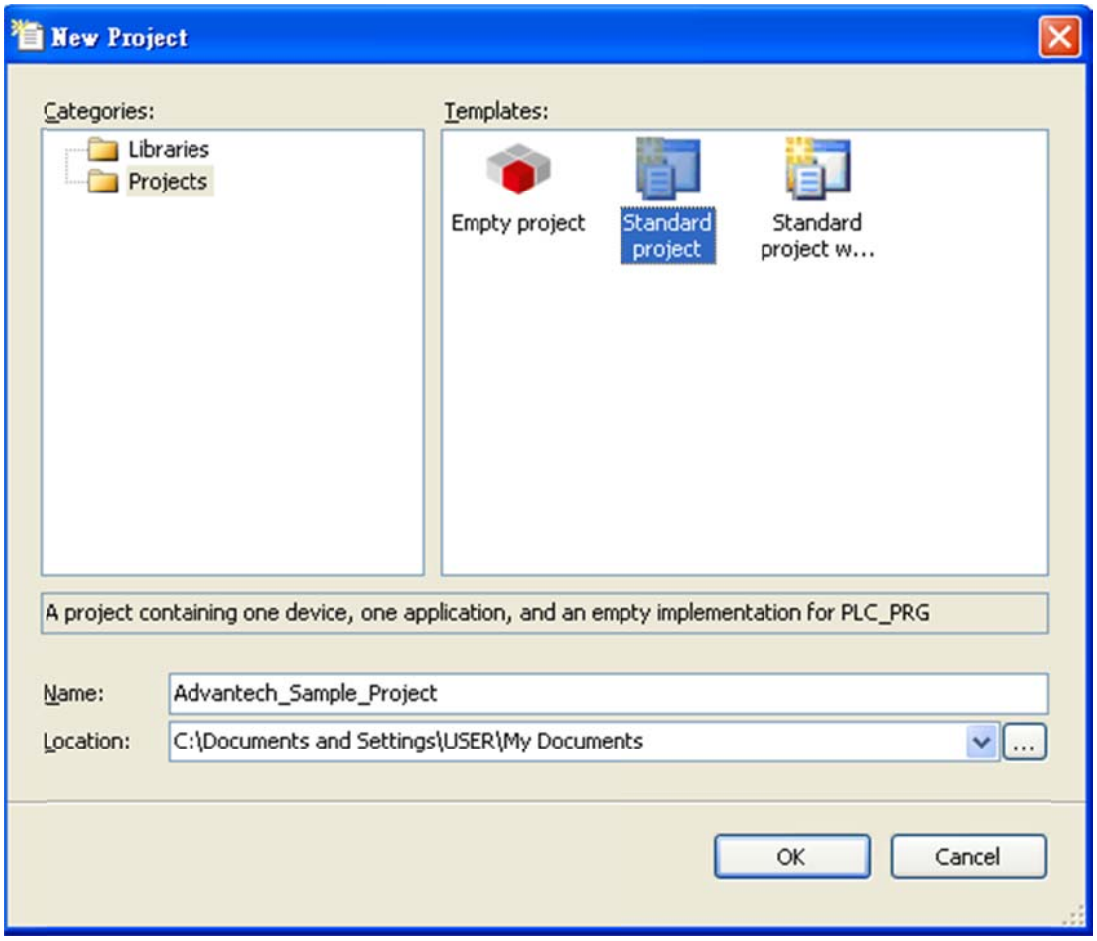
If the Advantech ADAM add-on package didn't show in manager, please refer to Chapter 2 and update your package.



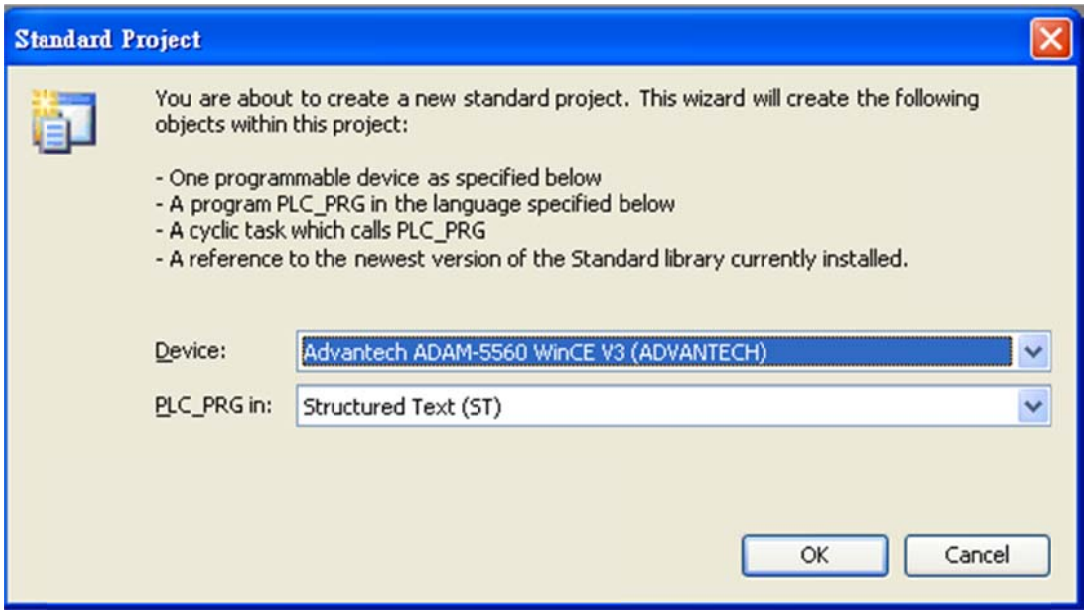
3.2. Create a Project

Step 1: To create a new project, choose command **New project** from the **File** menu:

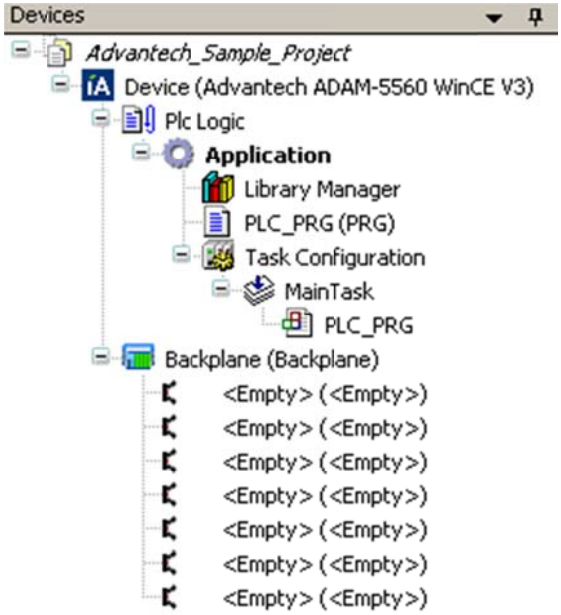
In the New Project dialog select **Standard project** in the 'Templates' field and enter a Name and a Location path for the project file. Press OK to confirm.



Step 2: You will then be prompted for choosing devices. Choose device **Advantech ADAM-5560 WinCE V3 (ADVANTECH)** and programming language **Structured Text (ST)** (depend on developer) for PLC_PRG. Press OK to open the new project.



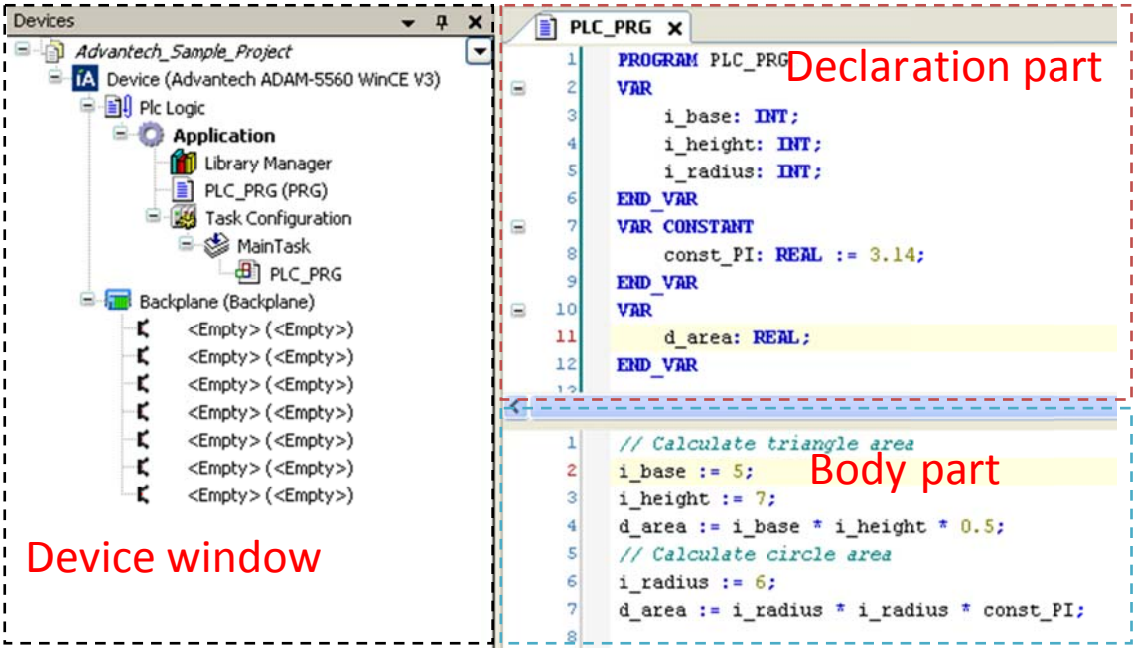
Step 3: The project name now will appear in the title bar of the CoDeSys user interface and the Devices window.



3.3. Write a Program

In the **Devices** window, double-chick **PLC_PRG(PRG)** and language editor window will open. The editor consists of a declaration part (upper) and a body part (lower), separated by a

screen divider. The declaration part shows line numbers at the left border and the embracing keywords "VAR" and "END_VAR" for the variables declaration.



In the declaration part of the editor put the cursor behind VAR and press the Return-key. A new empty line will be displayed where you enter the declaration of variables.

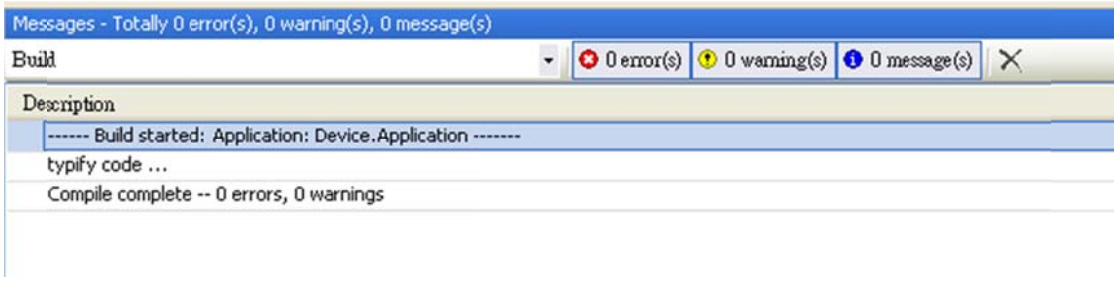
Here, we write a simple program to calculate the area of triangle and circle, so declare **i_base, i_height, i_radius** which are of type INTEGER, **d_area** of type REAL:

```
PROGRAM PLC_PRG
VAR
    i_base: INT;
    i_height: INT;
    i_radius: INT;
END_VAR
VAR CONSTANT
    const_PI: REAL := 3.14;
END_VAR
VAR
    d_area: REAL;
END_VAR
```

In the body part of the PLC_PRG editor put the cursor in line 1 and enter the following lines:

```
// Calculate triangle area
i_base := 5;
i_height := 7;
d_area := i_base * i_height * 0.5;
// Calculate circle area
i_radius := 6;
d_area := i_radius * i_radius * const_PI;
```

We need to check the program for syntactic errors and perform command **Build**  from the menu (**Build -> Build**) or press **<F11>**:



Note!

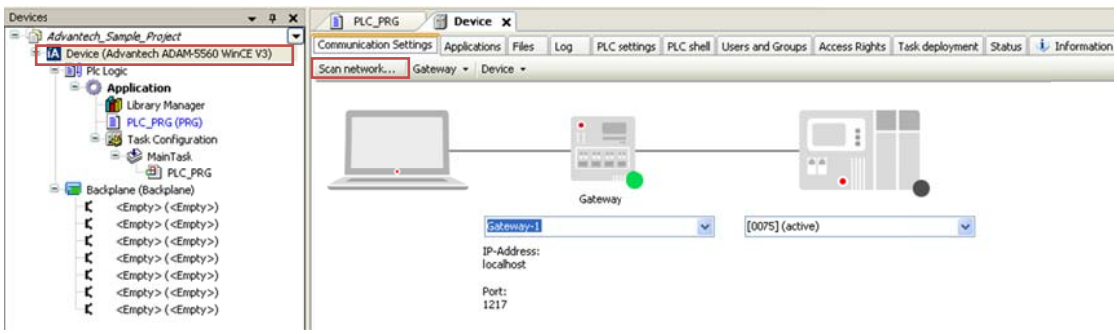
No code will be generated in this case. Error messages will be displayed in the Messages window which is placed at the lower part of the user interface per default.

3.4. Connect to the Target Device

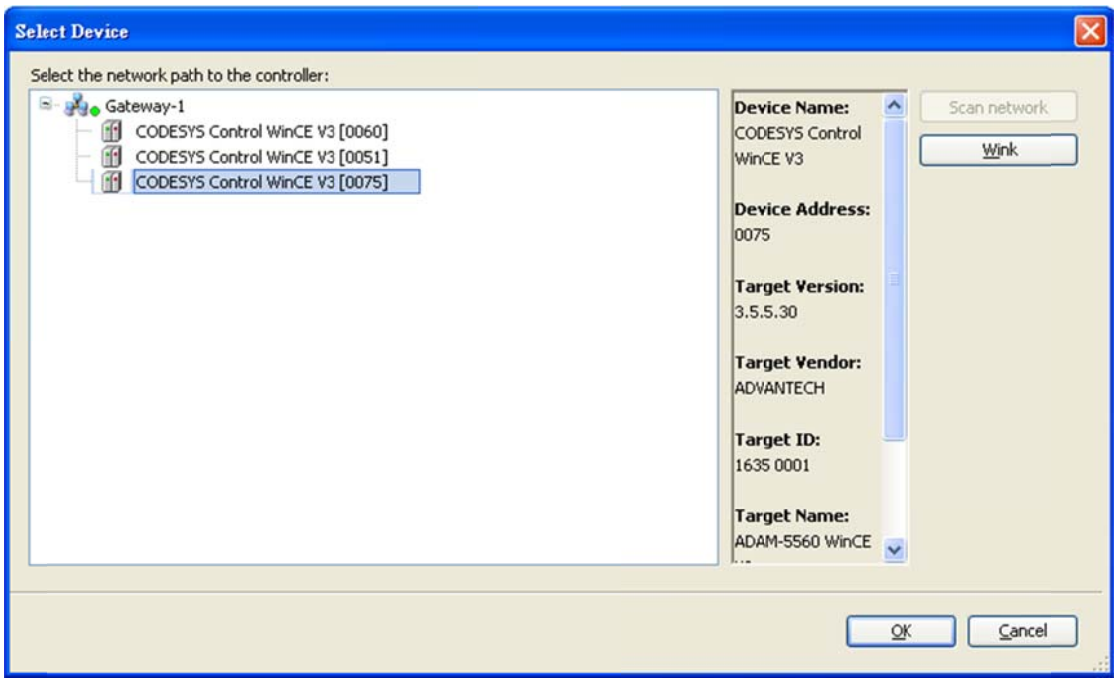
In this section, we want to discuss how to connect to ADAM-5560.

We need to set the active application by using Device editor. It displays an icon of programming device, the current gateway and the target device with their connection status.

The **Device editor** opens by double clicking the device name in the device tree.



Click the **Scan network** button to search for available devices in your local network. You will then be prompted for the device selection. Choose your target device and click OK to proceed.



In Device editor, it will show the connection status. Please check that the colored status points are all in green.

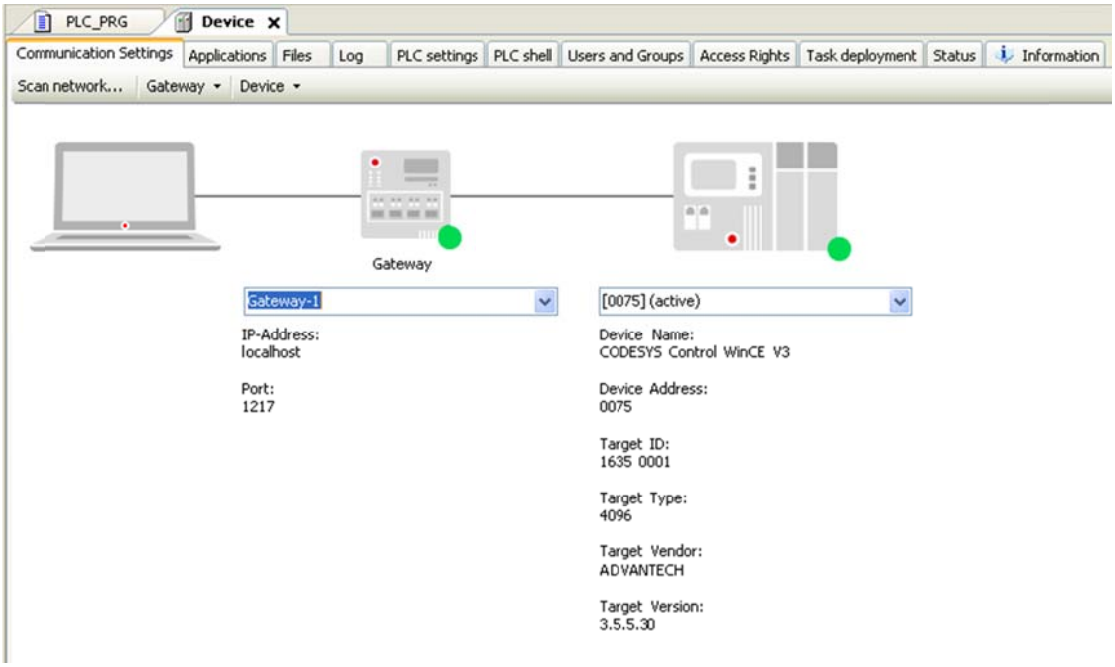
Note!

Meaning of the colored status point on the gateway and the device:

Red: Connection cannot be established

Green: Connection established

Black: Connection not defined



3.5. Run the Application

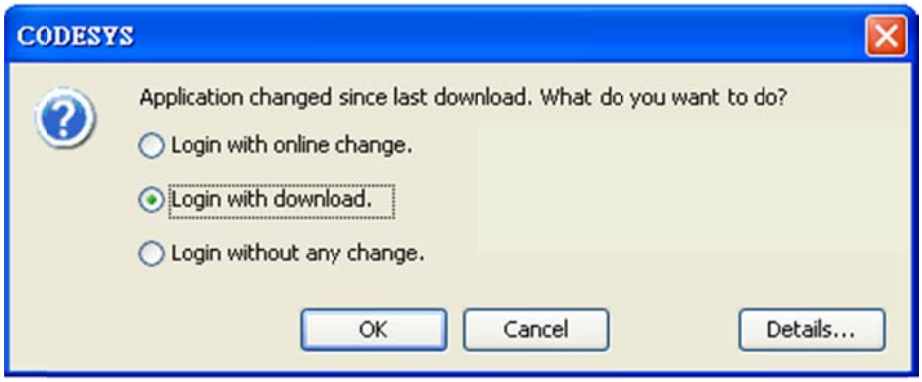
We can download the application by performing command **Login**  from the menu (**Online -> Login**) or press **<Alt+F8>**. You will then be prompted for choosing login options. Here, we choose “Login with download” for the first time and click OK to proceed.

Note!

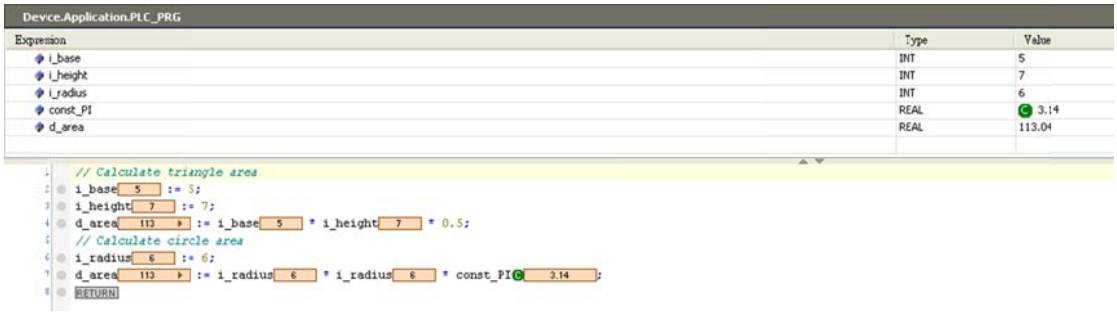
Meaning of login options:

“Login with online change”: Only the modified objects will be loaded.

“Login with download”: The complete application will be loaded and initialized completely. “Login without any change”: The latest modifications will not be loaded.



We run the program by performing command **Start** from the menu **(Debug -> Start)** or press **<F5>**. The online view of PLC_PRG will be opened: In the upper part a table shows the watch variables in application. In the lower part you see the code lines as entered in offline mode, supplemented by the little inline monitoring windows behind each variable, showing the actual value.



Stop the program by performing command **Stop** from the menu (**Debug -> Stop**) or press **<Shift+F8>**.

If you want to change into the offline mode and disconnect the programming system from the target device, perform command **Logout** from the menu (**Online -> Logout**) or press **<Ctrl+F8>**.

Chapter 4

4. Advantech I/O Modules

4.1. Insert I/O Modules into CODESYS

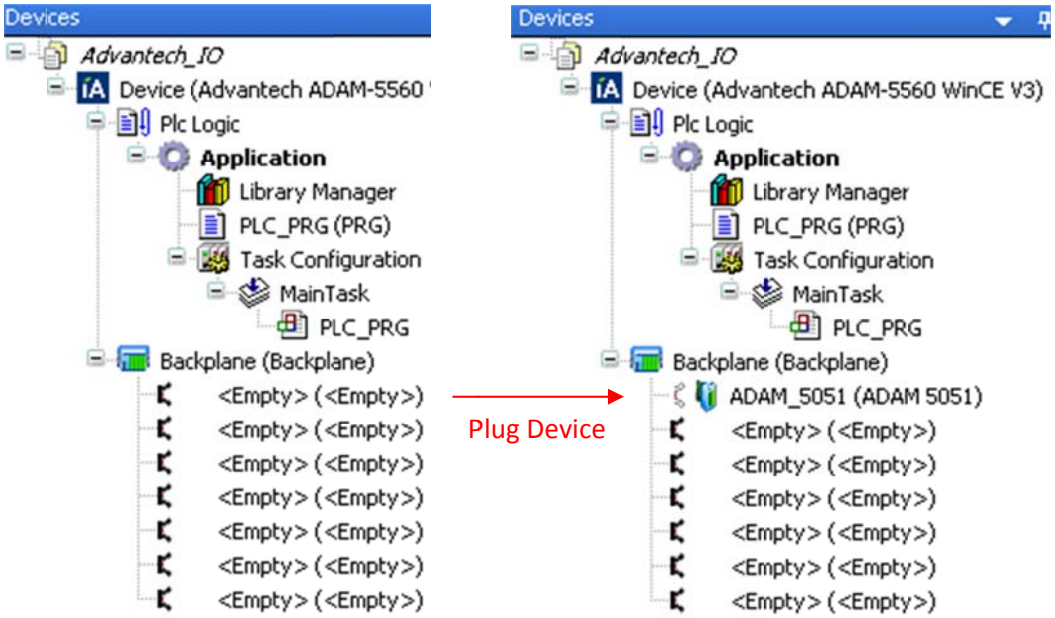
We can add and configure Advantech I/O modules as objects in the device tree.

Note!

ADAM-5560 does NOT support hot plugging for I/O modules.

An empty slot is identified by icon and entry <Empty> (<Empty>). An already occupied slot shows icon and the device name.

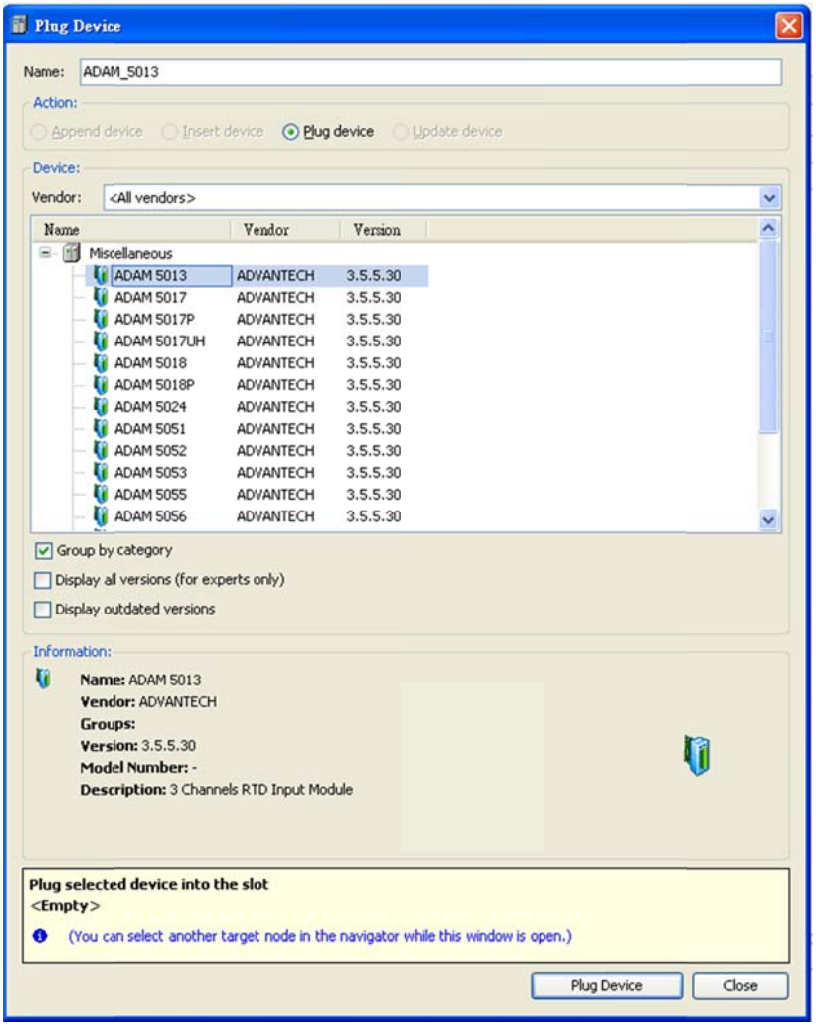
Choose one of available slot and click **Plug Device** in context menu.



It will open the **Plug Device dialog**, where you can choose one of available devices for the current slot. The existing entry will be replaced by the new one in an occupied slot.
Click **Plug Device** to proceed and then press **Close** to close the device dialog.

Note!

You can remove the existing device by click **Delete** in context menu.



4.2. Map Variables to I/O Modules

In this section, we want to discuss how to map variable of program to Advantech I/O modules. For more details on creating a new program please refer to chapter 3.

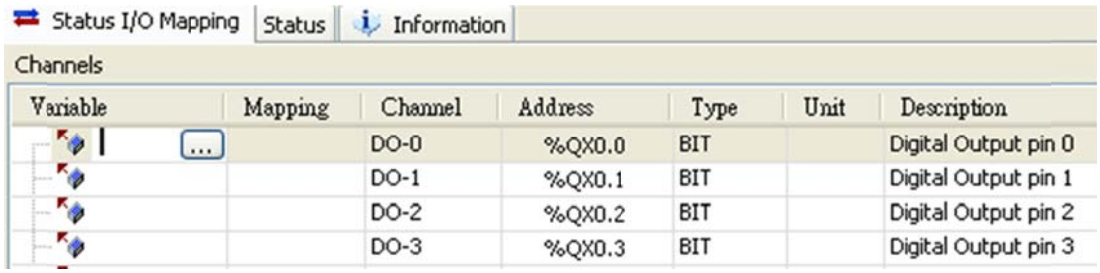
Here, we declare *bValue* in declaration part and set true in body part.

```
PROGRAM PLC_PRG
VAR
    bValue: BOOL;
END_VAR
```

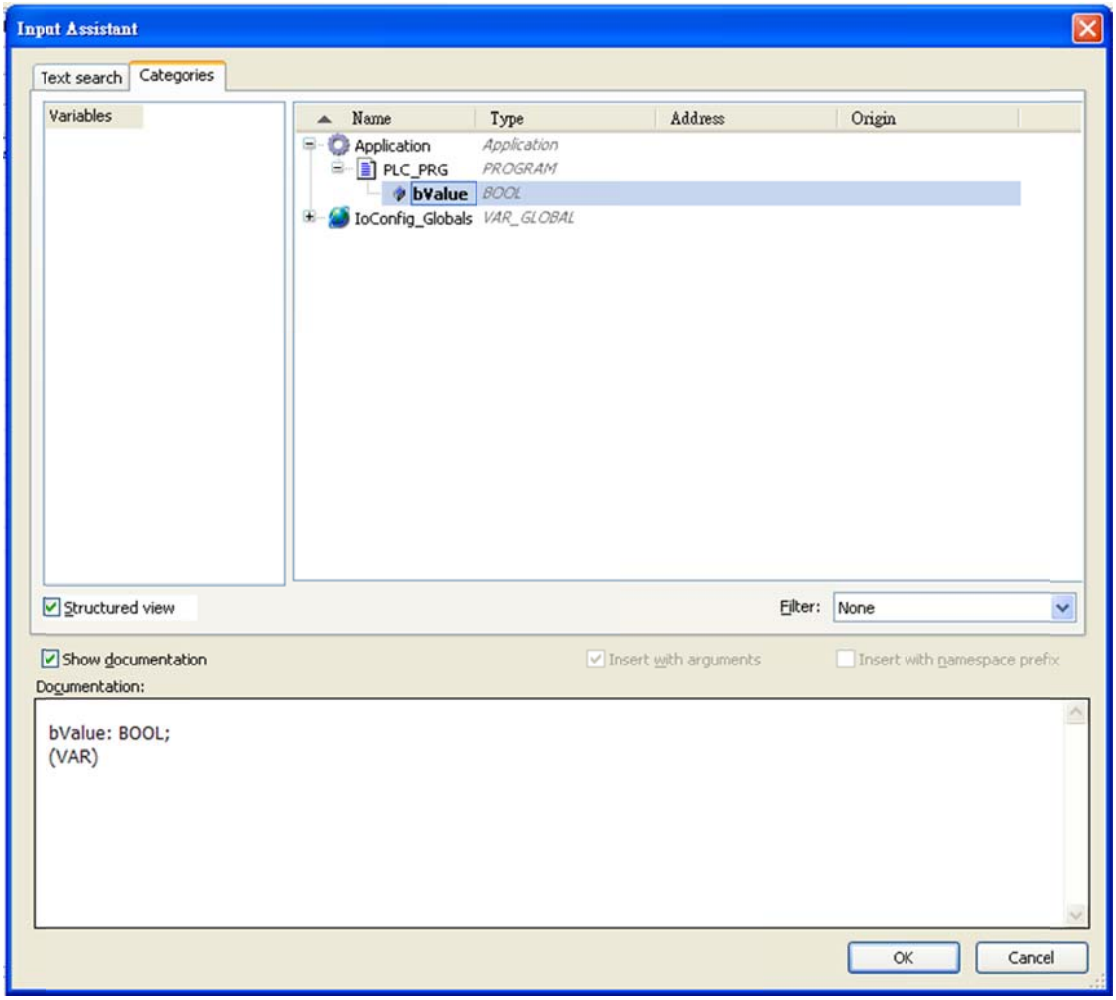
```
bValue:= true;
```

Open **Module Editor** by double clicking the device name in the device tree. Double-click on the variable column and choose mapping variable by clicking the button ... . In this example,

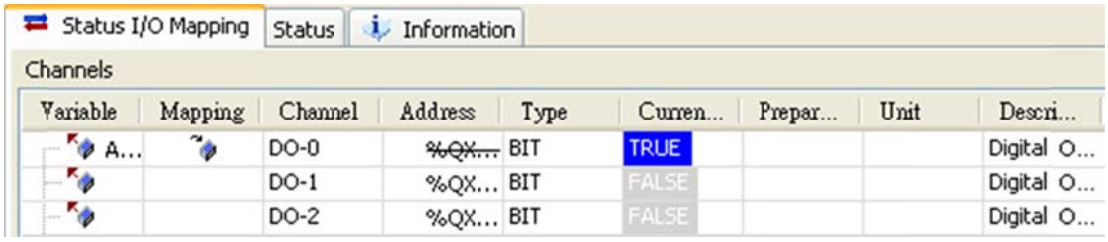
we try to map the variable (*bValue*) to channel 0, so we double-click on the first row of variable column.



It will open the **Input Assistant Dialog**, where you can choose one of available devices for the current slot. The existing entry will be replaced by the new one in an occupied slot. Click **Plug Device** to proceed.

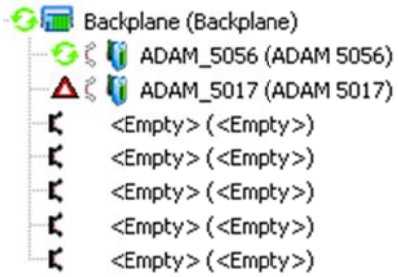


Now, we can download the application by performing command **Login** and then performing command **Start**. The channel-0 of I/O module will be lit up.



Note!

If the Advantech modules are correctly configured, it will show a green circle icon next to the device name in the device tree. If it shows a red triangle, see the chapter 5 for troubleshooting.



4.3. Support List

Advantech provides 21 types of ADAM-5000 I/O modules for various applications so far. Following table is the I/O modules support list. In the following section, we will introduce I/O modules according to their types.

Module	Name	Specification	Reference
Analog Input	ADAM-5013	3-ch RTD Input	Isolated
	ADAM-5017	8-ch AI	Isolated
	ADAM-5017P	8-ch AI w/Independent Input Range	Isolated
	ADAM-5017UH	8-ch Ultra High Speed AI (200KHz)	Isolated
	ADAM-5018	7-ch TC Input	Isolated
	ADAM-5018P	7-ch TC Input w/Independent Input Range	Isolated
Analog Output	ADAM-5024	4-ch AO	Isolated
	ADAM-5051	16-ch DI	Non-isolated

Digital Input	ADAM-5051D	16-ch DI w/LED	Non-isolated
	ADAM-5051S	16-ch Isolated DI w/LED	Isolated
	ADAM-5052	8-ch Isolated DI w/LED	Isolated
	ADAM-5053	32-ch Isolated DI	Isolated
Digital Output	ADAM-5056	16-ch DO	Non-isolated
	ADAM-5056D	16-ch DO w/LED	Non-isolated
	ADAM-5056S	16-ch Isolated DO w/LED	Isolated
	ADAM-5056SO	16-ch Source Type Isolated DO w/LED	Isolated
	ADAM-5057S	32-ch Isolated DO	Isolated
Digital I/O	ADAM-5055S	16-ch Isolated DI/O w/LED	Isolated
Relay Output	ADAM-5060	6-ch Relay Output	Isolated
	ADAM-5069	8-ch Power Relay Output w/LED	Isolated
Counter/ Frequency	ADAM-5081	4-ch/8-ch High Speed Counter/Frequency	Isolated

4.4. Digital Input Modules

In this section, we are going to introduce digital input modules.

The **Module editor** opens by double clicking the device name in the device tree. It consists of three tab pages, that is, **Status I/O Mapping, Status and Information.**

Status I/O Mapping: Show the I/O mapping status between variable to module channel. It consists of seven columns.

Mapping: The mapping status of each variable.

Note!

There are two categories of variables: **Channel values** and **Error ID**.

Channel values: The data type of each channel is in single bit. If the value is “true”, it means that the channel is on; “false” for off.

For detailed variable mapping information, see chapter 4.2.

Error ID: This variable holds the status of I/O module and its data type is in Word (16 Bits). Get module error ID by mapping the last variable in table. For detailed error ID information, see chapter 5.3.

Address: The starting physical address of the variables for this I/O group. The board shown below has 16 digital inputs. This will require either 16 Boolean addresses or 2 Byte addresses.

Note!

Meaning of address expression:

% = Directly Mapped variable

I = Physical Input

X = Single bit

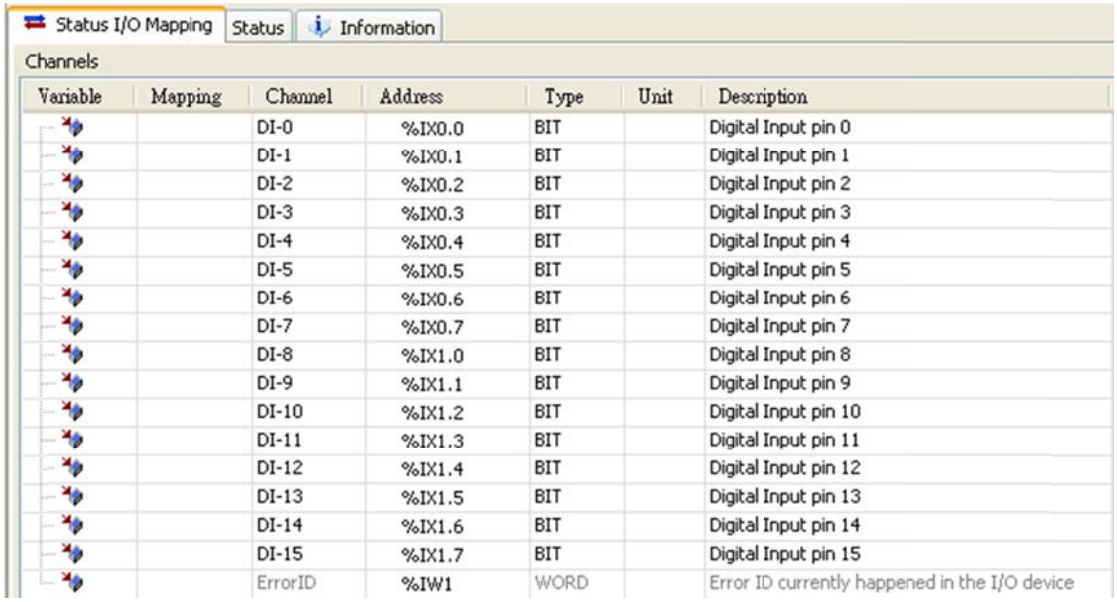
$(N1). $(N2) = The starting address. The first number means the starting byte; the second number means the starting bit.

Type: The data type of each variable.

Description: The description of each variable.

Status: The reserved page.

Information: Provide the brief information to current module.



Variable	Mapping	Channel	Address	Type	Unit	Description
		DI-0	%IX0.0	BIT		Digital Input pin 0
		DI-1	%IX0.1	BIT		Digital Input pin 1
		DI-2	%IX0.2	BIT		Digital Input pin 2
		DI-3	%IX0.3	BIT		Digital Input pin 3
		DI-4	%IX0.4	BIT		Digital Input pin 4
		DI-5	%IX0.5	BIT		Digital Input pin 5
		DI-6	%IX0.6	BIT		Digital Input pin 6
		DI-7	%IX0.7	BIT		Digital Input pin 7
		DI-8	%IX1.0	BIT		Digital Input pin 8
		DI-9	%IX1.1	BIT		Digital Input pin 9
		DI-10	%IX1.2	BIT		Digital Input pin 10
		DI-11	%IX1.3	BIT		Digital Input pin 11
		DI-12	%IX1.4	BIT		Digital Input pin 12
		DI-13	%IX1.5	BIT		Digital Input pin 13
		DI-14	%IX1.6	BIT		Digital Input pin 14
		DI-15	%IX1.7	BIT		Digital Input pin 15
		ErrorID	%IW1	WORD		Error ID currently happened in the I/O device

4.5. Digital Output Modules

In this section, we are going to introduce digital output modules.

The **Module editor** opens by double clicking the device name in the device tree. It consists of three tab pages, that is, **Status I/O Mapping, Status and Information.**

Status I/O Mapping: Show the I/O mapping status between variable to module channel. It consists of seven columns.

Mapping: The mapping status of each variable.

Note!

There are two categories of variables: **Channel values** and **Error ID**.

Channel values: The data type of each channel is in single bit. Set the value to "true" for switching on the channel; "false" for switching off.

For detailed variable mapping information, see chapter 4.2.

Error ID: This variable holds the status of I/O module and its data type is in Word (16 Bits). Get module error ID by mapping the last variable in table. For detailed error ID information, see chapter 5.3.

Address: The starting physical address of the variables for this I/O group. The board shown below has 16 digital inputs. This will require either 16 Boolean addresses or 2 Byte addresses.

Note!

Meaning of address expression:

% = Directly Mapped variable

Q = Physical Output

X = Single bit

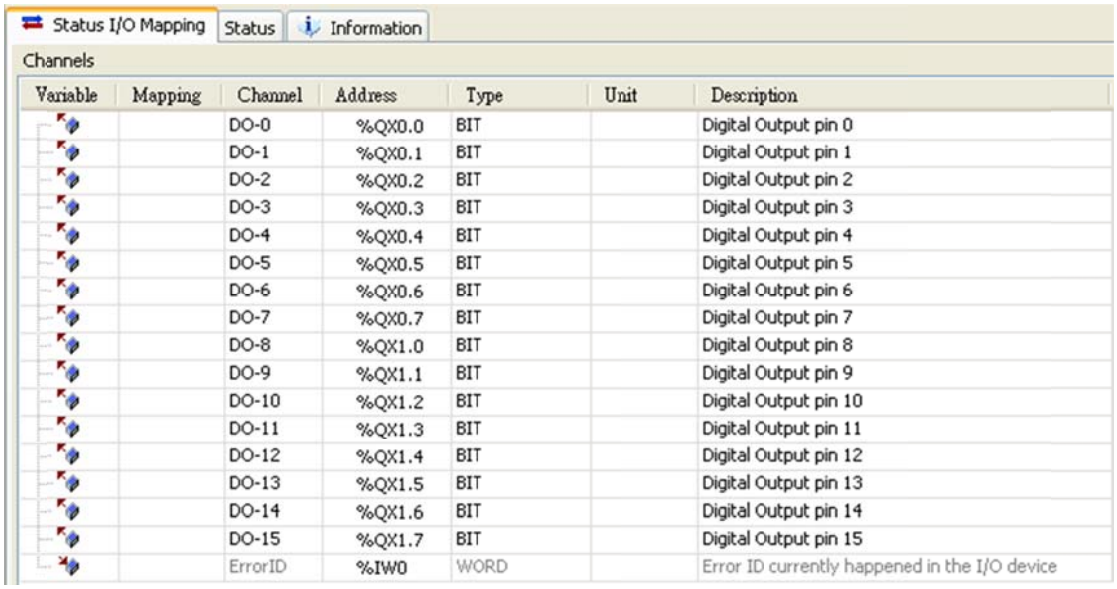
$(N1). $(N2) = The starting address. The first number means the starting byte; the second number means the starting bit.

Type: The data type of each variable.

Description: The description of each variable.

Status: The reserved page.

Information: Provide the brief information to current module.



Variable	Mapping	Channel	Address	Type	Unit	Description
		DO-0	%QX0.0	BIT		Digital Output pin 0
		DO-1	%QX0.1	BIT		Digital Output pin 1
		DO-2	%QX0.2	BIT		Digital Output pin 2
		DO-3	%QX0.3	BIT		Digital Output pin 3
		DO-4	%QX0.4	BIT		Digital Output pin 4
		DO-5	%QX0.5	BIT		Digital Output pin 5
		DO-6	%QX0.6	BIT		Digital Output pin 6
		DO-7	%QX0.7	BIT		Digital Output pin 7
		DO-8	%QX1.0	BIT		Digital Output pin 8
		DO-9	%QX1.1	BIT		Digital Output pin 9
		DO-10	%QX1.2	BIT		Digital Output pin 10
		DO-11	%QX1.3	BIT		Digital Output pin 11
		DO-12	%QX1.4	BIT		Digital Output pin 12
		DO-13	%QX1.5	BIT		Digital Output pin 13
		DO-14	%QX1.6	BIT		Digital Output pin 14
		DO-15	%QX1.7	BIT		Digital Output pin 15
		ErrorID	%IW0	WORD		Error ID currently happened in the I/O device

4.6. Analog Input Modules

In this section, we are going to introduce analog input modules.

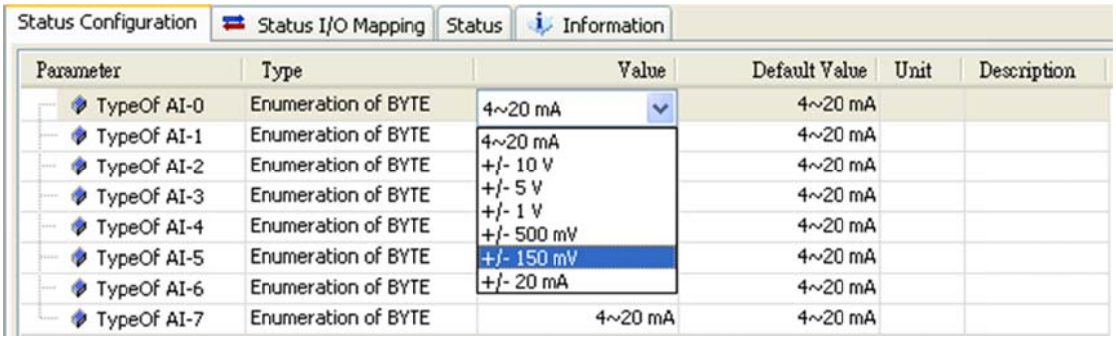
The **Module editor** opens by double clicking the device name in the device tree. It consists of four tab pages, that is, **Status Configuration**, **Status I/O Mapping, Status and Information.**

Status Configuration: Provide the channel status page for setting channel ranges.

Double-click on the value column of the particular channel.

Note!

For ADAM-5013, ADAM-5017 and ADAM-5018, all channels are restricted to the same channel range and base on the first channel (ch-0).



Status I/O Mapping: Show the I/O mapping status between variable to module channel.

Mapping: The mapping status of each variable.

Note!

There are two categories of variables: **Channel values** and **Error ID**.

Channel values: The data type of each channel is in REAL.

For detailed variable mapping information, see chapter 4.2.

Error ID: This variable holds the status of I/O module and its data type is in Word (16 Bits). Get module error ID by mapping the last variable in table. For detailed error ID information, see chapter 5.3.

Address: The starting physical address of the variables for this I/O group. The board shown below has 8 analog inputs. This will require 8 DWORD addresses.

Note!

Meaning of address expression:

% = Directly Mapped variable

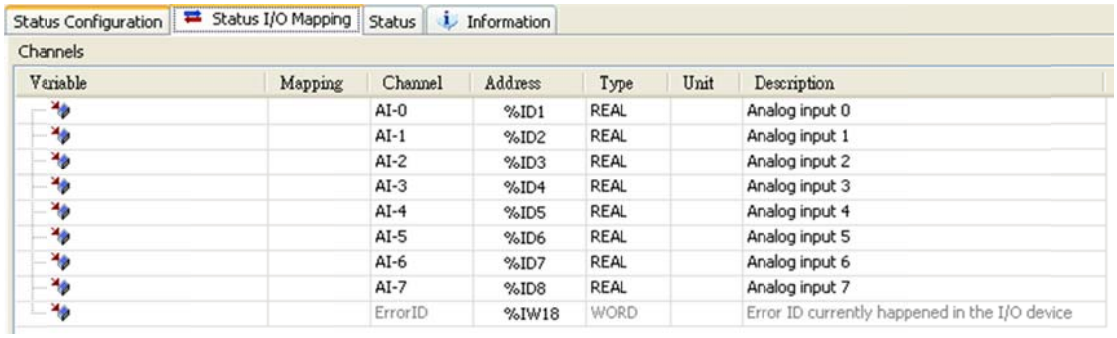
I = Physical Input

D = Double word (32 Bits)

$(N) = The starting address.

Type: The data type of each variable.

Description: The description of each variable.



Variable	Mapping	Channel	Address	Type	Unit	Description
		AI-0	%ID1	REAL		Analog input 0
		AI-1	%ID2	REAL		Analog input 1
		AI-2	%ID3	REAL		Analog input 2
		AI-3	%ID4	REAL		Analog input 3
		AI-4	%ID5	REAL		Analog input 4
		AI-5	%ID6	REAL		Analog input 5
		AI-6	%ID7	REAL		Analog input 6
		AI-7	%ID8	REAL		Analog input 7
		ErrorID	%IW18	WORD		Error ID currently happened in the I/O device

Status: The reserved page.

Information: Provide the brief information for current module.

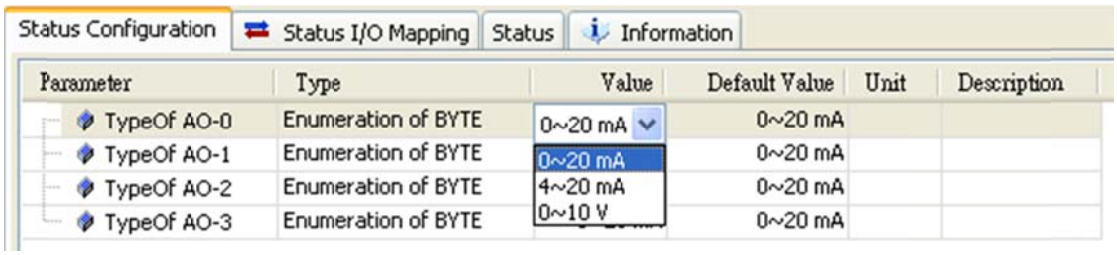
4.7. Analog Output Modules

In this section, we are going to introduce analog output modules.

The **Module editor** opens by double clicking the device name in the device tree. It consists of four tab pages, that is, **Status Configuration**, **Status I/O Mapping, Status and Information.**

Status Configuration: Provide the channel status page for setting channel ranges.

Double-click on the value column of the particular channel.



Status I/O Mapping: Show the I/O mapping status between local variable to module channel.

Mapping: The mapping status of each variable.

Note!

There are two categories of variables: **Channel values** and **Error ID**.

Channel values: The data type of each channel is in REAL.

For detailed variable mapping information, see chapter 4.2.

Error ID: This variable holds the status of I/O module and its data type is in Word (16 Bits). Get module error ID by mapping the last variable in table. For detailed error ID information, see chapter 5.3.

Address: The starting physical address of the variables for this I/O group. The board shown below has 4 analog outputs. This will require 4 DWORD addresses.

Note!

Meaning of address expression:

% = Directly Mapped variable

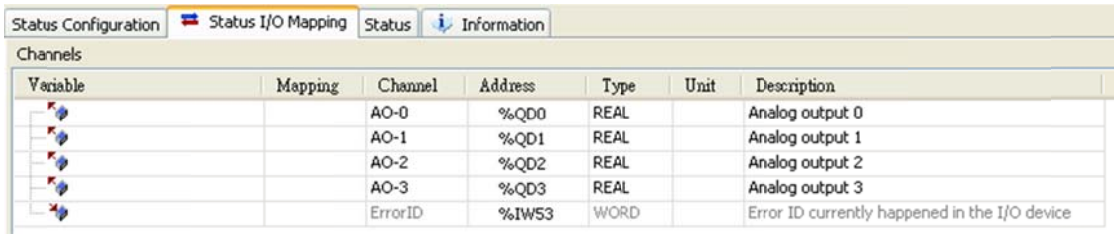
Q = Physical Output

D = Double word (32 Bits)

$(N) = The starting address.

Type: The data type of each variable.

Description: The description of each variable.



Status: The reserved page.

Information: Provide the brief information for current module.

4.8. Relay Output Modules

In this section, we are going to introduce relay output modules.

The **Module editor** opens by double clicking the device name in the device tree. It consists of three tab pages, that is, **Status I/O Mapping, Status and Information.**

Status I/O Mapping: Show the I/O mapping status between local variable to module channel. It consists of seven columns.

Mapping: The mapping status of each variable.

Note!

There are two categories of variables: **Channel values** and **Error ID**.

Channel values: The data type of each channel is in single bit. Set the value to "true" for switching on the channel; "false" for switching off.

For detailed variable mapping information, see chapter 4.2.

Error ID: This variable holds the status of I/O module and its data type is in Word (16 Bits). Get module error ID by mapping the last variable in table. For detailed error ID information, see chapter 5.3.

Address: The starting physical address of the variables for this I/O group. The board shown below has 6 relay outputs. This will require either 6 Boolean addresses or 1 Byte address.

Note!

Meaning of address expression:

% = Directly Mapped variable

Q = Physical Output

X = Single bit

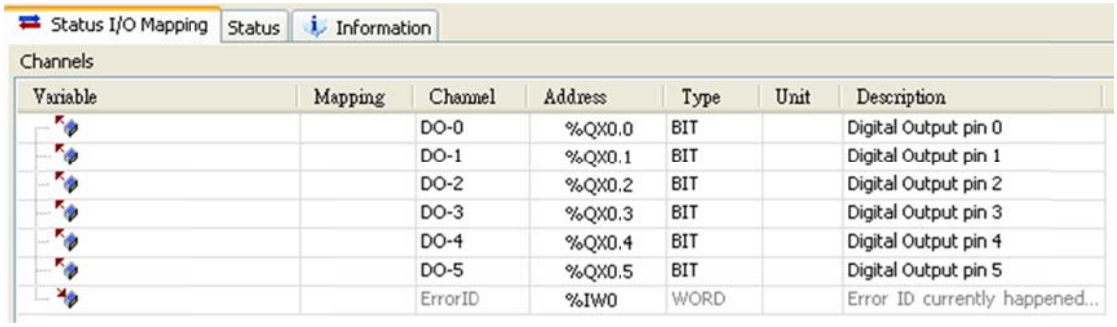
$(N1). $(N2) = The starting address. The first number means the starting byte; the second number means the starting bit.

Type: The data type of each variable.

Description: The description of each variable.

Status: The reserved page.

Information: Provide the brief information to current module.



4.9. Counter/Frequency Modules

In this section, we are going to introduce counter/frequency modules.

The **Module editor** opens by double clicking the device name in the device tree. It consists of four tab pages, that is, **Status Configuration**, **Status I/O Mapping, Status and Information.**

Status Configuration: Provide the channel status page for setting channel ranges.

Double-click on the value column of the particular channel. The module contains counter channels and DO/Alarm channels as shown image below.

For counter channels, we offer five types of counting mode (Bi-direction, Up/Down, A/B Phase) for different application purposes. The counter will count up or down according to your applications. This counting function helps us obtain the most accurate data.

Before starting counting, we have to set **Startup Value**, i.e. Initial value**,** and the default is 0. The data type of startup value is in DWORD.

Parameter	Type	Value	Default Value	Unit	Description
CTR 0					Channel config of CTR 0
Mode	Enumeration of BYTE	Bi-directory	Bi-directory		
Startup Value	DWORD(16#0..16#FFFFFFFF)		0		
CTR 2					Channel config of CTR 2
Mode	Enumeration of BYTE		Bi-directory		
Startup Value	DWORD(16#0..16#FFFFFFFF)		0		
CTR 4					Channel config of CTR 4
Mode	Enumeration of BYTE	Bi-directory	Bi-directory		
Startup Value	DWORD(16#0..16#FFFFFFFF)	0	0		
CTR 6					Channel config of CTR 6
Mode	Enumeration of BYTE	Bi-directory	Bi-directory		
Startup Value	DWORD(16#0..16#FFFFFFFF)	0	0		

Bi-directory
Up/Down
A/B-1X
A/B-2X
A/B-4X

We can also set **Digital Filter** (in us) for high or low level minimum signal width to reduce noise spike.

Every counter channel can measure the frequency of assigned time window and the range is from 0.1Hz to 1MHz. We can assigned **Frequency Acquisition Time** (in ms) to get counter number within the time range.

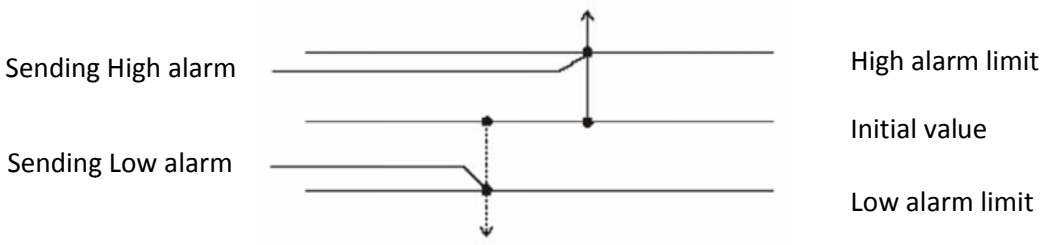
Digital Filter	UINT(1..65000)	1	1	us	Digital Filter
Freq. Acq. Time	UINT(1..1000)	1	1	ms	Freq. Acq. Time

For DO/Alarm channels, you can change the setting by double-clicking and selecting an item from the pull-down menus:

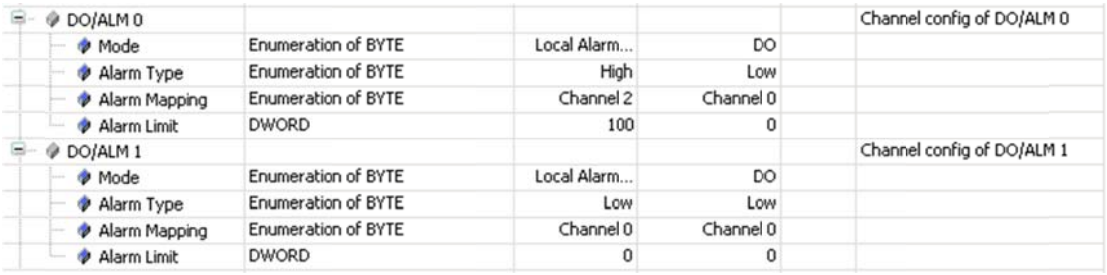
Mode: Select "DO" or "Local Alarm Latch"

Alarm Type: Select "High" or "Low", i.e. high alarm or low alarm.

Alarm Limit: Set alarm limit value.



Alarm Mapping: Select counter channel number.



DO/ALM 0					Channel config of DO/ALM 0
Mode	Enumeration of BYTE	Local Alarm...	DO		
Alarm Type	Enumeration of BYTE	High	Low		
Alarm Mapping	Enumeration of BYTE	Channel 2	Channel 0		
Alarm Limit	DWORD	100	0		
DO/ALM 1					Channel config of DO/ALM 1
Mode	Enumeration of BYTE	Local Alarm...	DO		
Alarm Type	Enumeration of BYTE	Low	Low		
Alarm Mapping	Enumeration of BYTE	Channel 0	Channel 0		
Alarm Limit	DWORD	0	0		

Status I/O Mapping: Show the I/O mapping status between local variable to module channel.

Mapping: The mapping status of each variable. For detailed variable mapping information, see chapter 4.2.

Note!

There are three categories of variables: **Status variables**, **Setting variables** and **Error ID**.

Status variables: These variables are read-only and hold the module status.

Name	Data type	Information
Counting Number	DWORD	The total counting number.
Counting Status	Single Bit	“true” for starting counting; “false” for stop.
Overflow Status	Single Bit	“true” for overflow
Latch Status	Single Bit	“true” for reach alarm latch.

Setting variables: These variables are read-write. You can change the setting by changing the column value.

Name	Data type	Information
DO value	Single Bit	Set the value to “true” for switching on the channel; “false” for switching off.

Clear to Startup	Single Bit	Set “true” for set current counting number to startup counting number.
Clear Overflow	Single Bit	Set “true” for clear overflow.
Clear Alarm Latch	Single Bit	Set “true” for clear the alarm latch.

For detailed variable mapping information, see chapter 4.2.

Error ID: This variable holds the status of I/O module and its data type is in Word (16 Bits). Get module error ID by mapping the last variable in table. For detailed error ID information, see chapter 5.3.

Address: The starting physical address of the variables for this I/O group. The board shown below has 4 counter inputs and 4 digital outputs. This will require 4 DWORD addresses for counter channel values and 4 Boolean addresses for digital outputs.

Note!

Meaning of address expression:

% = Directly Mapped variable

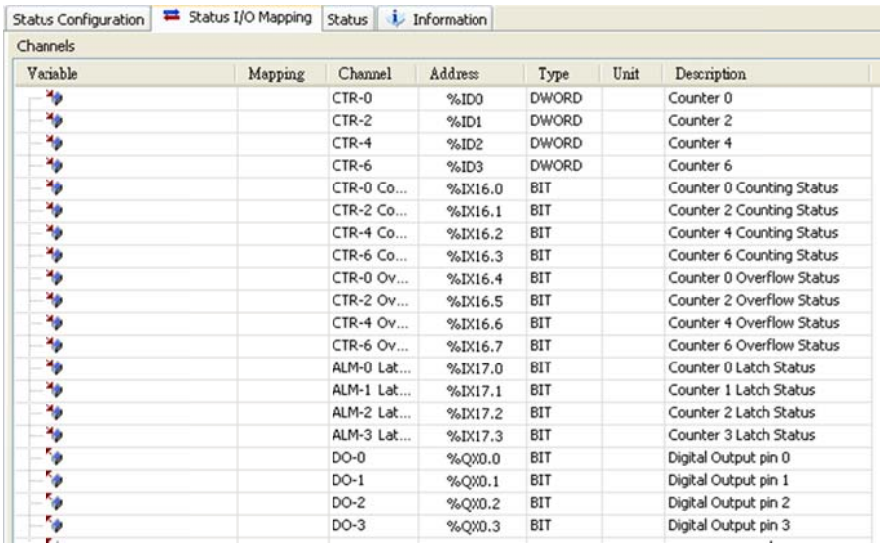
I = Physical Input **Q** = Physical Output

X = Single bit **D** = Double word (32 Bits)

$(N) = The starting address.

Type: The data type of each variable.

Description: The description of each variable.

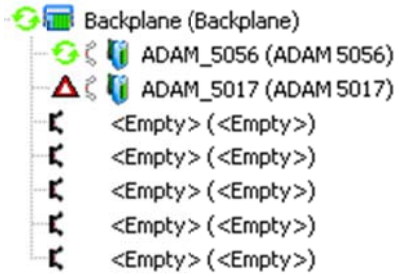

Status Configuration
Status I/O Mapping
Status
Information
Channels
Variable
Mapping
Channel
Address
Type
Unit
Description
CTR-0 %ID0 DWORD Counter 0
CTR-2 %ID1 DWORD Counter 2
CTR-4 %ID2 DWORD Counter 4
CTR-6 %ID3 DWORD Counter 6
CTR-0 Co... %IX16.0 BIT Counter 0 Counting Status
CTR-2 Co... %IX16.1 BIT Counter 2 Counting Status
CTR-4 Co... %IX16.2 BIT Counter 4 Counting Status
CTR-6 Co... %IX16.3 BIT Counter 6 Counting Status
CTR-0 Ov... %IX16.4 BIT Counter 0 Overflow Status
CTR-2 Ov... %IX16.5 BIT Counter 2 Overflow Status
CTR-4 Ov... %IX16.6 BIT Counter 4 Overflow Status
CTR-6 Ov... %IX16.7 BIT Counter 6 Overflow Status
ALM-0 Lat... %IX17.0 BIT Counter 0 Latch Status
ALM-1 Lat... %IX17.1 BIT Counter 1 Latch Status
ALM-2 Lat... %IX17.2 BIT Counter 2 Latch Status
ALM-3 Lat... %IX17.3 BIT Counter 3 Latch Status
DO-0 %QX0.0 BIT Digital Output pin 0
DO-1 %QX0.1 BIT Digital Output pin 1
DO-2 %QX0.2 BIT Digital Output pin 2
DO-3 %QX0.3 BIT Digital Output pin 3

Chapter 5

5. Diagnosis and Troubleshooting

5.1. Error Notification

In chapter 4, we introduce how to write a program to control Advantech I/O modules. If the Advantech modules are correctly configured, it will show a green circle icon next to the device name in the device tree after performing command **Login** and **Start**. If it shows a red triangle , it means that I/O module encountered several errors while running.



5.2. Log Information

We can get log information from **Advantech CoDeSys** or **target machine**, i.e. ADAM-5560.

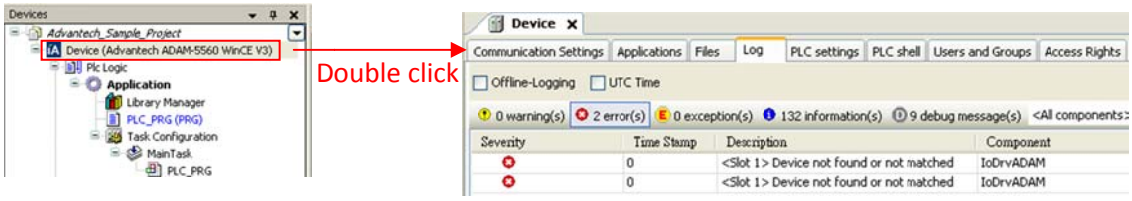
In Advantech CoDeSys development environment, double click the device name in the device tree to open **Device editor**. Select the **Log** dialog and it will display the log of the Advantech I/O module. A log entry line contains the following information:

Severity: There are four categories: warnings, errors, exceptions, information. The display of the entries of each category can be switched on or off by using the corresponding button from the bar above the listing. Each button always contains the current number of loggings in the respective category.

Time Stamp: Date and Time.

Description: Description of the event, for example “Device not found or not matched."

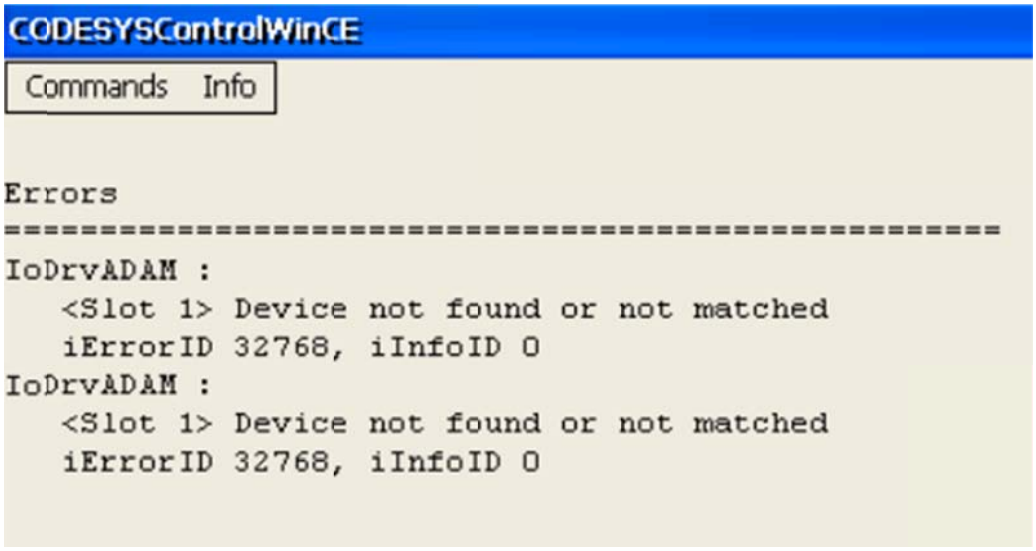
Component: ID and name of the component



On target machine, we can also get error ID from CoDeSys WinCE runtime.

In the ADAM-5560 environment, open runtime by double-clicking the runtime icon which is available on the lower-right corner of the desktop.

Get error ID by performing command Errors from the menu (**Info -> Errors**).



5.3. Error ID

Following table is the error ID for I/O modules.

Error ID	Description
0x8000	The module didn't exist or match the setting module. Make sure that the setting module matches for the device that is being plugged and check your module is plugged in ADAM-5560 appropriately.

0x8001	The system failed to open the module. Please close all programs and reboot. If the system cannot returns to normal condition or the error occurred, please contact Advantech for technical support.
0x8002	The system was unable to complete configuration. Please power-off the system and plug the module again. If the error occurred, please replace a new module and contact Advantech for technical support.
0x8003	The system failed to read value from the module. Please power-off the system and plug the module again. If the error occurred, please replace a new module and contact Advantech for technical support.
0x8004	The system failed to write value to the module. Please power-off the system and plug the module again. If the error occurred, please replace a new module and contact Advantech for technical support.
0x8005	For counter module, the system failed to start/stop counter. Please power-off the system and plug the module again. If the error occurred, please replace a new module and contact Advantech for technical support.
0x8006	For counter module, the system failed to clear counting value. Please power-off the system and plug the module again. If the error occurred, please replace a new module and contact Advantech for technical support.
0x8007	For counter module, the system failed to clear overflow flag. Please power-off the system and plug the module again. If the error occurred, please replace a new module and contact Advantech for technical support.
0x8008	For counter module, the system failed to clear alarm flag. Please power-off the system and plug the module again. If the error occurred,

	please replace a new module and contact Advantech for technical support.