IABU Headquarters

Delta Electronics, Inc.
Taoyuan Technology Center
No.18, Xinglong Rd., Taoyuan City,
Taoyuan County 33068, Taiwan
TEL: 886-3-362-6301 / FAX: 886-3-371-6301

Asia

Delta Electronics (Jiangsu) Ltd.
Wujiang Plant 3
1688 Jiangxing East Road,
Wujiang Economic Development Zone
Wujiang City, Jiang Su Province,
People's Republic of China (Post code: 215200)
TEL: 86-512-6340-3008 / FAX: 86-769-6340-7290

Delta Greentech (China) Co., Ltd.
238 Min-Xia Road, Pudong District,
ShangHai, P.R.C.
Post code : 201209
TEL: 86-21-58635678 / FAX: 86-21-58630003

Delta Electronics (Japan), Inc.
Tokyo Office
2-1-14 Minato-ku Shibadaimon,
Tokyo 105-0012, Japan
TEL: 81-3-5733-1111 / FAX: 81-3-5733-1211

Delta Electronics (Korea), Inc.
1511, Byucksan Digital Valley 6-cha, Gasan-dong,
Geumcheon-gu, Seoul, Korea, 153-704
TEL: 82-2-515-5303 / FAX: 82-2-515-5302

Delta Electronics Int'l (S) Pte Ltd
4 Kaki Bukit Ave 1, #05-05, Singapore 417939
TEL: 65-6747-5155 / FAX: 65-6744-9228

Delta Electronics (India) Pvt. Ltd.
Plot No 43 Sector 35, HSIIDC
Gurgaon, PIN 122001, Haryana, India
TEL : 91-124-4874900 / FAX : 91-124-4874945

Americas

Delta Products Corporation (USA)
Raleigh Office
P.O. Box 12173,5101 Davis Drive,
Research Triangle Park, NC 27709, U.S.A.
TEL: 1-919-767-3800 / FAX: 1-919-767-8080

Delta Greentech (Brasil) S.A
Sao Paulo Office
Rua Itapeva, 26 - 3° andar Edificio Itapeva One-Bela Vista
01332-000-São Paulo-SP-Brazil
TEL: +55 11 3568-3855 / FAX: +55 11 3568-3865

Europe

Deltronics (The Netherlands) B.V.
Eindhoven Office
De Witbogt 15, 5652 AG Eindhoven, The Netherlands
TEL: 31-40-2592850 / FAX: 31-40-2592851

AH-0109320-01

*We reserve the right to change the information in this catalogue without prior notice.

2012-06-15

AH500 Quick Start

www.delta.com.tw/ia

Chapter 1 Introduction

Table of Contents

1.1 Manuals

Delta Electronics, Inc. provides users with the manuals related to the AH500 series programmable logic controllers according to different application requirements.

AH500 Hardware Manual: It introduces hardware specifications, addressing, wiring, maintenance, troubleshooting, and etc.

AH500 Operation Manual: It introduces the configuration of the hardware, the setting of the connection, the operation of the CPU module, the setting of the software, and etc.

AH500 Programming Manual: It introduces devices and instructions.

AH500 Module Manual: It introduces module specifications, installation, setting, troubleshooting, and etc.

AH500 Motion Control Module Manual: It introduces the specifications for the motion control modules, the wiring, the instructions, and the functions.

ISPSoft User Manual: It introduces the use of ISPSoft, including the variables, the connection, the programs, and the function blocks.

PMSoft User Manual: It introduces the use of PMSoft, including the editing mode, the connection, and the encryption.

AH500 Quick Start: It helps users create and use the system in a short time. AH500 Quick Start not only introduces the basic system frameworks, but also teaches users to write a program step by step by means of simple examples, and download the program which includes the variables and the function blocks to the CPU module. Users can experience the convenience brought by the new functions. If an error occurs when the system runs, please refer to section 3.3.2 for more information about debugging the program. (Please refer to chapter 12 in AH500 Operation Manual for more information about the troubleshooting.)

The graphic representations in the manual

🖱 : Clicking the left mouse button

🖱2 : Double-clicking the left mouse button

🖱↕ : Pressing and holding the left mouse button, and then moving the mouse without releasing the button.

⌨ : Typing with a keyboard

① : Operating sequence (The graphic representation is used when the operating sequence is mentioned. For example, 🖱① and ②⌨ .)

❶ : Number used with a picture

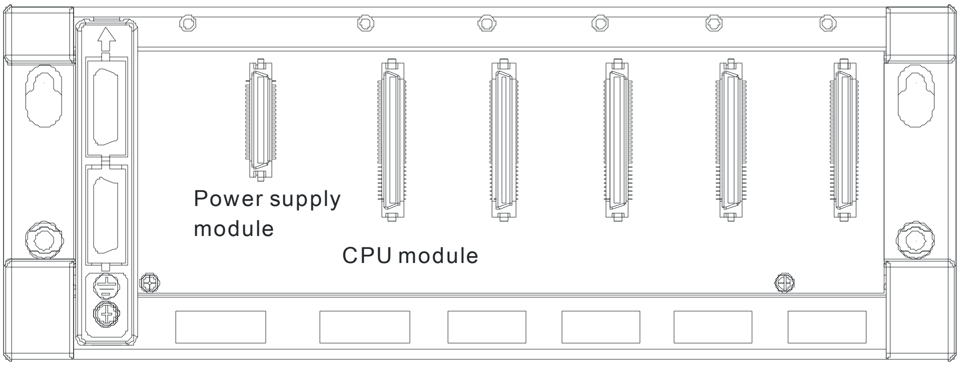
1.2 System Frameworks

The AH500 series programmable logic controller is a medium type of programmable logic control system. The execution speed and the memory capacity are increased. Besides, the complete program development function of function blocks is supported. In order to meet users' more advanced application requirements, the AH500 series programmable logic controllers provide more flexible system extension frameworks. Under such system frameworks, users do not need to use several CPU modules to control the system because of the fact that there are too many I/O points or the equipment is too far away. The completeness of the system is retained, and users can be more efficient in developing the projects.

The minimum framework requirement for the AH500 system:

To create the AH500 system, one main backplane, one power supply module, and one CPU module are needed for the operation of the CPU module.

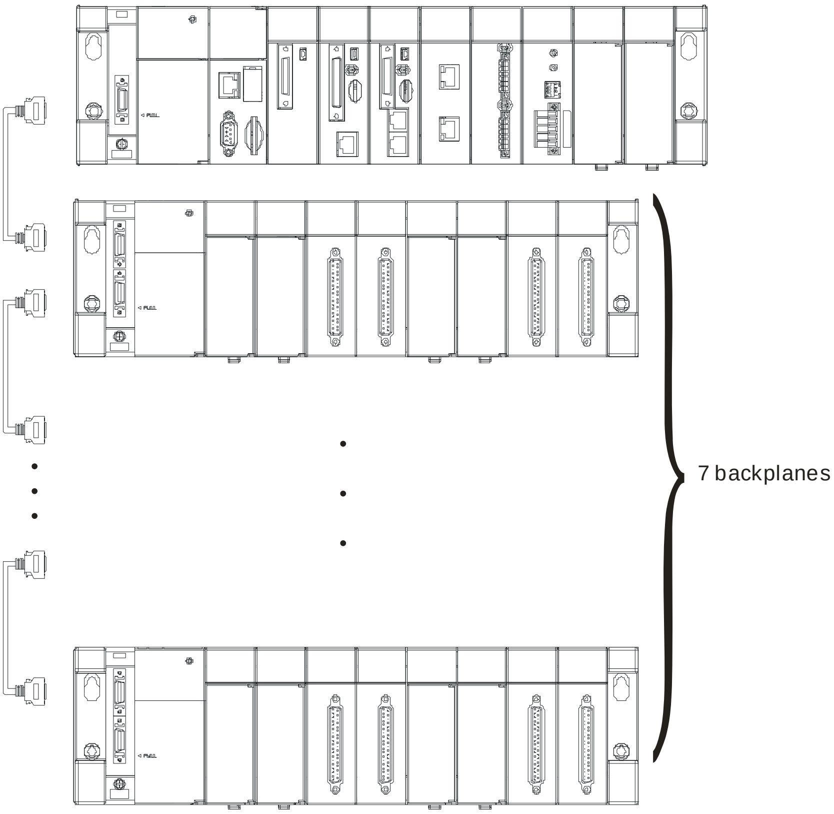
Main backplane (four-slot AHBP04M1-5A)



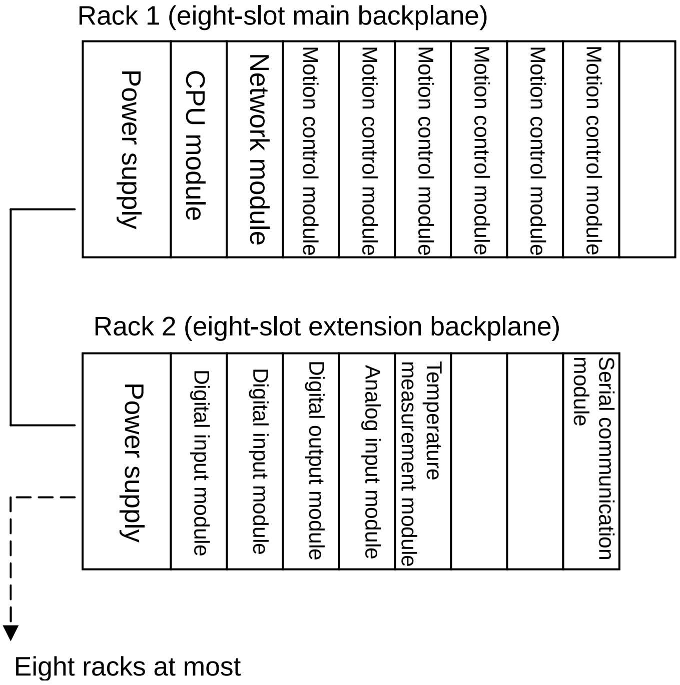
The common framework of the AH500 system (for original equipment manufacturers):

The AH500 system can meet most equipment development requirements in the application field of original equipment manufacturers. Generally speaking, one eight-slot main backplane or one twelve-slot main backplane is chosen. Some advanced equipment can be used with a six-slot extension backplane or an eight-slot extension backplane under the original framework to increase the number of I/O points and the number of axes, or decrease the wiring cost.

The framework: Rack 1~rack 8



The configuration is as follows.

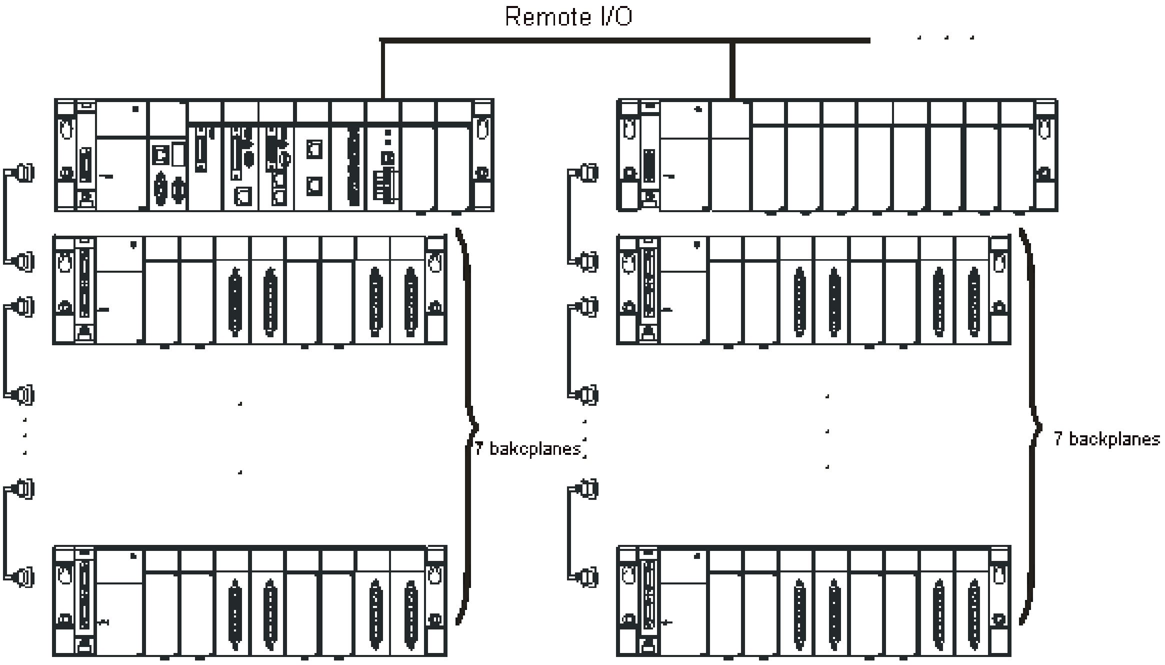


Note: Owing to the consideration to the data transmission speed, the motion control modules and the network modules (exclusive of the serial communication modules) have to be installed with the CPU module on the same backplane. Otherwise, the system can not operate properly.

The common framework of the AH500 system (for system integration):

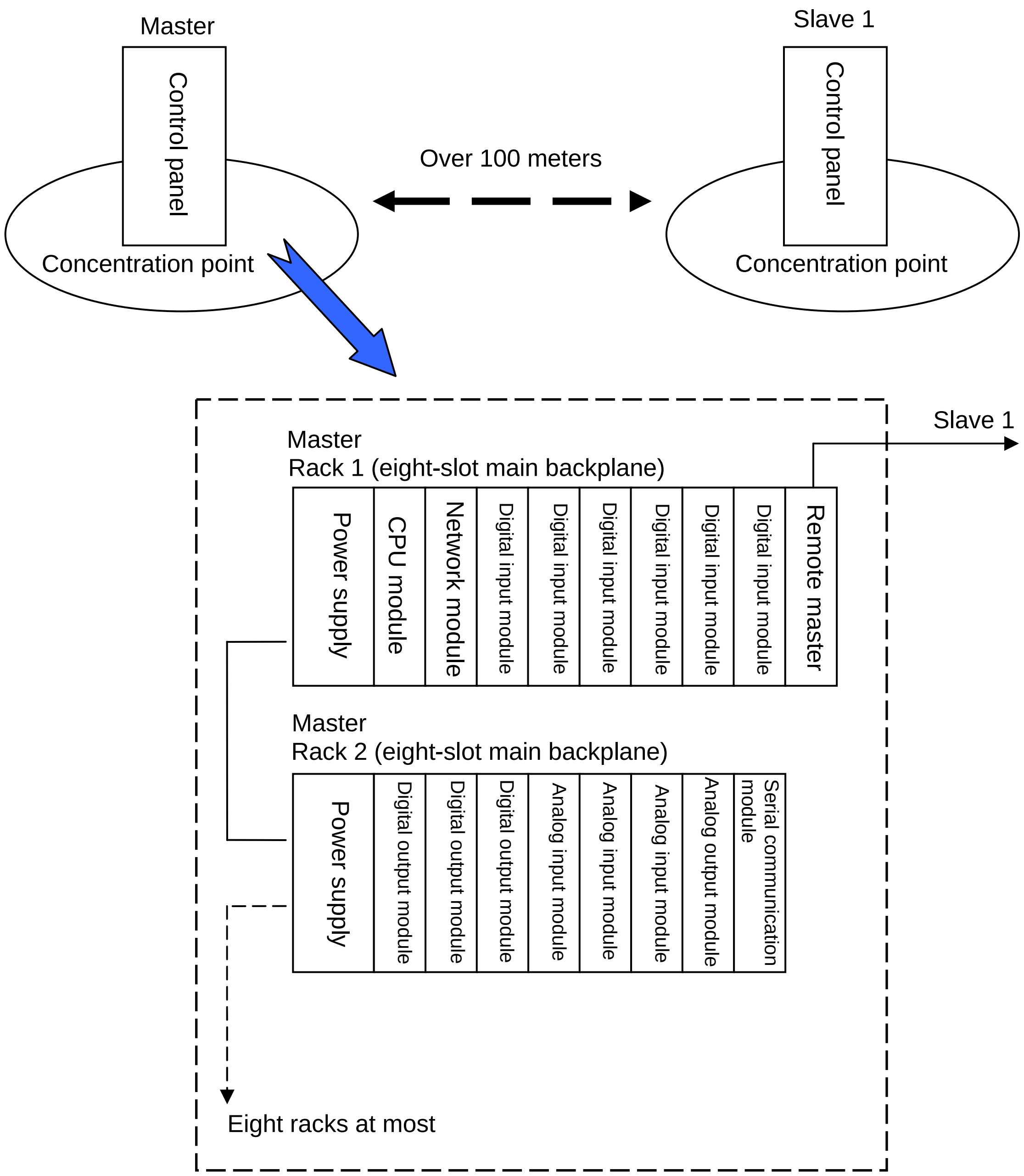
The AH500 system can meet most system control requirements in the application field of the system integration. Generally speaking, the system framework is related to the positions of the equipments. A control panel is usually placed among the equipments which are concentrated to save the wiring cost. However, there is usually more than one concentration point, and the distance between the concentration points is over 100 meters. If it is necessary for users to place more than two control panels, the RTU modules are required. The remote framework of the AH500 system not only fills the remote requirement, but also is combined with the extension of the local I/O. Under most conditions, a control panel instead of a backplane can be regarded as a remote I/O station. In addition, the backplanes, the power supply modules, and other modules are compatible with one another. The convenience of planning the system is increased, and the difficulty of choosing the products is also decreased.

System framework: Rack 1~rack 8 + RTU modules

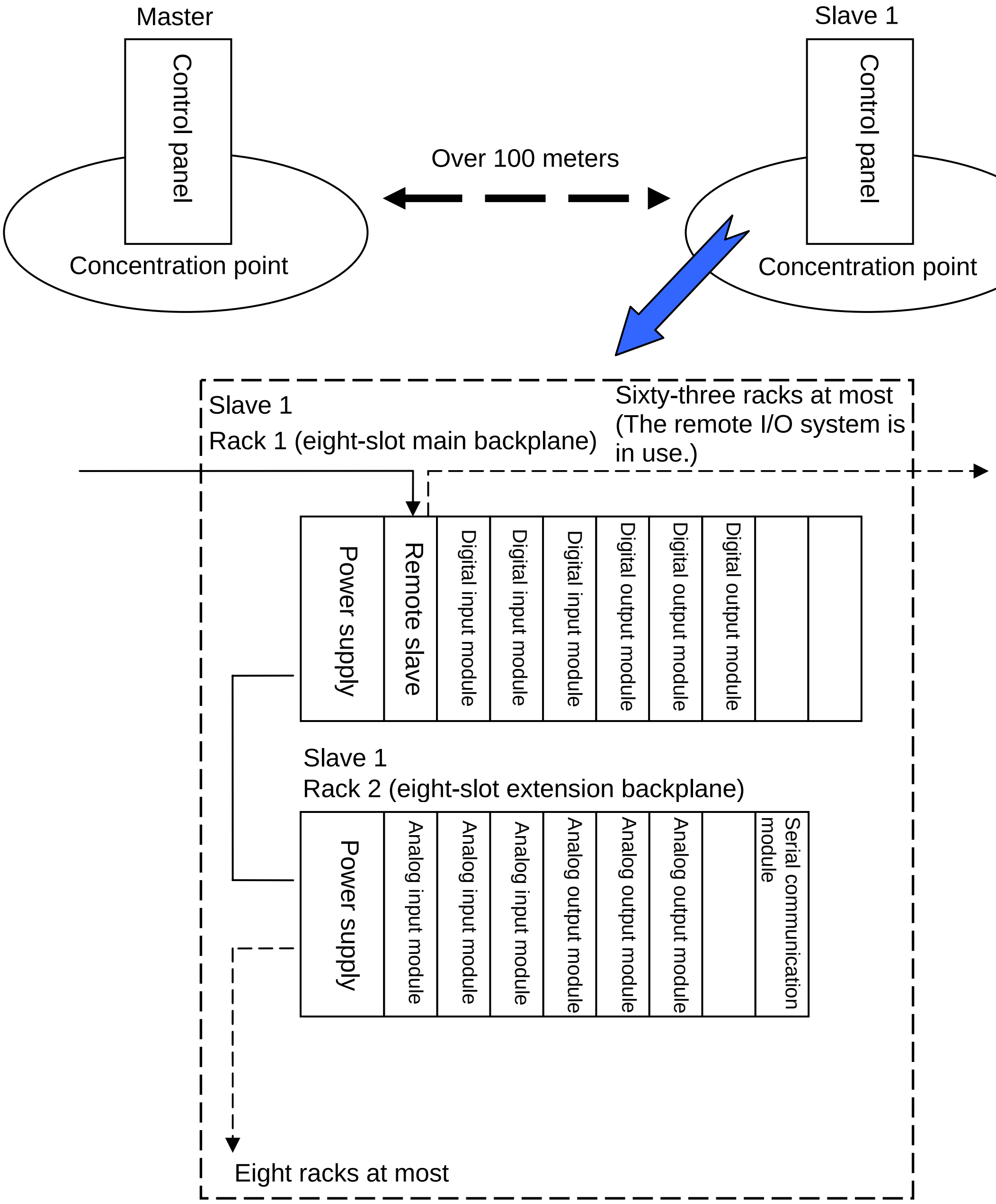


Note: Users are provided with the special cables to connect the extension backplanes. The length of a special cable can be up to 100 meters. They are also provided with the fiber cables which are used with the special adapters. The length of a fiber cable can be up to 2 kilometers.

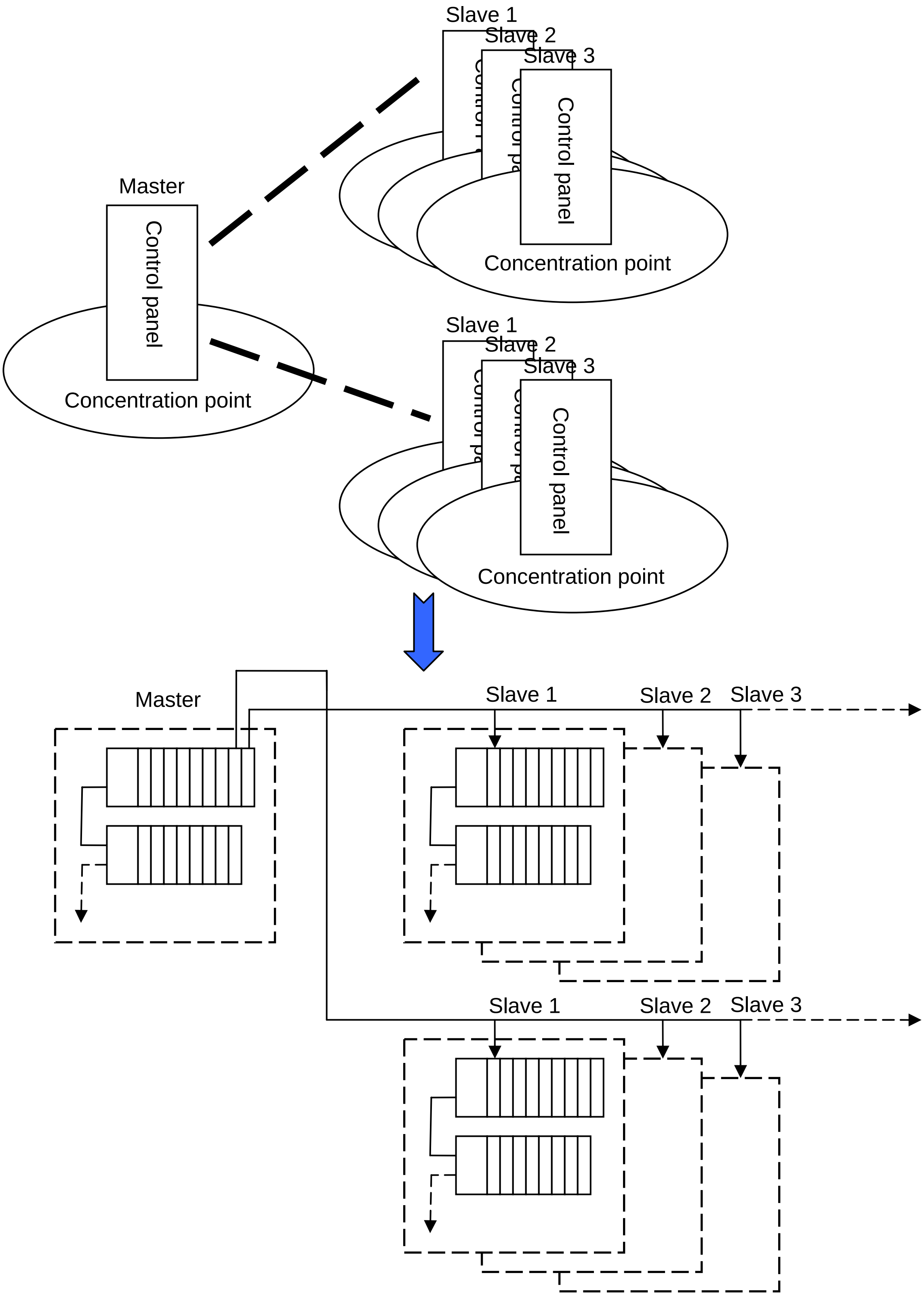
System planning 1



System planning 2



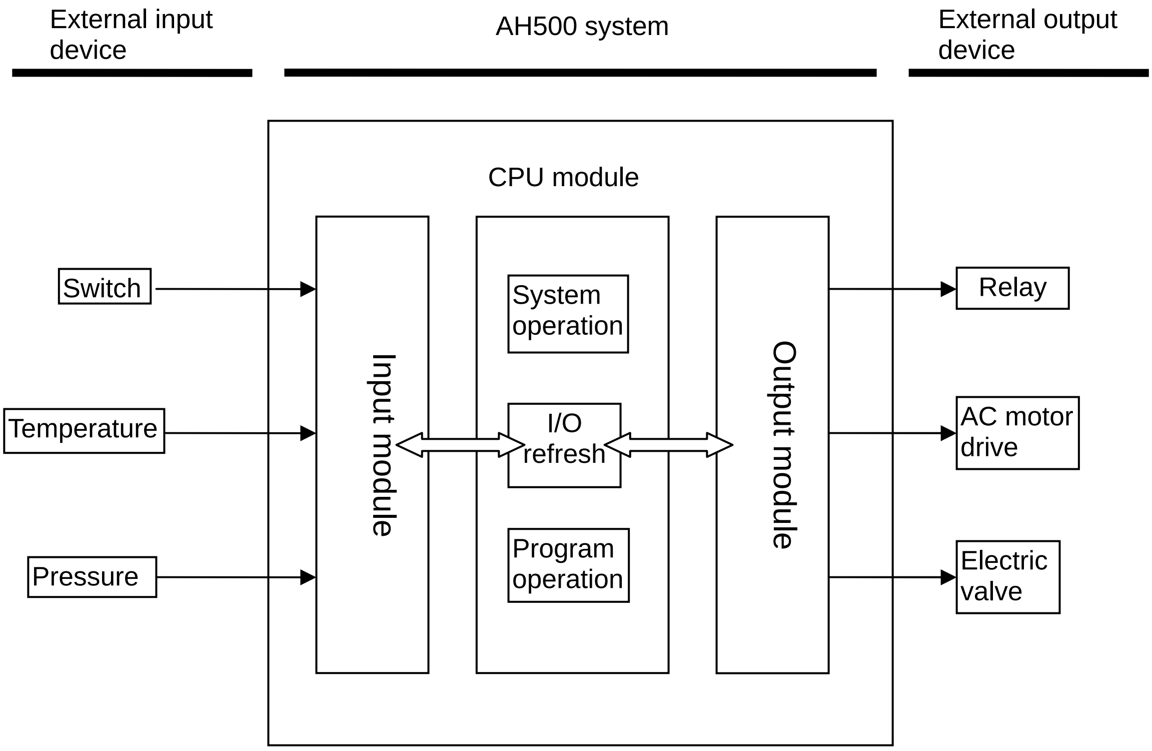
System planning 3



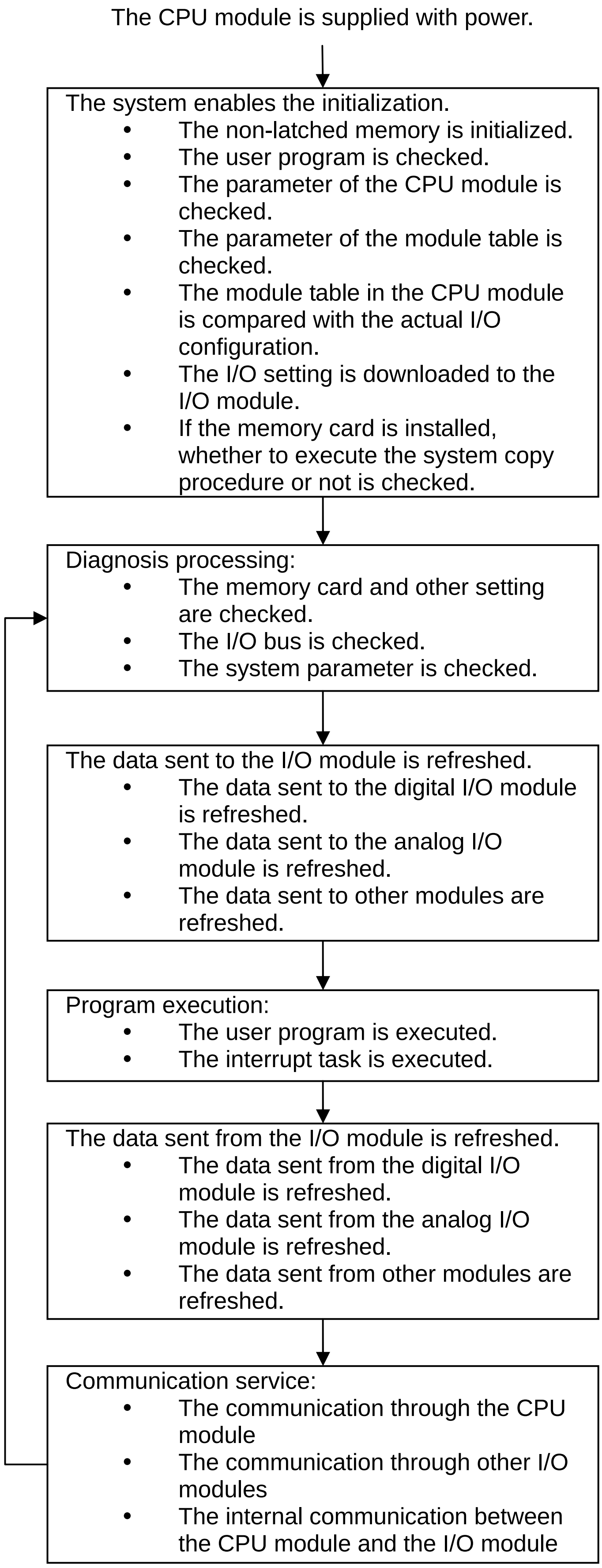
Note: The remote masters in DeviceNet are the network modules. They have to be installed with the CPU modules on the same backplane. A CPU module can support eight masters, and a master can be connected to sixty-three slaves. Besides, a slave can be connected to seven extension backplanes at most.

1.3 The Operation of the CPU module

The CPU module is the nucleus of the AH500 system. It is responsible for not only the execution of the logic program, but also the data exchange and the processing of the communication data. The relation between the AH500 system and the external devices are illustrated below.



The operation of the CPU module is illustrated above. The system procedures related to the initialization, the diagnosis, and the communication, and the program procedures related to the external interrupts and timed interrupts are simplified. Users can refer to other manuals for more information. The operation of the CPU module is described as follows.



The CPU module is supplied with power.

The system enables the initialization.

- The non-latched memory is initialized.
- The user program is checked.
- The parameter of the CPU module is checked.
- The parameter of the module table is checked.
- The module table in the CPU module is compared with the actual I/O configuration.
- The I/O setting is downloaded to the I/O module.
- If the memory card is installed, whether to execute the system copy procedure or not is checked.

Diagnosis processing:

- The memory card and other setting are checked.
- The I/O bus is checked.
- The system parameter is checked.

The data sent to the I/O module is refreshed.

- The data sent to the digital I/O module is refreshed.
- The data sent to the analog I/O module is refreshed.
- The data sent to other modules are refreshed.

Program execution:

- The user program is executed.
- The interrupt task is executed.

The data sent from the I/O module is refreshed.

- The data sent from the digital I/O module is refreshed.
- The data sent from the analog I/O module is refreshed.
- The data sent from other modules are refreshed.

Communication service:

- The communication through the CPU module
- The communication through other I/O modules
- The internal communication between the CPU module and the I/O module

Chapter 2 Programming

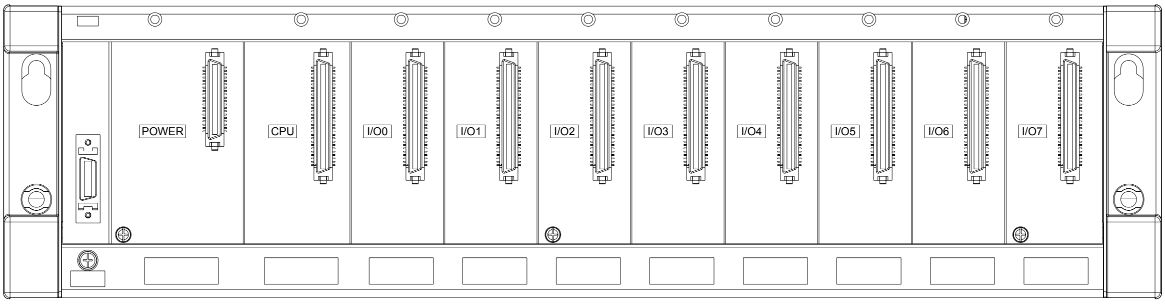
Table of Contents

2.1 Preparations

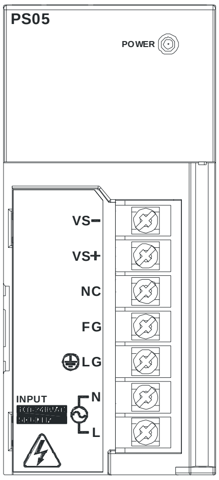
2.1.1 The Hardware

The hardware needed in the example is as follows.

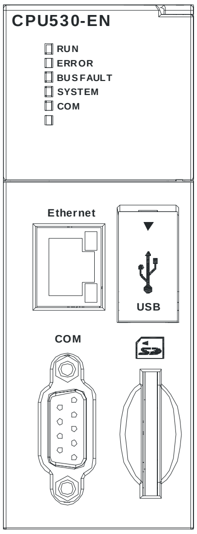
1. Eight-slot main backplane
   AHBP08M1-5A x 1



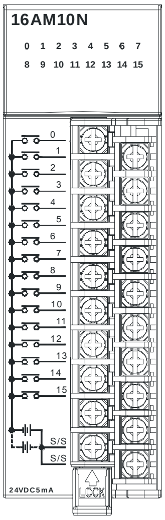
2. Power supply module
   AHPS05-5Ax 1



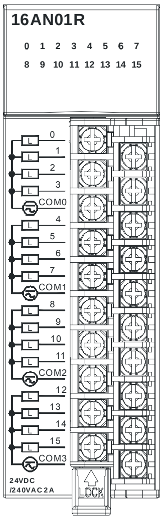
3. CPU module (with the built-in network function)
   AHCPU530-EN x 1



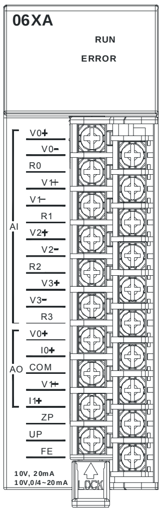
4. Digital input module (16 inputs)
   AH16AM10N-5A x 1



5. Digital output module (16 outputs)
   AH16AN01R-5A x 1



6. Analog input/output module (6 channels)
   AH06XA-5A x 1



2.1.2 The Software

The software needed in the example is as follows.

- ISPSoft version 2.0 and above
- COMMGR version 1.0 and above

2.1.3 Tools and Materials

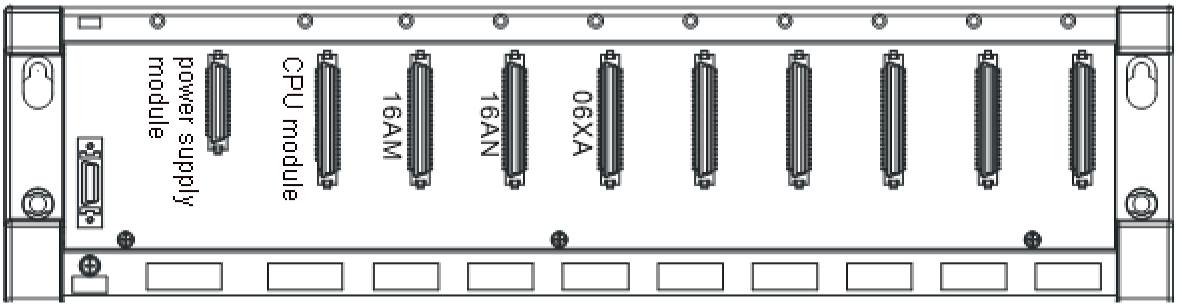
The tools and the materials need in the example are as follows.

- A personal computer in which the software mentioned above is installed
- A 100~240 V AC and 50/60 Hz power supply socket
- A 24 V DC power supply
- A cable
- A screwdriver
- An USB cable or a network cable (If users want to connect the Ethernet port or the COM port (RS-232/RS-485) on the CPU module to the computer, they can refer to section 2.3.2 in ISPSoft User Manual for more information. If users want to know more about installing the USB driver, they can refer to appendix A in AH500 Operation Manual.)
- If necessary, users can prepare the accessories such as a switch and a bulb (to simulate the activity of the external equipment).

2.2 Installation

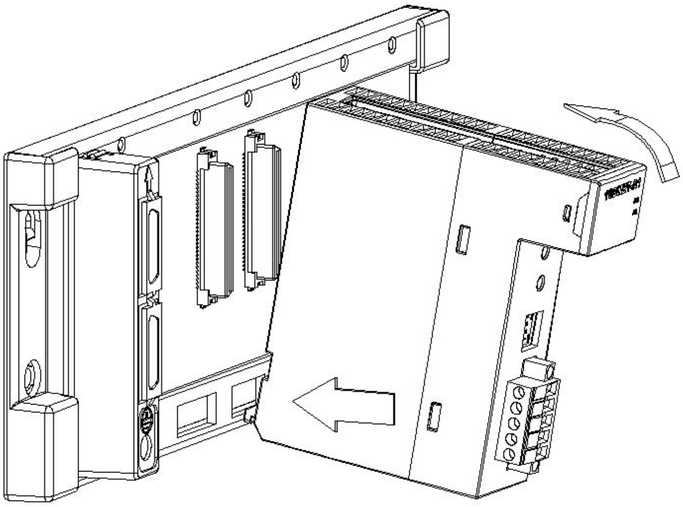
2.2.1 Installing Modules

Please install the modules on the main backplane, as illustrated below.

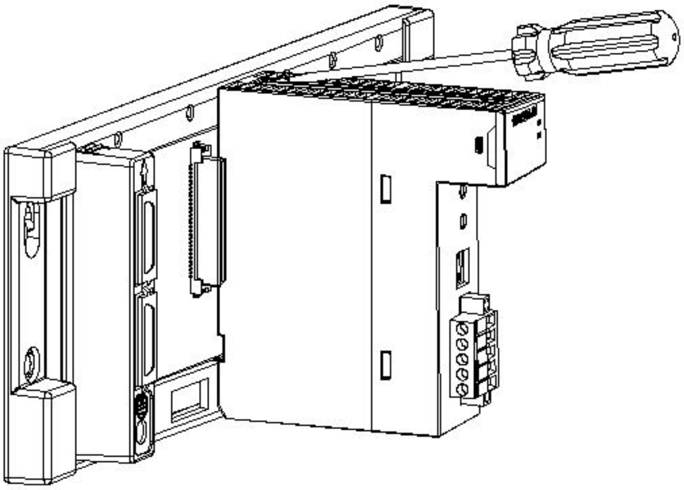


Connect the module to the connector on the backplane, make sure that the module is installed on the backplane properly, and tighen the the screw.

1. Insert the projection under the module into the hole in the backplane.
2. Push the module in the direction indicated by the arrow until it clicks.



3. Tighten the screw on the module.

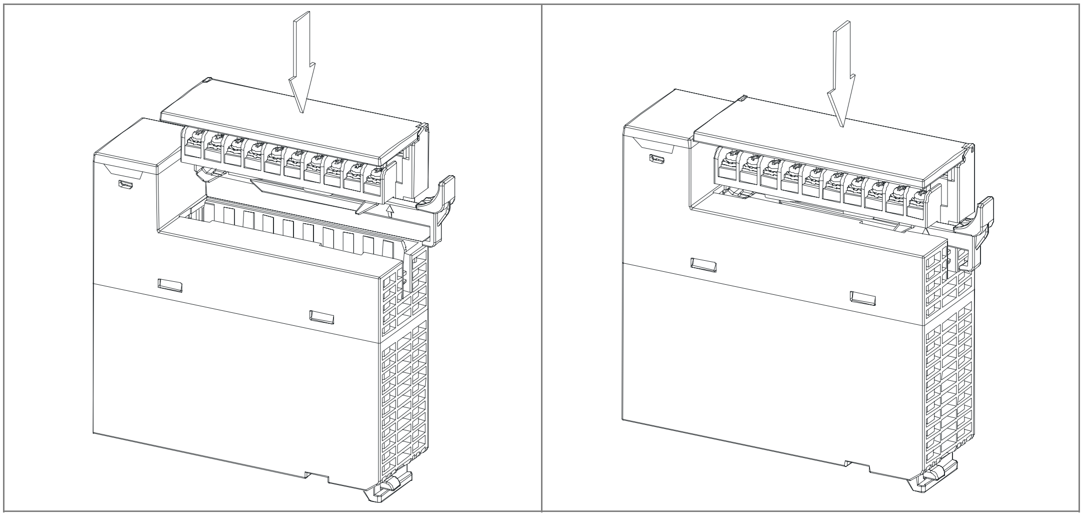


2.2.2 Installing Removable Terminal Blocks

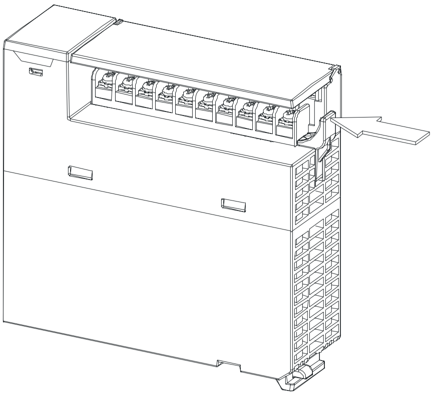
Please install the removable terminal block on the module, as illustrated below.

- **Installation**

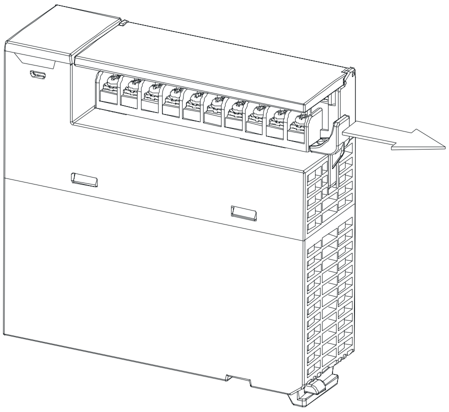
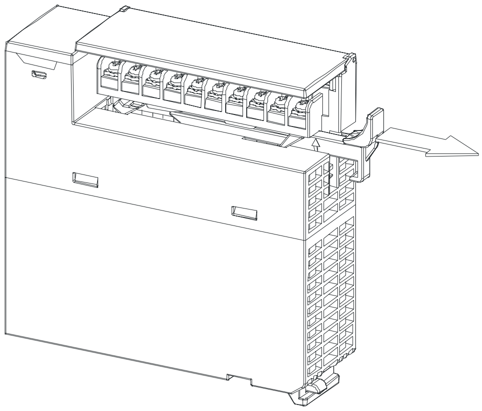
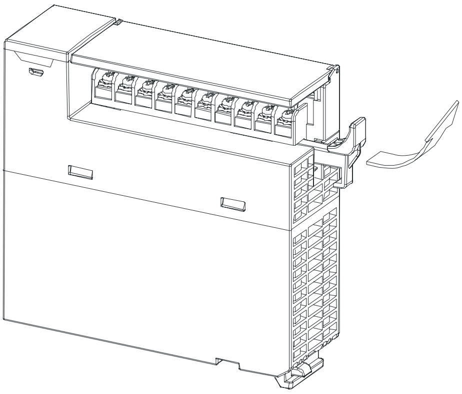
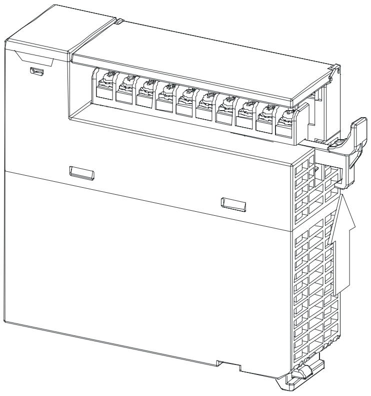
1. Level the terminal block at the printed circuit board, and press it into the module.



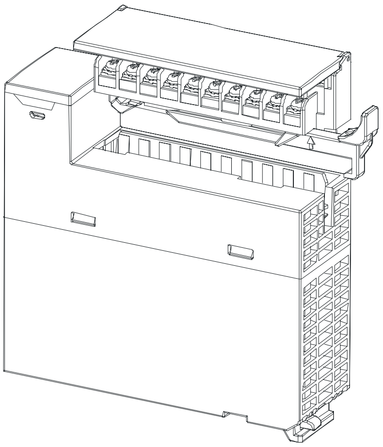
2. Press the clip in the direction indicated by the arrow.



- **Removal**

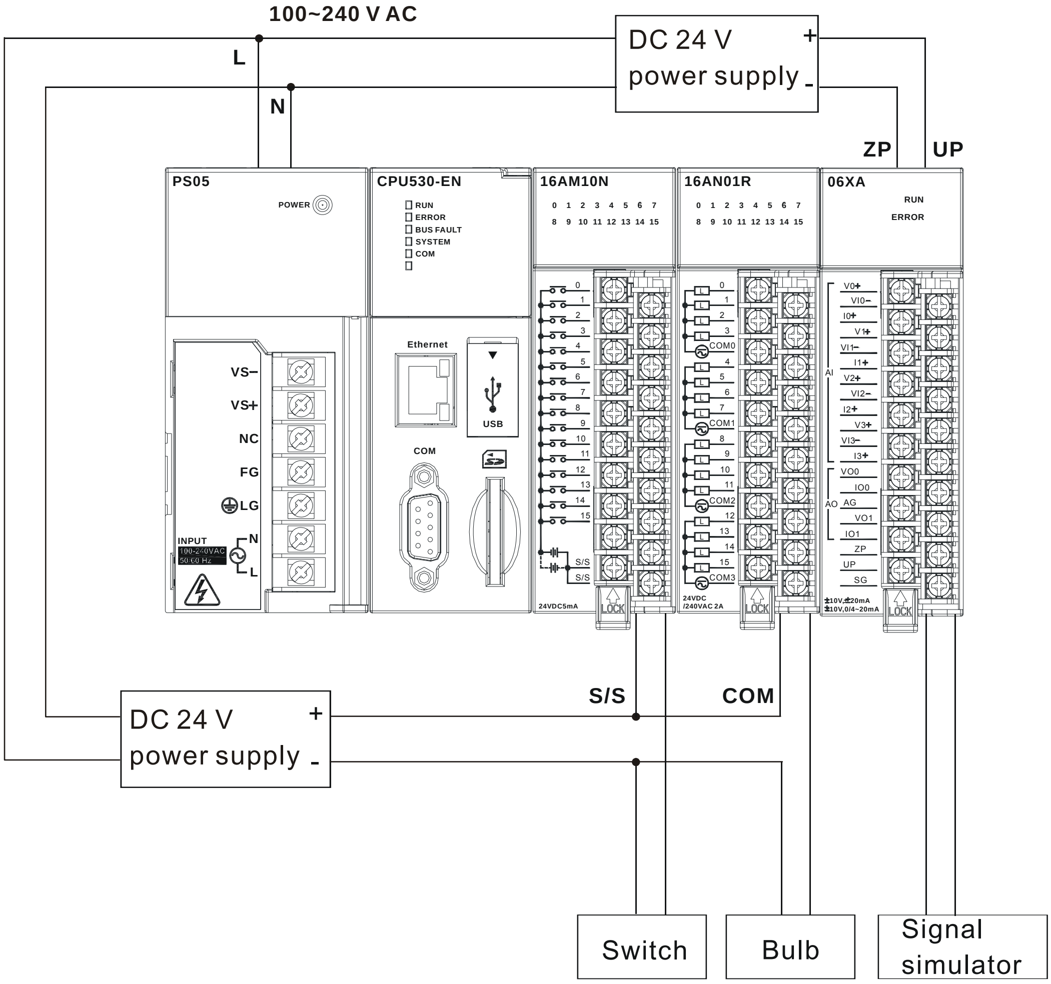
1. Pull the clip in the direction indicated by the arrow.	
	
2. Pull up the clip.	
	

3. The terminal block is removed.



2.3 Wiring

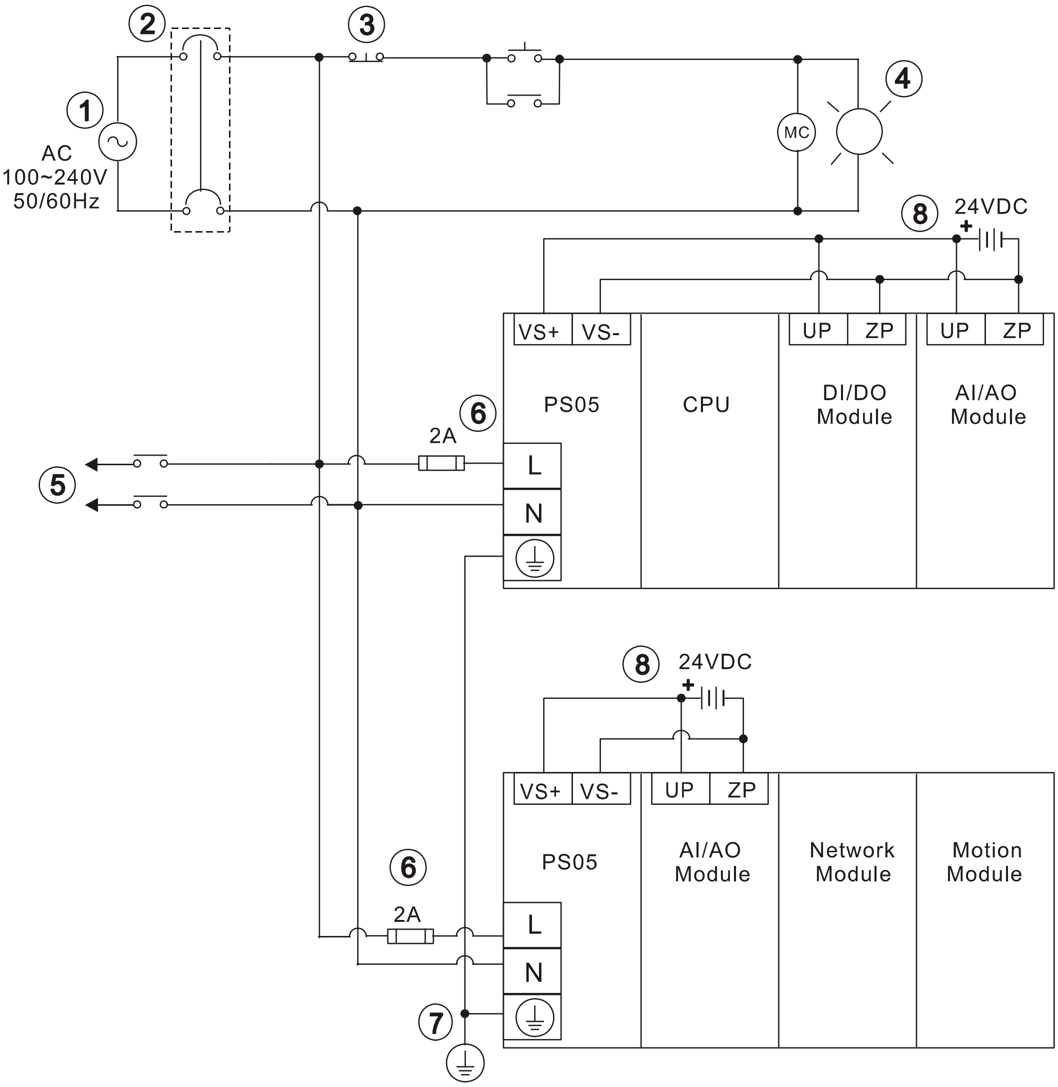
After the modules are installed, the wiring of the modules follows. In order for the following example to proceed smoothly, the power supply module and the analog module have to be wired. Be sure to cut off the power supply before wiring the modules. To lend convenience and reality to the simulation, the signal lines can be connected to the switch and the bulb according to the personal needs. The rough framework is as follows.



The wiring of the modules is described in detail below. (Please refer to AH500 Hardware Manual for more information.)

2.3.1 Wiring the Power Supply Module

- The alternating-current input voltage is within the range between 100 V AC and 240 V AC. Please connect the power supply to the terminals L and N. **If the 110 V AC or the 220 V AC power supply is connected to the input terminals VS+ and VS-, the PLC will be damaged.**
- In order to ensure the 24 V DC external power supply is provided stably, it can be connected to VS+ and VS-. If the PLC detects that the voltage of the external power supply is lower than the working voltage, users can write a protective program. (Please refer to section 6.6 in AH500 Operation Manual for more information.)
- The length of the wire connects with the ground is 1.6 millimeters.
- If the power cut lasts for less than 10 milliseconds, the PLC keeps running without being affected. If the power cut lasts for long, or if the voltage of the power supply decreases, the PLC stops running, and there is no output. When the power supply returns to normal, the PLC resumes. (Users have to notice that there are latched auxiliary relays and registers in the PLC when they write the program.)
- Please use the single-core cable or the twin-core cable. The diameter of the cable used should be with the range between 12 AWG and 22 AWG. The torque applied to the screw on the terminal should be within the range between 5 kg-cm (4.3 Ib-in) and 8 kg-cm (6.9 Ib-in). Please use the copper conducting wire. The temperature of the copper conducting wire should be 60/75°C.
- Safety wiring: The PLC controls many devices, and the activity of any device affects the activity of other devices. If any device breaks down, the whole automatic control system goes out of control, and the danger occurs. The protection circuit is as follows.

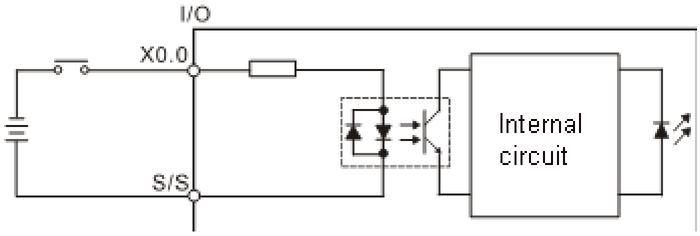


①	Alternating-current power supply: 100~240 V AC, 50/60 Hz
②	Circuit breaker
③	Emergency stop: The emergency stop button can be used to cut off the power when an emergency occurs.
④	Power indicator
⑤	AC power load
⑥	Fuse (2A)
⑦	The ground impedance is less than 100 Ω.
⑧	Direct-current power supply: 24 V DC

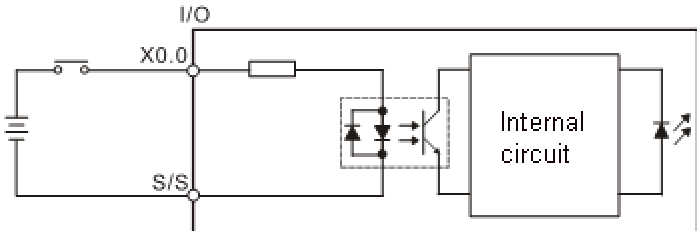
2.3.2 Wiring the Digital Input Module

The input signal is the direct-current power input. Sinking and sourcing are the current driving capability of a circuit. They are defined as follows.

- **Sinking**



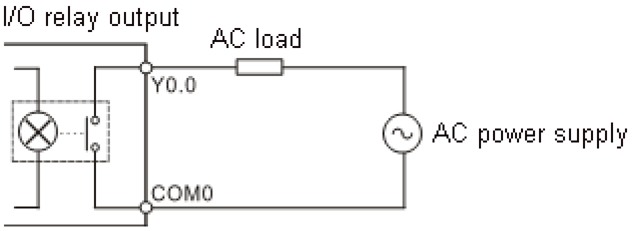
- **Sourcing**



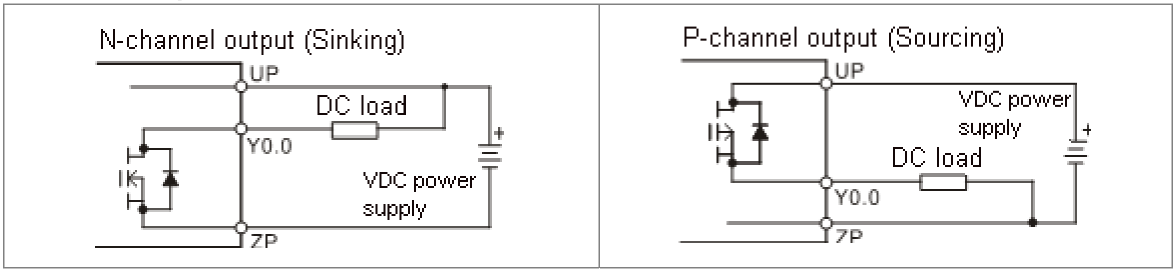
2.3.3 Wiring the Digital Output Module

The output unit can be the relay output, the transistor output, or the TRIAC output.

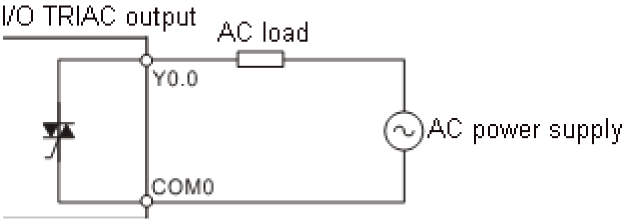
Relay output



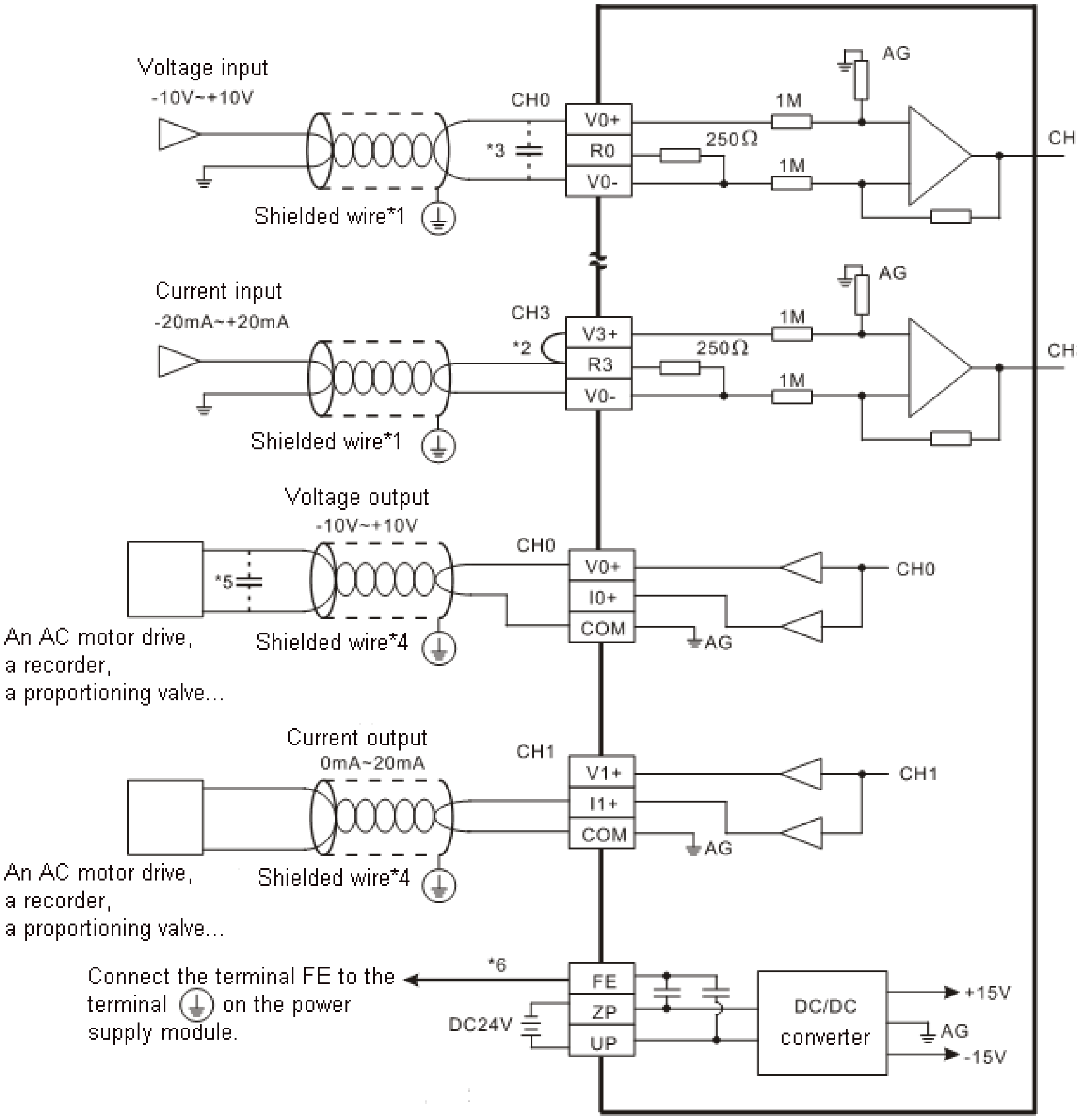
Transistor output



TRIAC output



2.3.4 Wiring the Analog Input/Output Module



*1. Please isolate the analog input from other wires.

*2. If the module is connected to the current signal, the terminals V+ and R have to be short-circuited.

*3. If the ripple of the input voltage results in the noise interference with the wiring, please connect the module to the capacitor having a capacitance within the range between 0.1 μF and 0.47 μF with a working voltage of 25 V.

*4. Please isolate the analog output from other wires.

*5. If the ripple is large for the input terminal of the load and results in the noise interference with the wiring, please connect the module to the capacitor having a capacitance within the range between 0.1 μF and 0.47 μF with a working voltage of 25 V.

*6. Please connect the terminal FE on AH06XA-5A to the terminal ⏚ on the power supply module.

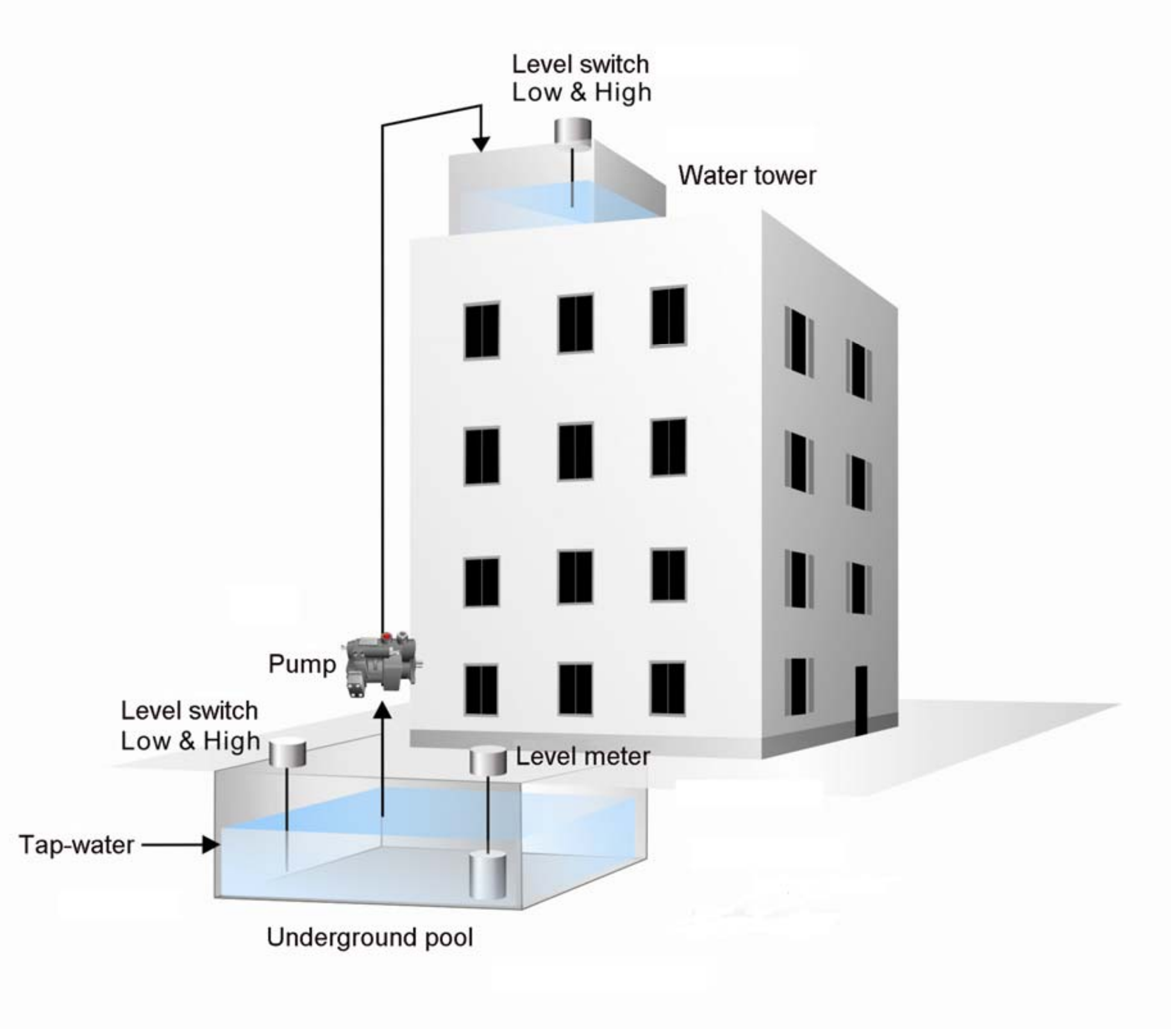
2.3.5 Supplying Power

After the wiring is complete, the CPU module can be supplied with power. Make sure that the CPU module is set to STOP before the CPU module is supplied with power. The CPU module executes the initialization after it is supplied with power. Owing to the fact that there is no hardware configuration in the CPU module, the error LED indicator is ON after the check is complete. This is a normal phenomenon which can be ignored temporarily.

2.4 Exemplification

After the hardware is installed, the wiring of the modules is complete, and the CPU module is supplied with power, users can write the program. In order for users to have a precise object and direction, the manual provides users with an example before they write the program. The manual teaches users how to create a new project and how to download the program to the CPU module step by step. The following are the contents of the example and the illustration of the framework.

- **System framework**



- **Control action**

The example is the basic design of the water supply of the multi-storey building. The tap-water is automatically supplied to the underground pool, and the water in the underground pool can be transported to the water tower on the top of the building through the pump. The water is distributed to every story in the building by means of gravity, and the action of the pump is controlled by the level switch of the underground pool and that that of the water tower. In order to monitor the water supply, the level meter is installed in the underground pool. The water storage capacity of the underground pool is monitored at all times.

The devices connected to the modules:

1. A single-point level switch (contact A)

   The single-point level switch is installed in the underground pool, and the signal contact is connected to the digital input module.
2. A two-point level switch (contact A)

The two-point level switch is installed in the water tower on the top of the building, and the signal contact is connected to the digital input module.

3. A pump

   The pump is installed near the underground pool. However, the device to which the PLC actually connects is not the pump but the control panel. Generally speaking, three digital inputs and one digital output are connected to the digital input/output module. (Remote control x 1 (DI) & Run x 1 (DI) & Trip x 1 (DI) & Start x 1 (DO))

4. A level meter

   The level meter is installed in the underground pool, and the signal contact is connected to the analog input module. (0~10V correspond to 0~10M. 0 V represents that the water is 0.0 meters deep, and 10 V represents that the water is 10.0 meters deep.)

The control condition of the pump:

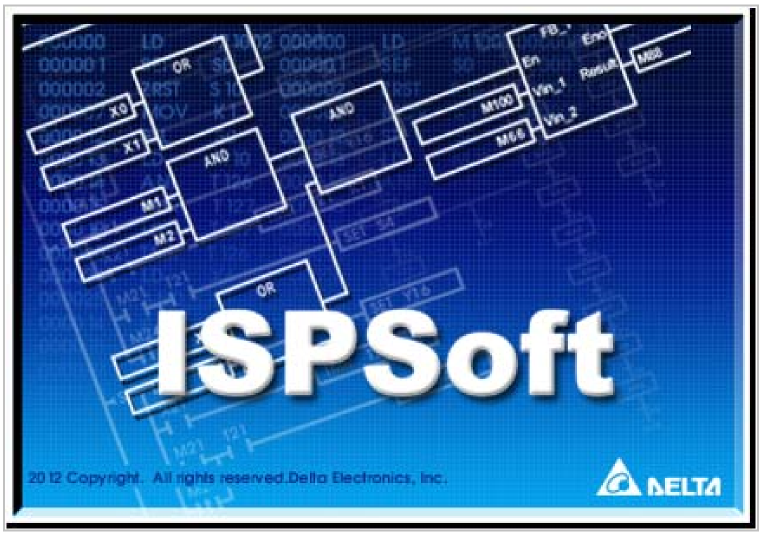
1. Start: If the water level inside the water tower is low and that inside the underground pool is not low, the pump will refill the water tower.
2. Stop: If the water level inside the water tower is high, or if the water level inside the underground pool is low, the pump stops running.

2.5 Creating Projects

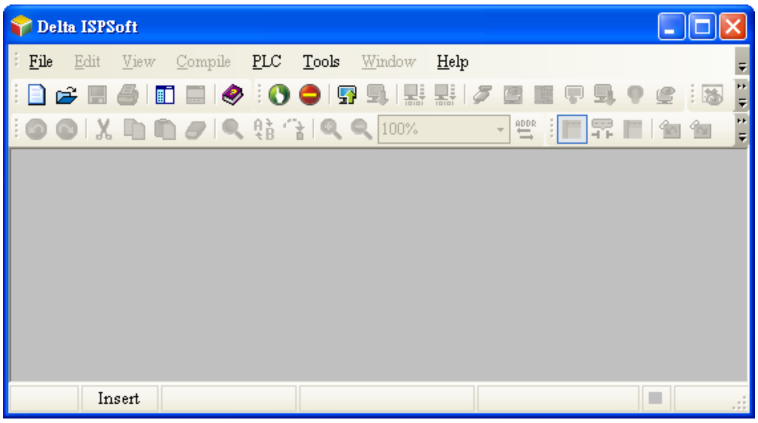
After users install the hardware and understand the control logic in the example, they can write the program.

Step 1: Open ISPSoft (Start→Programs→ Delta Industrial Automation→PLC→ISPSoftx.xx→ISPSoftx.xx)

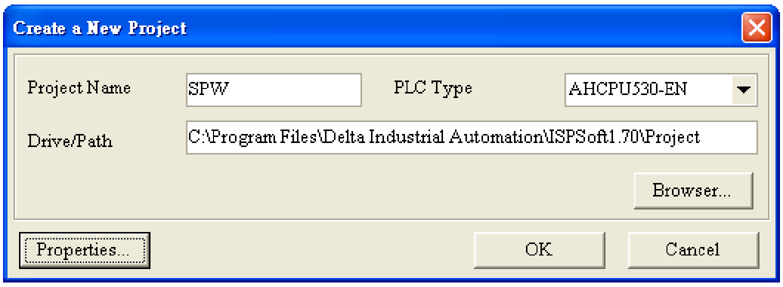
The start screen of the software:



Entering the main screen of the software:



Step 2: Click to create a project. The example is related to the water supply, and therefore the project name is SPW. The PLC type is AHCPU530-EN, and the file path is the default path. Finally, click **OK**.



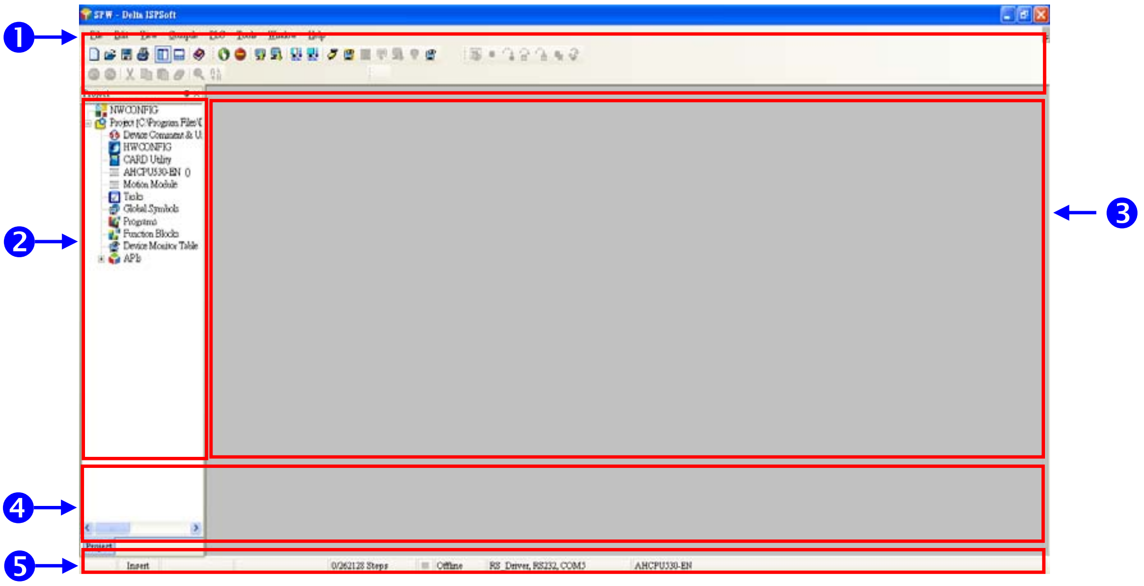
Create a New Project

Project Name: SPW

PLC Type: AHCPU530-EN

Drive/Path: C:\Program Files\Delta Industrial Automation\ISPSoft1.70\Project

Next, the environment of the project is displayed, and the project name SPW appears in the upper left corner.



The operation interface of the software:

❶ Function area: The main functions of the software are in this area. Many functions which are frequently used are placed on the toolbar, and other functions are placed on the menus.

❷ Project management area: The framework of the project is displayed in this area. Users can understand the relation among the objects on the basis of the tree structure. The efficiency in managing the project is also increased.

❸ Work area: The editing work is in this area.

❹ Output area: The information resulting from the execution of the function is displayed in this area.

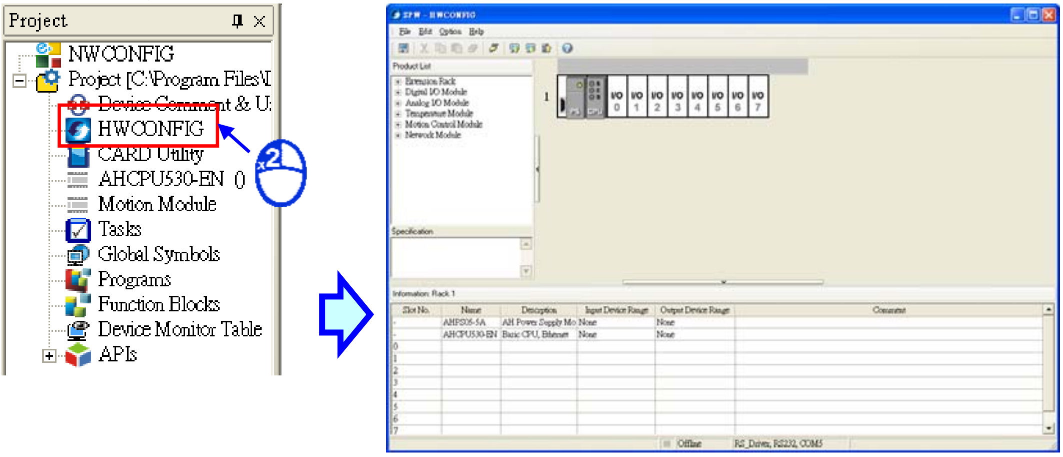
❺ Status area: The project and the communication information are displayed in this area.

2.6 The Hardware Configuration

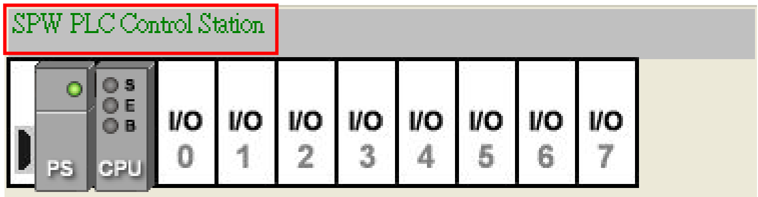
After the project is created, users can configure the hardware. Suppose the configuration is as follows.

- Digital input module 16AM10N-5A/16AM30N-5A (16 inputs)→X0.0~X0.15
- Digital output module 16AN01R-5A/16AN01T-5A/16AN01P-5A/16AN01S-5A (16 outputs) →Y0.0 ~ Y0.15
- Four-channel analog input module AH06XA-5A→D0~D7
- Two-channel analog output module AH06XA-5A→D100~D103
- The water level inside the underground pool is low.→X0.0
- The water level inside the water tower is low.→X0.2
- The water level inside the water tower is high.→X0.3
- Remote control of the pump→X0.5
- The pump runs.→X0.6
- The pump trips.→X0.7
- The pump starts.→Y0.0
- Water level inside the underground pool→D0

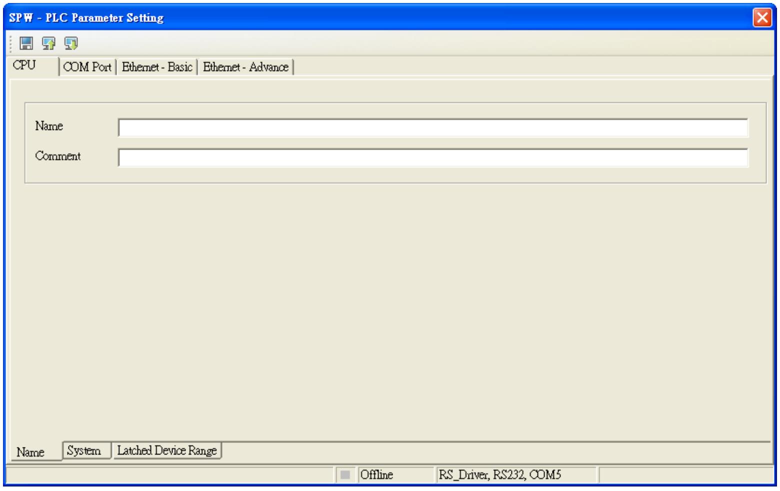
With the information above, user can configure the hardware practically. Find HWCONFIG in the window at the left side of the main screen of the software, and double-click the left mouse button to open the configuration window.



Users can configure the hardware according to the information above. After users type “SPW PLC Control Station”, the project can be shared.



Double-click the CPU module to open the **PLC Parameter Setting** window.

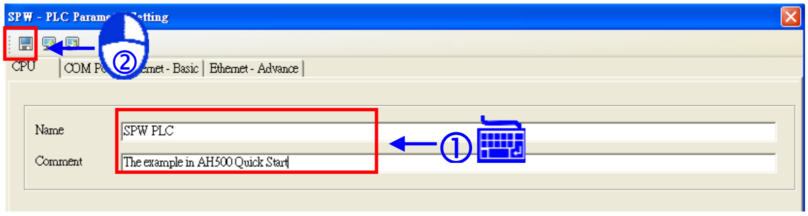


Name

Comment

Offline | RS_Driver, RS232, COM5

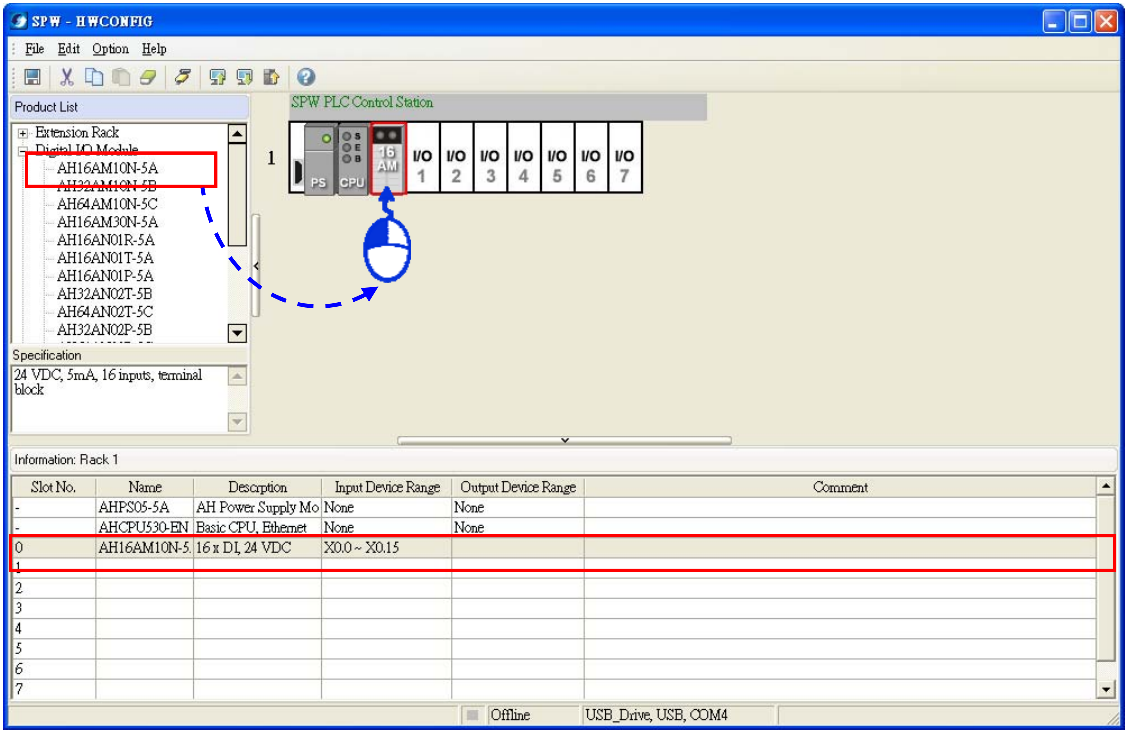
Type “SPW PLC” in the **Name** box, and then type “The example in AH500 Quick Start” in the **Comment** box. Finally, save the setting, and close the window.



Name: SPW PLC

Comment: The example in AH500 Quick Start

User can begin to place the modules. First, find the first module which is needed, that is, AH16AM10N-5A, in the product list. Then, drag the module to I/O 0 and drop it.



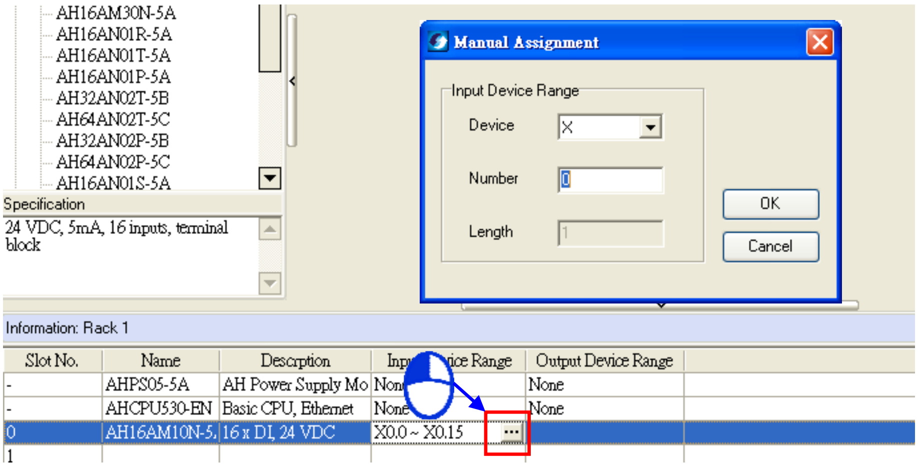
Specification: 24 VDC, 5mA, 16 inputs, terminal block

Information: Rack 1

Slot No.	Name	Description	Input Device Range	Output Device Range	Comment
-	AHPS05-5A	AH Power Supply Mo	None	None	
-	AHCPU530-EN	Basic CPU, Ethernet	None	None	
0	AH16AM10N-5.	16 x DI, 24 VDC	X0.0 ~ X0.15		
1					
2					
3					
4					
5					
6					
7					

Offline | USB_Drive, USB, COM4

After the module is placed in the appropriate position, the system automatically distributes the addresses to the module. The default addresses are X0.0~X0.15. They exactly meet the need, and therefore they do not need to be modified. If they do not meet the need, users can click the column to modify the addresses.



24 VDC, 5mA, 16 inputs, terminal block

Device: X

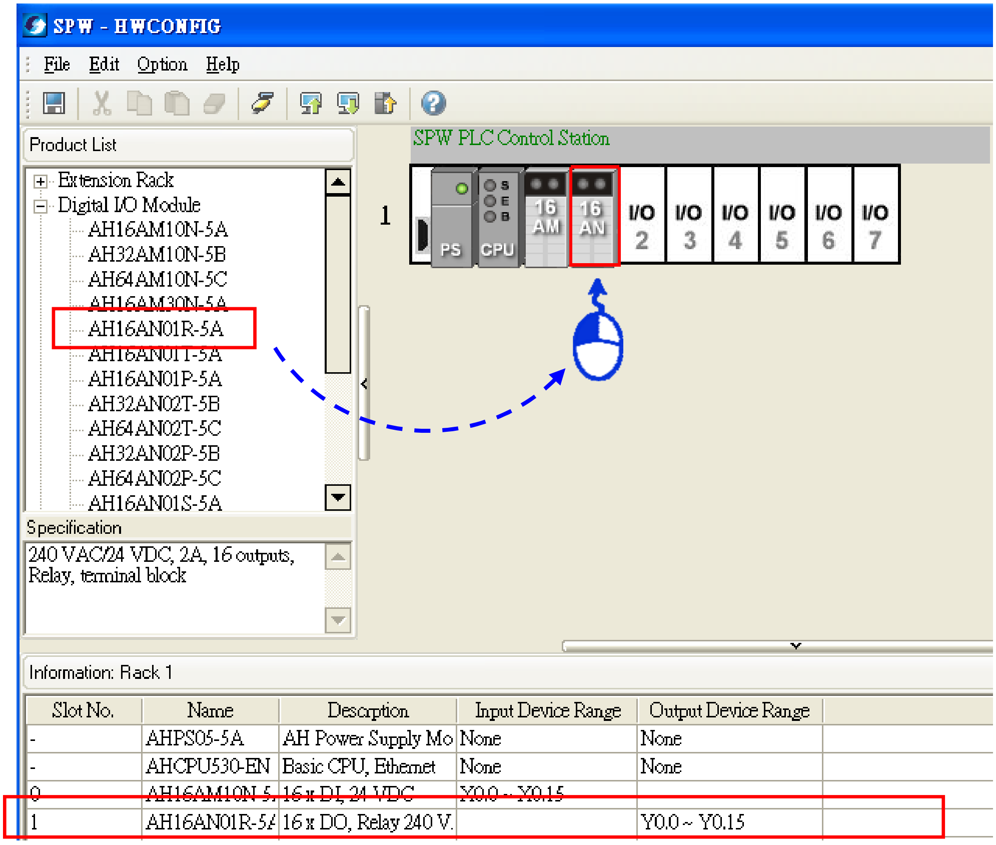
Number: 0

Length: 1

Information: Rack 1

Slot No.	Name	Description	Input Device Range	Output Device Range
-	AHPS05-5A	AH Power Supply Mo	None	None
-	AHCPU530-EN	Basic CPU, Ethernet	None	None
0	AH16AM10N-5,	16 x DI, 24 VDC	X0.0 ~ X0.15 ...	
1				

Users can drag AH16AN01R-5A to I/O 1 and drop it. The addresses are Y0.0~Y0.15.

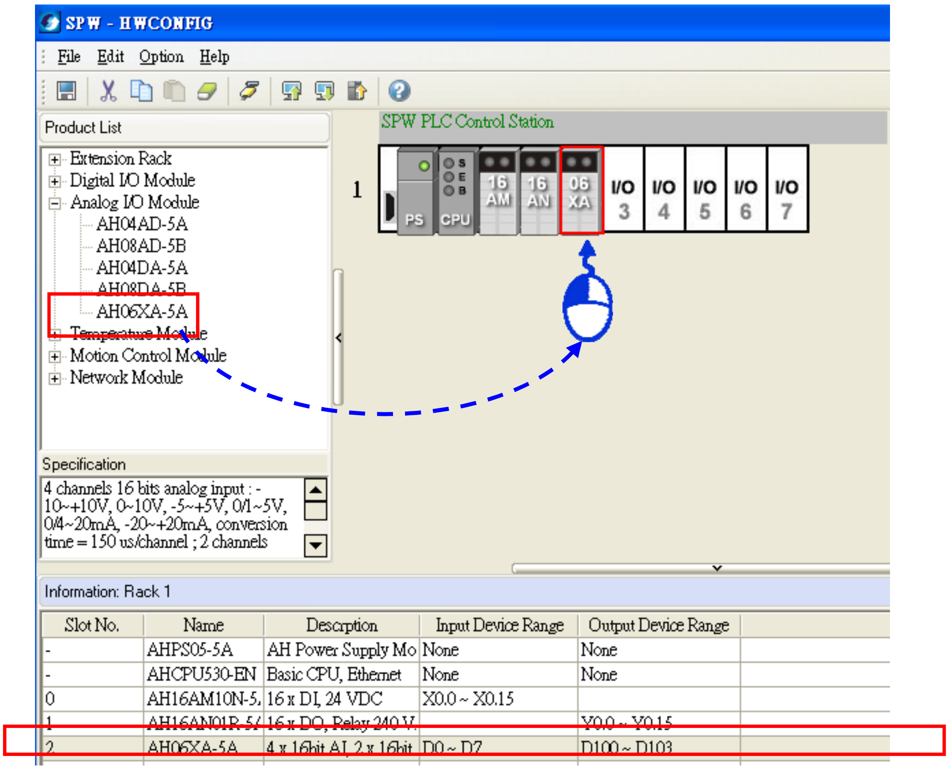


240 VAC/24 VDC, 2A, 16 outputs, Relay, terminal block

Information: Rack 1

Slot No.	Name	Description	Input Device Range	Output Device Range
-	AHPS05-5A	AH Power Supply Mo	None	None
-	AHCPU530-EN	Basic CPU, Ethernet	None	None
0	AH16AM10N-5,	16 x DI, 24 VDC	X0.0 ~ X0.15	
1	AH16AN01R-5/	16 x DO, Relay 240 V.		Y0.0 ~ Y0.15

Users can drag AH06XA-5A to I/O 2 and drop it. The input device range is D0~D7, and the output device range is D100~D103. (The default output device range is D0~D11. Please remember to modify it.)



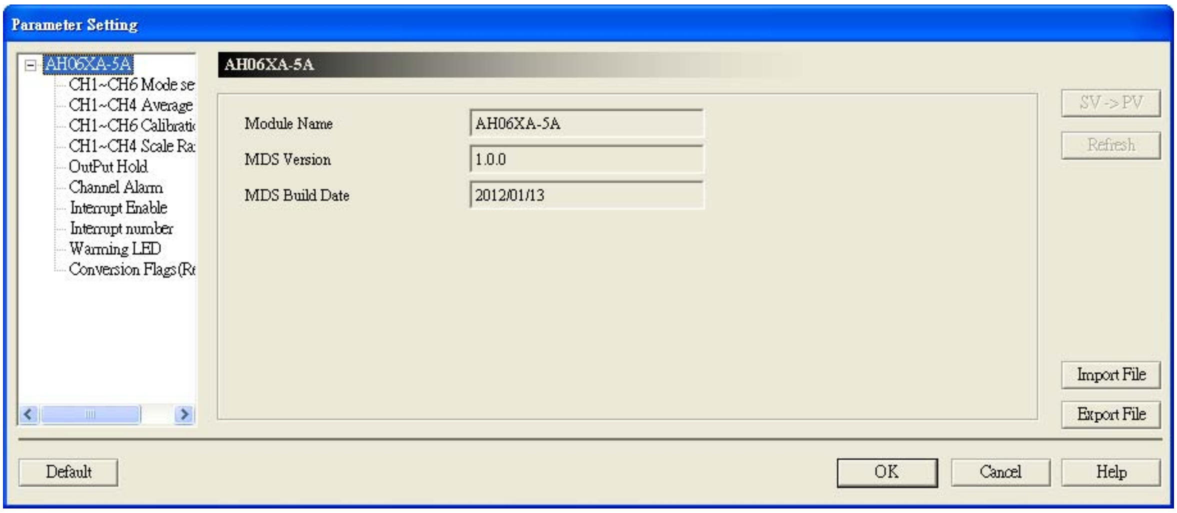
Specification

4 channels 16 bits analog input : -10~+10V, 0~10V, -5~+5V, 0/1~5V, 0/4~20mA, -20~+20mA, conversion time = 150 us/channel ; 2 channels

Information: Rack 1

Slot No.	Name	Description	Input Device Range	Output Device Range
-	AHPS05-5A	AH Power Supply Mo	None	None
-	AHCPU530-EN	Basic CPU, Ethernet	None	None
0	AH16AM10N-5.	16 x DI, 24 VDC	X0.0 ~ X0.15	
1	AH16AN01R-5/	16 x DO, Relay 240 V.		Y0.0 ~ Y0.15
2	AH06XA-5A	4 x 16bit AI, 2 x 16bit	D0 ~ D7	D100 ~ D103

In order to complete the basic setting, users need to set the relation between the signals and the conversion values for the analog module. Double-click the module to open the **Parameter Setting** window.



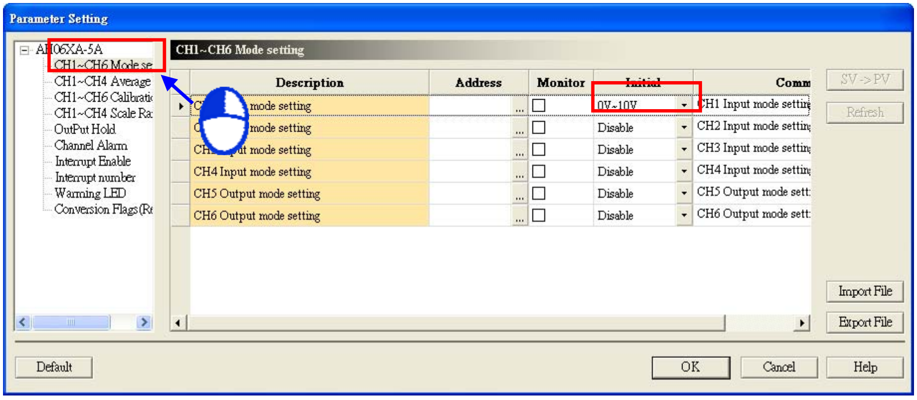
AH06XA-5A

Module Name: AH06XA-5A

MDS Version: 1.0.0

MDS Build Date: 2012/01/13

The voltages of the signals are 0 V~10 V.

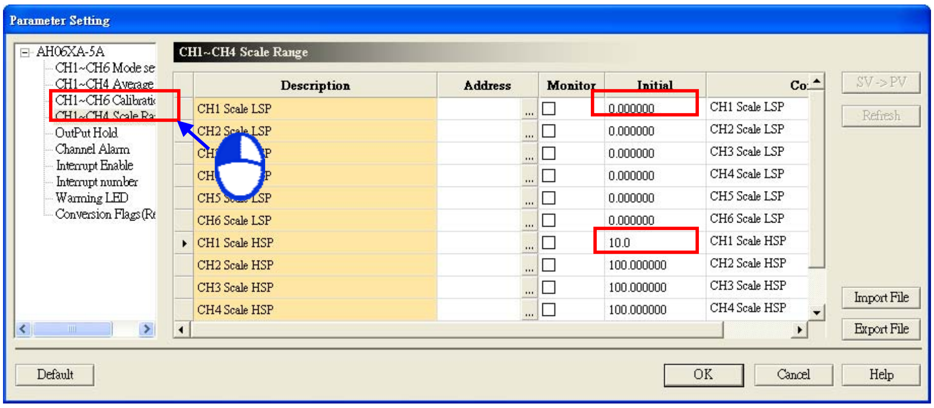


Parameter Setting

CH1~CH6 Mode setting

Description	Address	Monitor	Initial	Comm
C mode setting	...	□	0V~10V	CH1 Input mode settin
C mode setting	...	□	Disable	CH2 Input mode settin
CH t mode setting	...	□	Disable	CH3 Input mode settin
CH4 Input mode setting	...	□	Disable	CH4 Input mode settin
CH5 Output mode setting	...	□	Disable	CH5 Output mode sett:
CH6 Output mode setting	...	□	Disable	CH6 Output mode sett:

The conversion values are 0.0 and 10.0. Click **OK**.

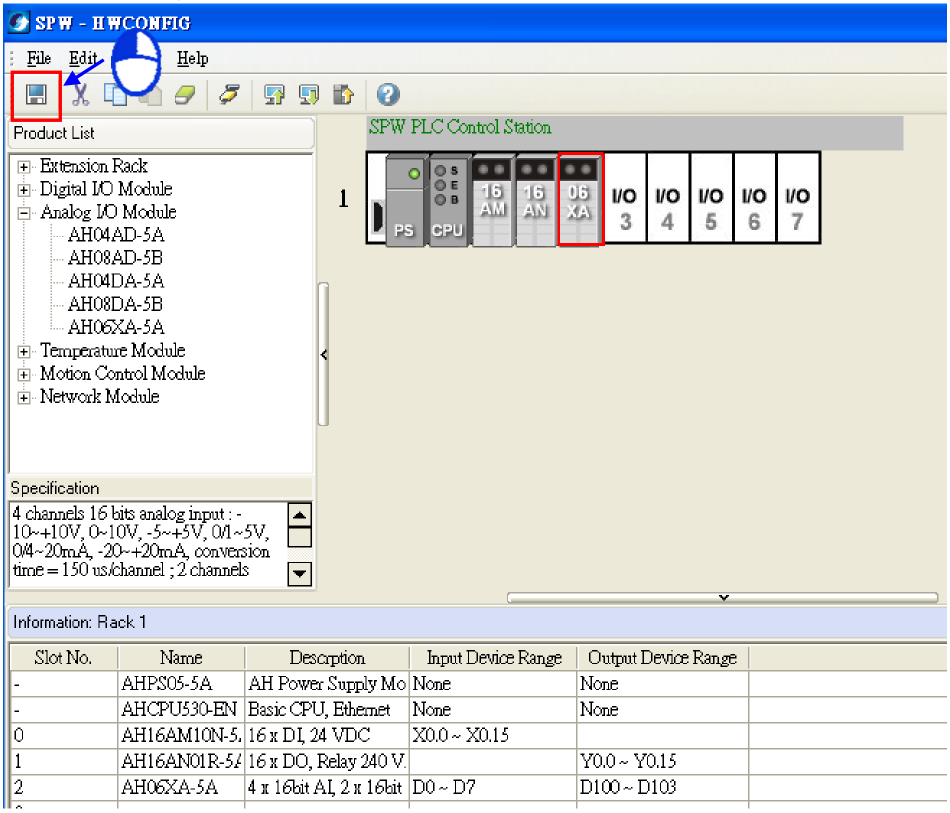


Parameter Setting

CH1~CH4 Scale Range

Description	Address	Monitor	Initial	Co
CH1 Scale LSP	...	□	0.000000	CH1 Scale LSP
CH2 Scale LSP	...	□	0.000000	CH2 Scale LSP
CH P	...	□	0.000000	CH3 Scale LSP
CH P	...	□	0.000000	CH4 Scale LSP
CH5 LSP	...	□	0.000000	CH5 Scale LSP
CH6 Scale LSP	...	□	0.000000	CH6 Scale LSP
CH1 Scale HSP	...	□	10.0	CH1 Scale HSP
CH2 Scale HSP	...	□	100.000000	CH2 Scale HSP
CH3 Scale HSP	...	□	100.000000	CH3 Scale HSP
CH4 Scale HSP	...	□	100.000000	CH4 Scale HSP

After the hardware configuration is complete, please save the file and exit.



SPW - HWCONFIG

Specification

4 channels 16 bits analog input : -10~+10V, 0~10V, -5~+5V, 0/1~5V, 0/4~20mA, -20~+20mA, conversion time = 150 us/channel ; 2 channels

Information: Rack 1

Slot No.	Name	Description	Input Device Range	Output Device Range
-	AHPS05-5A	AH Power Supply Mo	None	None
-	AHCPU530-EN	Basic CPU, Ethernet	None	None
0	AH16AM10N-5.	16 x DI, 24 VDC	X0.0 ~ X0.15	
1	AH16AN01R-5/	16 x DO, Relay 240 V.		Y0.0 ~ Y0.15
2	AH06XA-5A	4 x 16bit AI, 2 x 16bit	D0 ~ D7	D100 ~ D103

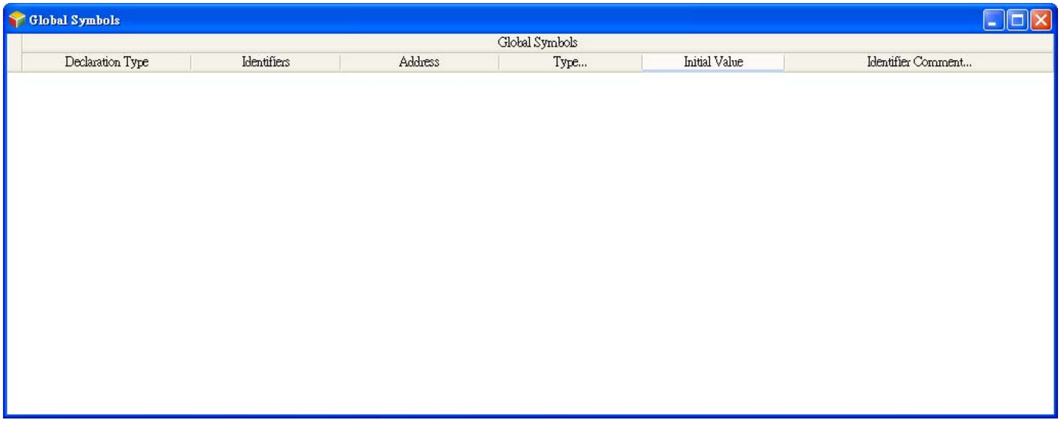
2.7 Creating Global Symbols

In order to make the program more readable and the connection with the SCADA system more convenient, the I/O addresses are accompanied with the global symbols. Users can use the symbols when they write the program. The global symbol table also supports import and export. As to the system equipped with many inputs and outputs, users can use Microsoft Excel to make the editing more convenient.

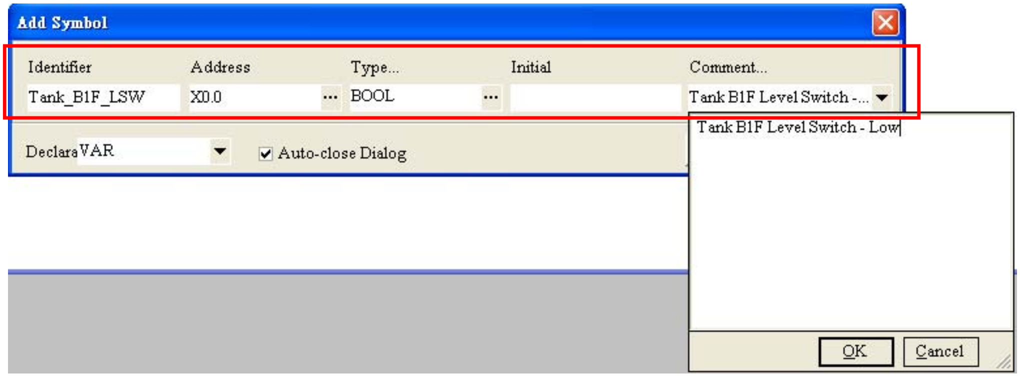
The global symbols created are as follows.

Global symbol table		
Bit (for the I/O on the PLC)		
Address	**Identifier**	**Data type**
X0.0	Tank_B1F_LSW	BOOL
X0.2	Tank_RF_LSW	BOOL
X0.3	Tank_RF_HSW	BOOL
X0.5	SPP01_Remote	BOOL
X0.6	SPP01_Run	BOOL
X0.7	SPP01_Trip	BOOL
Y0.0	SPP01_Start	BOOL
D0	Tank_B1F_LT	REAL
Bit (for the SCADA system)		
M0	SPP01_Auto	BOOL
M1	SPP01_Man_SW	BOOL

With the information above, users can create the global symbols. Find Global Symbols in the window at the left side of the main screen of the software, and double-click the left mouse button to open the **Global Symbols** window.



Double-click the blank to open the **Add Symbol** window. Type “X0.0” in the **Address** box. The default values in the **Type…** box, the **Initial** box, and the **Comment…** box remain unchanged. Click **OK** to complete the typing.



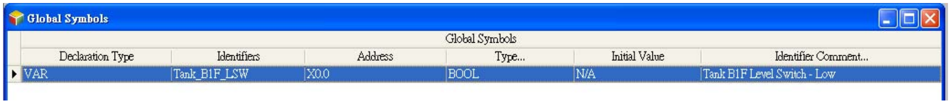
Identifier	Address	Type...	Initial	Comment...
Tank_B1F_LSW	X0.0	BOOL		Tank B1F Level Switch -...

Comment: Tank B1F Level Switch - Low

Declara VAR

☑ Auto-close Dialog

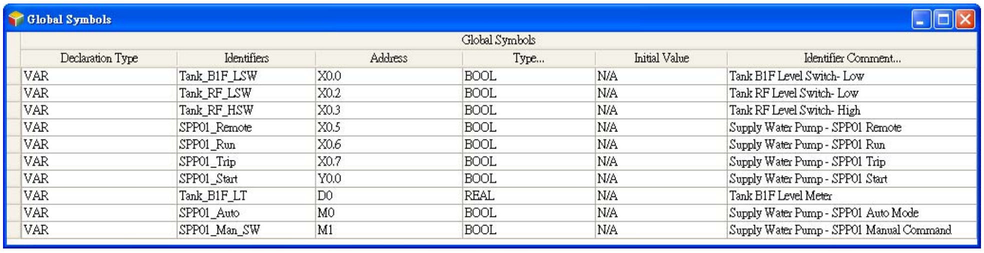
Users can see a new piece of data.



Global Symbols

Declaration Type	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	Tank_B1F_LSW	X0.0	BOOL	N/A	Tank B1F Level Switch - Low

Users can create the global symbols for the data in the table above in the same way. (For the analog module, the data type is a real floating-point number. Therefore, the symbol occupies two data registers, and the address in the **Address** box is the initial address.)



Global Symbols

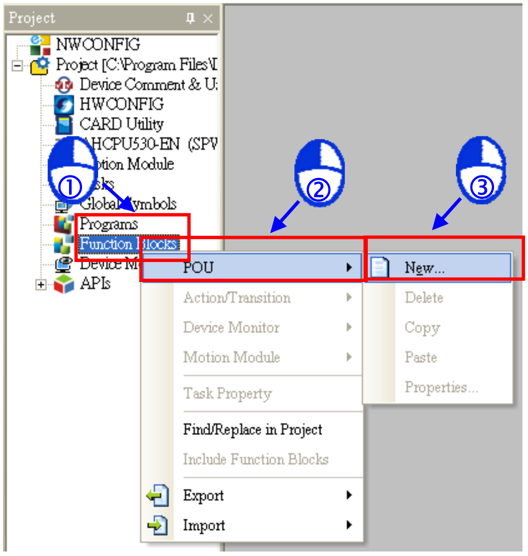
Declaration Type	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	Tank_B1F_LSW	X0.0	BOOL	N/A	Tank B1F Level Switch- Low
VAR	Tank_RF_LSW	X0.2	BOOL	N/A	Tank RF Level Switch- Low
VAR	Tank_RF_HSW	X0.3	BOOL	N/A	Tank RF Level Switch- High
VAR	SPP01_Remote	X0.5	BOOL	N/A	Supply Water Pump - SPP01 Remote
VAR	SPP01_Run	X0.6	BOOL	N/A	Supply Water Pump - SPP01 Run
VAR	SPP01_Trip	X0.7	BOOL	N/A	Supply Water Pump - SPP01 Trip
VAR	SPP01_Start	Y0.0	BOOL	N/A	Supply Water Pump - SPP01 Start
VAR	Tank_B1F_LT	D0	REAL	N/A	Tank B1F Level Meter
VAR	SPP01_Auto	M0	BOOL	N/A	Supply Water Pump - SPP01 Auto Mode
VAR	SPP01_Man_SW	M1	BOOL	N/A	Supply Water Pump - SPP01 Manual Command

2.8 Creating Function Blocks

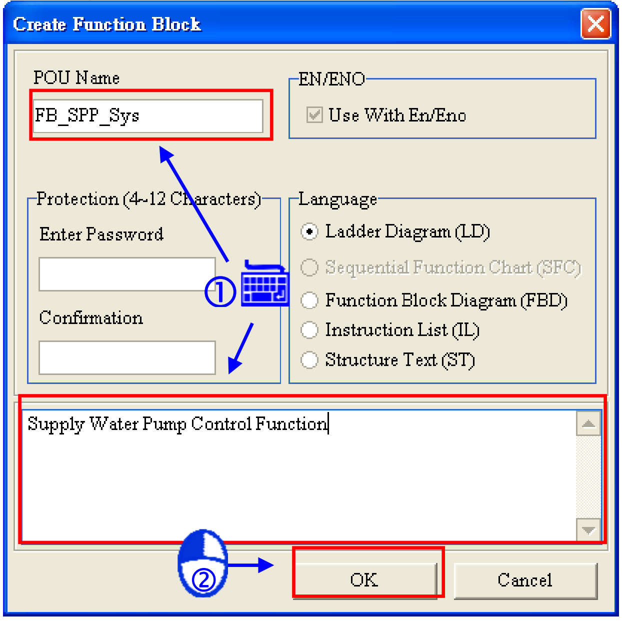
The procedure in this example is to create a function block first. Users also can create the main program first. There is no absolute relation between the function block and the main program. They are actually executed alternately. However, users are recommended to create a function first when the function is used repeatedly.

The control relation among the underground pool, the water tower, and the pump can be represented by a function block. There are usually two water systems in a multi-storey building. If users create the function blocks, they only need to change the variables of the input pins and those of the output pins to complete the second water system.

Users can create a function block which contains the relation among the underground pool, the water tower, and the pump. Find Function Blocks in the window at the left side of the main screen of the software, and click the right mouse button to add a new POU.



Type “FB_SPP_Sys” in the **POU Name** box, and then type “Supply Water Pump Control Function” in the **POU Comment** box. Finally, click **OK**.



Create Function Block

POU Name: FB_SPP_Sys

EN/ENO: ☑ Use With En/Eno

Protection (4~12 Characters)

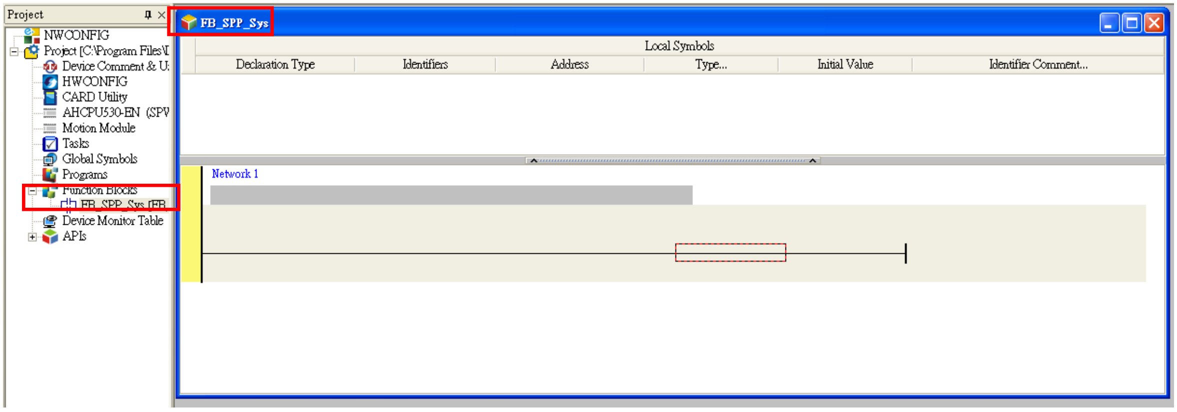
Enter Password

Confirmation

Language:
- ● Ladder Diagram (LD)
- ○ Sequential Function Chart (SFC)
- ○ Function Block Diagram (FBD)
- ○ Instruction List (IL)
- ○ Structure Text (ST)

Supply Water Pump Control Function

Users can see the **FB_SPP_Sys** window on the main screen.



Users need to create the local symbols. In order to make the use of the function block more convenient, the system automatically distributes the addresses to the module, and users are not allowed to type the addresses by themselves. Users surely can use the addresses and the global symbols in the internal program. However, the use of the addresses or the uses of the global symbols will decrease the convenience of using the function block. (If a local symbol is the same as a global symbol, the local symbol used in the function block has high priority.)

The local symbols created are as follows.

Declaration type	Identifier	Data type
VAR_INPUT	Tank_B_LSW	BOOL
VAR_INPUT	Tank_R_LSW	BOOL
VAR_INPUT	Tank_R_HSW	BOOL
VAR_INPUT	Pump_Remote	BOOL
VAR_INPUT	Pump_Run	BOOL
VAR_INPUT	Pump_Trip	BOOL
VAR_IN_OUT	Pump_Auto	BOOL
VAR_IN_OUT	Pump_Man_SW	BOOL
VAR_OUTPUT	Pump_Start	BOOL
VAR	Pump_Out	BOOL

VAR_INPUT

When the program is executed, the value of the external variable is brought into the internal variable. If the value of the corresponding internal variable is altered, it is not transmitted to the external variable. VAR_INPUT is often used if the value of the external variable should not be modified. Most of inputs in this example are digital inputs, and these inputs should not be modified. The modification of the values of these variables affects the execution of the program or the use of the function block. In order to prevent the values of these variables from being modified in the program, the declaration type should be VAR_INPUT.

VAR_IN_OUT

When the program is executed, the value of the external variable is brought into the internal variable. After the program comes to an end, the value is transmitted to the external variable. VAR_IN_OUT is often used if the value of the variable should be modified. Generally speaking, Pump_Auto and Pump_Man_SW in this example are used in the SCADA system to set the control mode of the pump. It seems that VAR_INPUT meets the need. However, users need to switch the control mode of the pump from the automatic mode to the manual mode to stop the command from being outputted when the pump trips. Therefore, the declaration type for these two variables should be VAR_IN_OUT.

VAR_OUTPUT

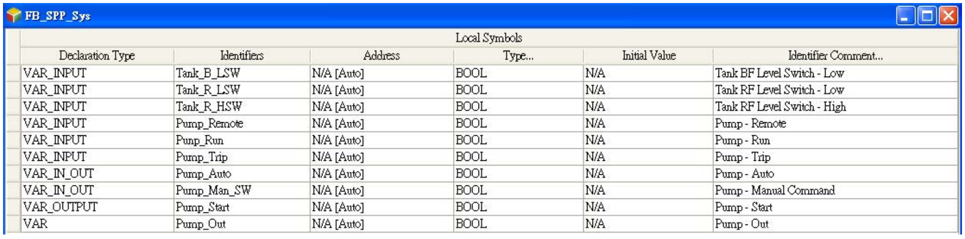
When the program is executed, the value of the external variable is not brought into the internal variable, but the value memorized before is used instead. After the program comes to an end, the value is transmitted to the external variable. Generally speaking, the variable appears at the output of the instruction.

VAR

When the program is executed, VAR is regarded as an internal variable, and the value memorized before is used. Generally speaking, the variable is used as a register when it is used in the program.

If the same variable (function block type) is assigned to the function blocks which are called many times in the program, the initial value of VAR_OUTPUT and that of VAR are not necessarily the same as those used last time.

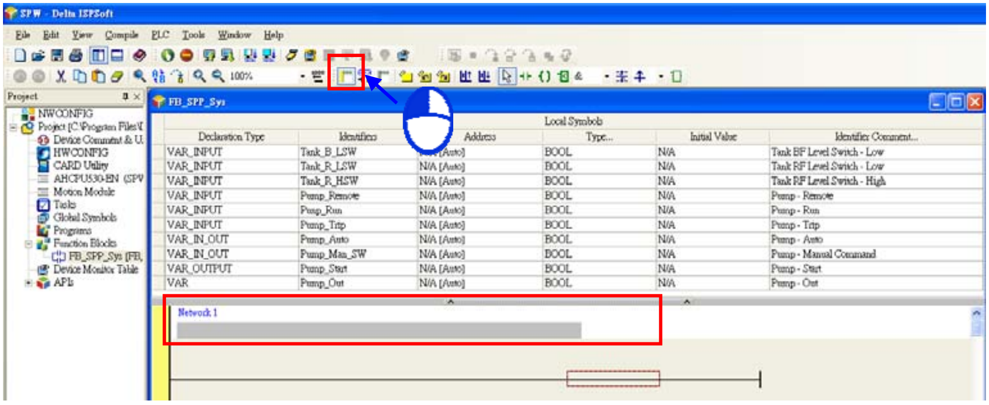
Users can create the global symbols as follows.



Local Symbols

Declaration Type	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR_INPUT	Tank_B_LSW	N/A [Auto]	BOOL	N/A	Tank BF Level Switch - Low
VAR_INPUT	Tank_R_LSW	N/A [Auto]	BOOL	N/A	Tank RF Level Switch - Low
VAR_INPUT	Tank_R_HSW	N/A [Auto]	BOOL	N/A	Tank RF Level Switch - High
VAR_INPUT	Pump_Remote	N/A [Auto]	BOOL	N/A	Pump - Remote
VAR_INPUT	Punp_Run	N/A [Auto]	BOOL	N/A	Pump - Run
VAR_INPUT	Pump_Trip	N/A [Auto]	BOOL	N/A	Pump - Trip
VAR_IN_OUT	Pump_Auto	N/A [Auto]	BOOL	N/A	Pump - Auto
VAR_IN_OUT	Pump_Man_SW	N/A [Auto]	BOOL	N/A	Pump - Manual Command
VAR_OUTPUT	Pump_Start	N/A [Auto]	BOOL	N/A	Pump - Start
VAR	Pump_Out	N/A [Auto]	BOOL	N/A	Pump - Out

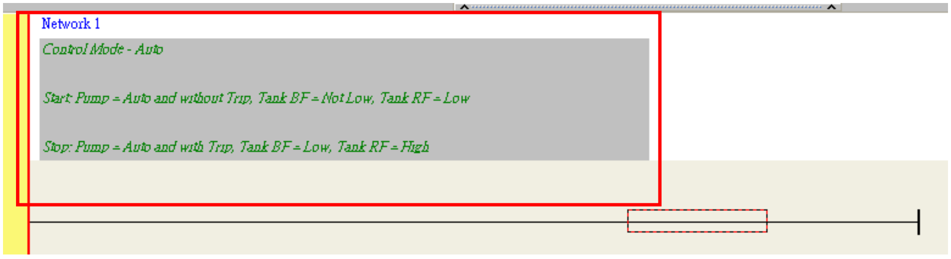
Users can begin to write the program in the function block. In order to improve the convenience of scanning the program, users are recommended to write the comments. If users want to write the network comments, they have to click .



Local Symbols

Declaration Type	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR_INPUT	Tank_B_LSW	[Auto]	BOOL	N/A	Tank BF Level Switch - Low
VAR_INPUT	Tank_R_LSW	N/A [Auto]	BOOL	N/A	Tank RF Level Switch - Low
VAR_INPUT	Tank_R_HSW	N/A [Auto]	BOOL	N/A	Tank RF Level Switch - High
VAR_INPUT	Pump_Remote	N/A [Auto]	BOOL	N/A	Pump - Remote
VAR_INPUT	Punp_Run	N/A [Auto]	BOOL	N/A	Pump - Run
VAR_INPUT	Pump_Trip	N/A [Auto]	BOOL	N/A	Pump - Trip
VAR_IN_OUT	Pump_Auto	N/A [Auto]	BOOL	N/A	Pump - Auto
VAR_IN_OUT	Pump_Man_SW	N/A [Auto]	BOOL	N/A	Pump - Manual Command
VAR_OUTPUT	Pump_Start	N/A [Auto]	BOOL	N/A	Pump - Start
VAR	Pump_Out	N/A [Auto]	BOOL	N/A	Pump - Out

Type the following description as the comment on network 1.



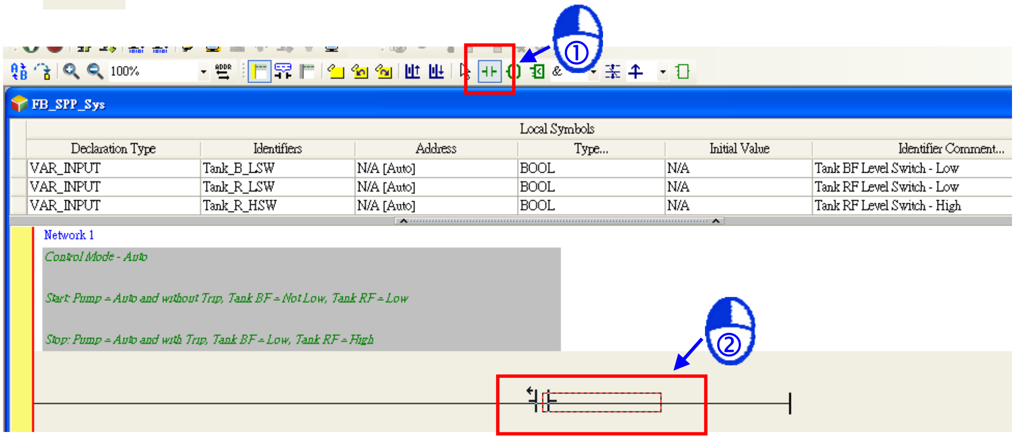
Network 1

Control Mode - Auto

Start: Pump = Auto and without Trip, Tank BF = Not Low, Tank RF = Low

Stop: Pump = Auto and with Trip, Tank BF = Low, Tank RF = High

Users can begin to edit the program code, and the programming language used here is the ladder diagram. Please click ⊣⊢ , and then move the mouse to the red frame. When the cursor becomes ⊣⊢, users can click the left mouse button.



FB_SPP_Sys

Local Symbols

Declaration Type	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR_INPUT	Tank_B_LSW	N/A [Auto]	BOOL	N/A	Tank BF Level Switch - Low
VAR_INPUT	Tank_R_LSW	N/A [Auto]	BOOL	N/A	Tank RF Level Switch - Low
VAR_INPUT	Tank_R_HSW	N/A [Auto]	BOOL	N/A	Tank RF Level Switch - High

Network 1

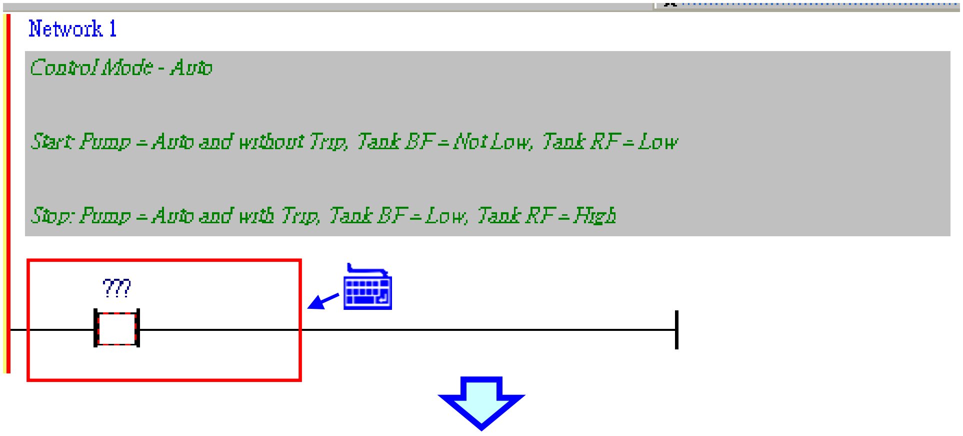
Control Mode - Auto

Start: Pump = Auto and without Trip, Tank BF = Not Low, Tank RF = Low

Stop: Pump = Auto and with Trip, Tank BF = Low, Tank RF = High

Click ???, and type “Pump_Remote”. When “Pump_Remote” is typed, the drop-down list appears. Users can select the item directly, or type the words by themselves. After “Pump_Remote” is typed, press the enter key.

(If the address appears, users can click ADDR to switch the mode. In either mode, users can use the words or the addresses when they write the program.)

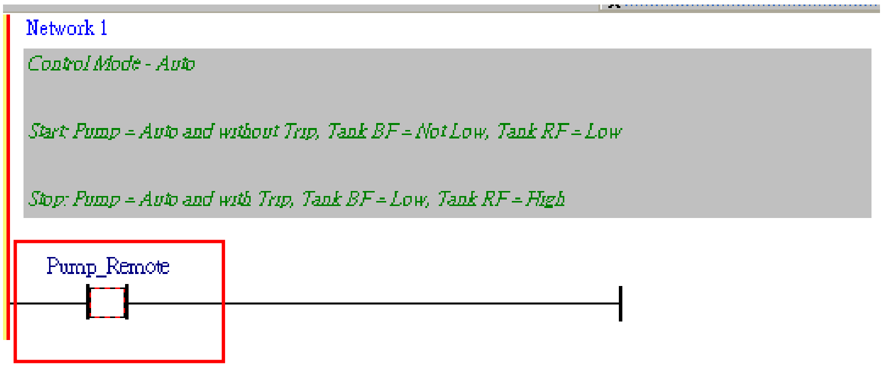


Network 1

Control Mode - Auto

Start: Pump = Auto and without Trip, Tank BF = Not Low, Tank RF = Low

Stop: Pump = Auto and with Trip, Tank BF = Low, Tank RF = High



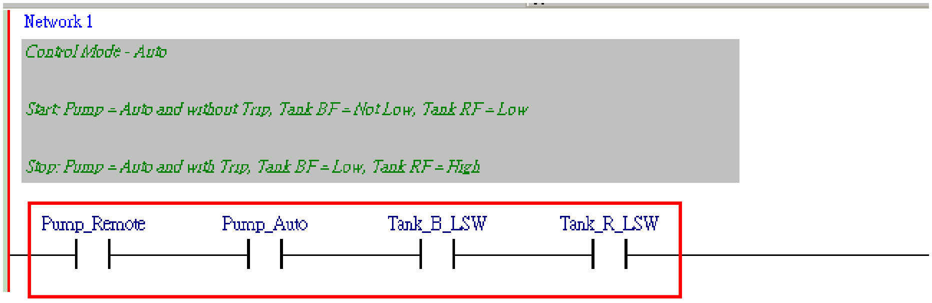
Network 1

Control Mode - Auto

Start: Pump = Auto and without Trip, Tank BF = Not Low, Tank RF = Low

Stop: Pump = Auto and with Trip, Tank BF = Low, Tank RF = High

Users can write the following program in the same way.



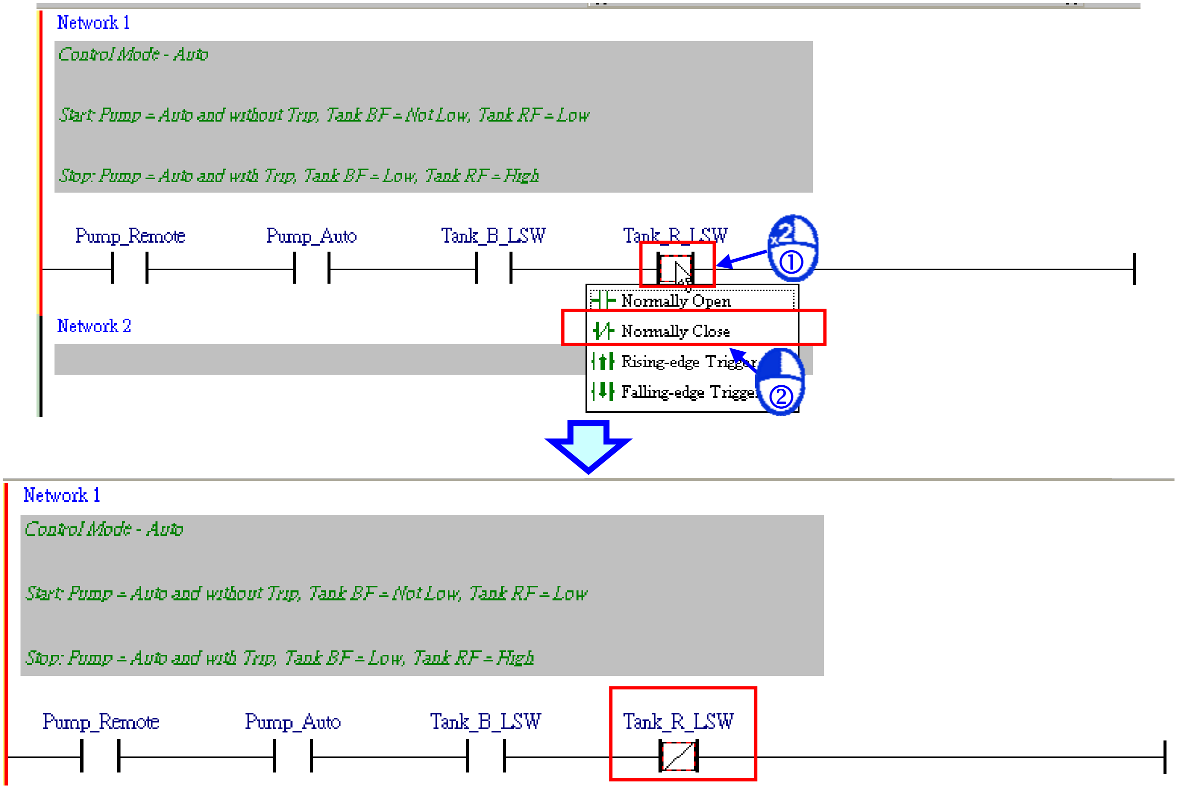
Network 1

Control Mode - Auto

Start: Pump = Auto and without Trip, Tank BF = Not Low, Tank RF = Low

Stop: Pump = Auto and with Trip, Tank BF = Low, Tank RF = High

To meet the condition that there is water in the underground pool and no water in the water tower, the state of Tank_R_LSW should be OFF. Users can double-click the contact marked "Tank_R_LSW", and the drop-down list appears. Select "Normally Closed".



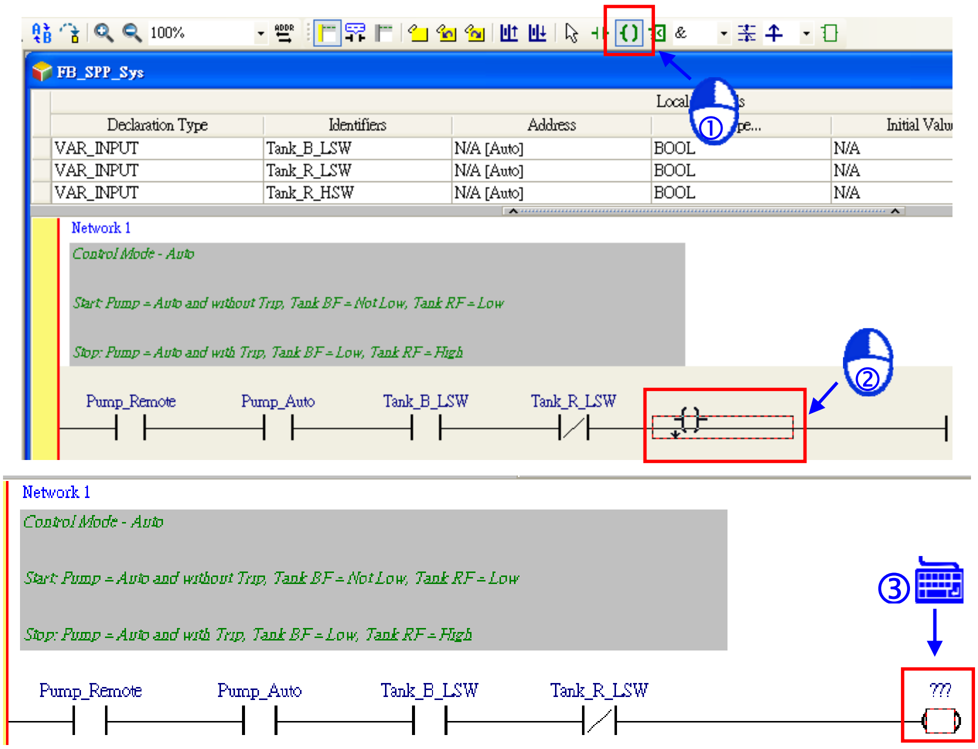
Network 1

Control Mode - Auto

Start: Pump = Auto and without Trip, Tank BF = Not Low, Tank RF = Low

Stop: Pump = Auto and with Trip, Tank BF = Low, Tank RF = High

Please click { }, and then move the mouse to the red frame.



Declaration Type	Identifiers	Address	Type	Initial Value
VAR_INPUT	Tank_B_LSW	N/A [Auto]	BOOL	N/A
VAR_INPUT	Tank_R_LSW	N/A [Auto]	BOOL	N/A
VAR_INPUT	Tank_R_HSW	N/A [Auto]	BOOL	N/A

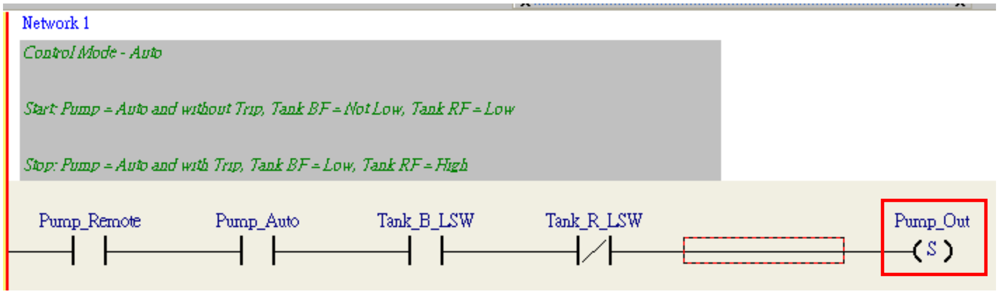
Network 1

Control Mode - Auto

Start: Pump = Auto and without Trip, Tank BF = Not Low, Tank RF = Low

Stop: Pump = Auto and with Trip, Tank BF = Low, Tank RF = High

Click ???, and type “Pump_Out”. Users can double-click the coil marked “Pump_Out”, and the drop-down list appears. Select “Set”.



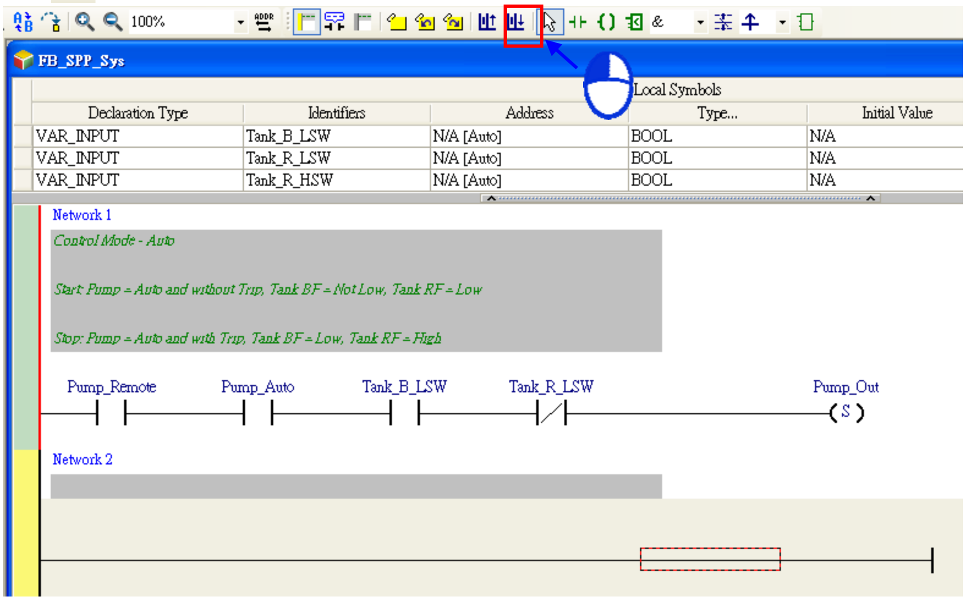
Network 1

Control Mode - Auto

Start: Pump = Auto and without Trip, Tank BF = Not Low, Tank RF = Low

Stop: Pump = Auto and with Trip, Tank BF = Low, Tank RF = High

Please click , to add a new netwok.



Declaration Type	Identifiers	Address	Type...	Initial Value
VAR_INPUT	Tank_B_LSW	N/A [Auto]	BOOL	N/A
VAR_INPUT	Tank_R_LSW	N/A [Auto]	BOOL	N/A
VAR_INPUT	Tank_R_HSW	N/A [Auto]	BOOL	N/A

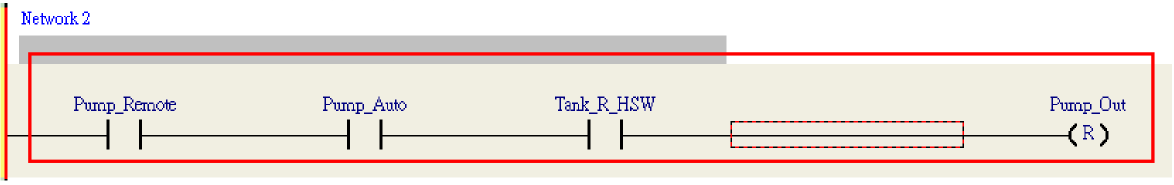
Network 1

Control Mode - Auto

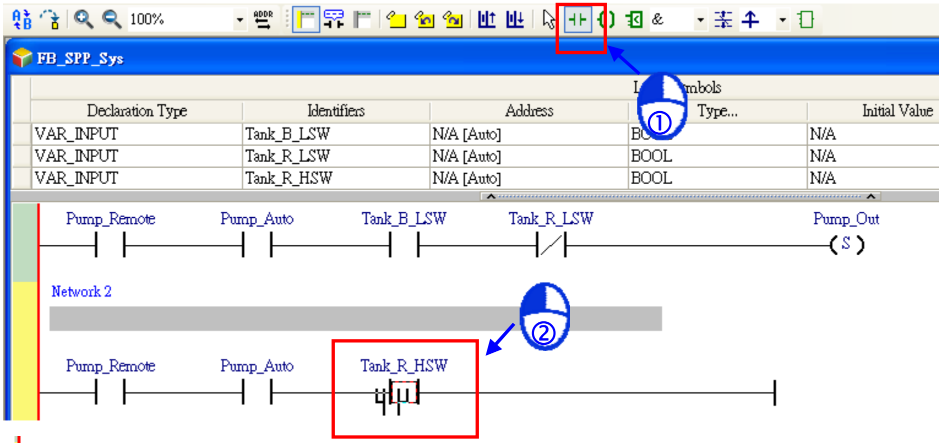
Start: Pump = Auto and without Trip, Tank BF = Not Low, Tank RF = Low

Stop: Pump = Auto and with Trip, Tank BF = Low, Tank RF = High

Users can write the following program in the same way.

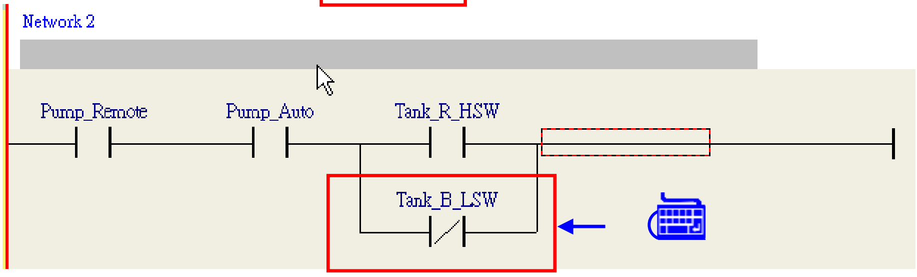


To meet the condition that there is no water in the underground pool, or the condition that the pump trips, users need to click ┤├, and move the mouse to the position under a contact. When ┤├ appears, users can click the left mouse button. Then, users need to click ???, and type “Tank_B_LSW”. Finally, users can double-click the coil marked “Tank_B_LSW”. When the drop-down list appears, select “Set”.

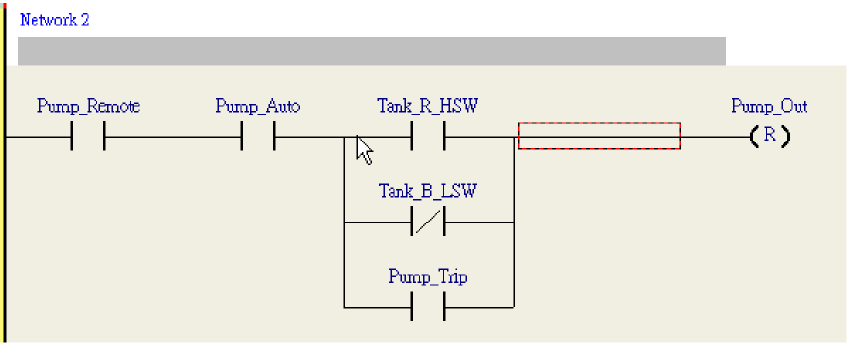


FB_SPP_Sys

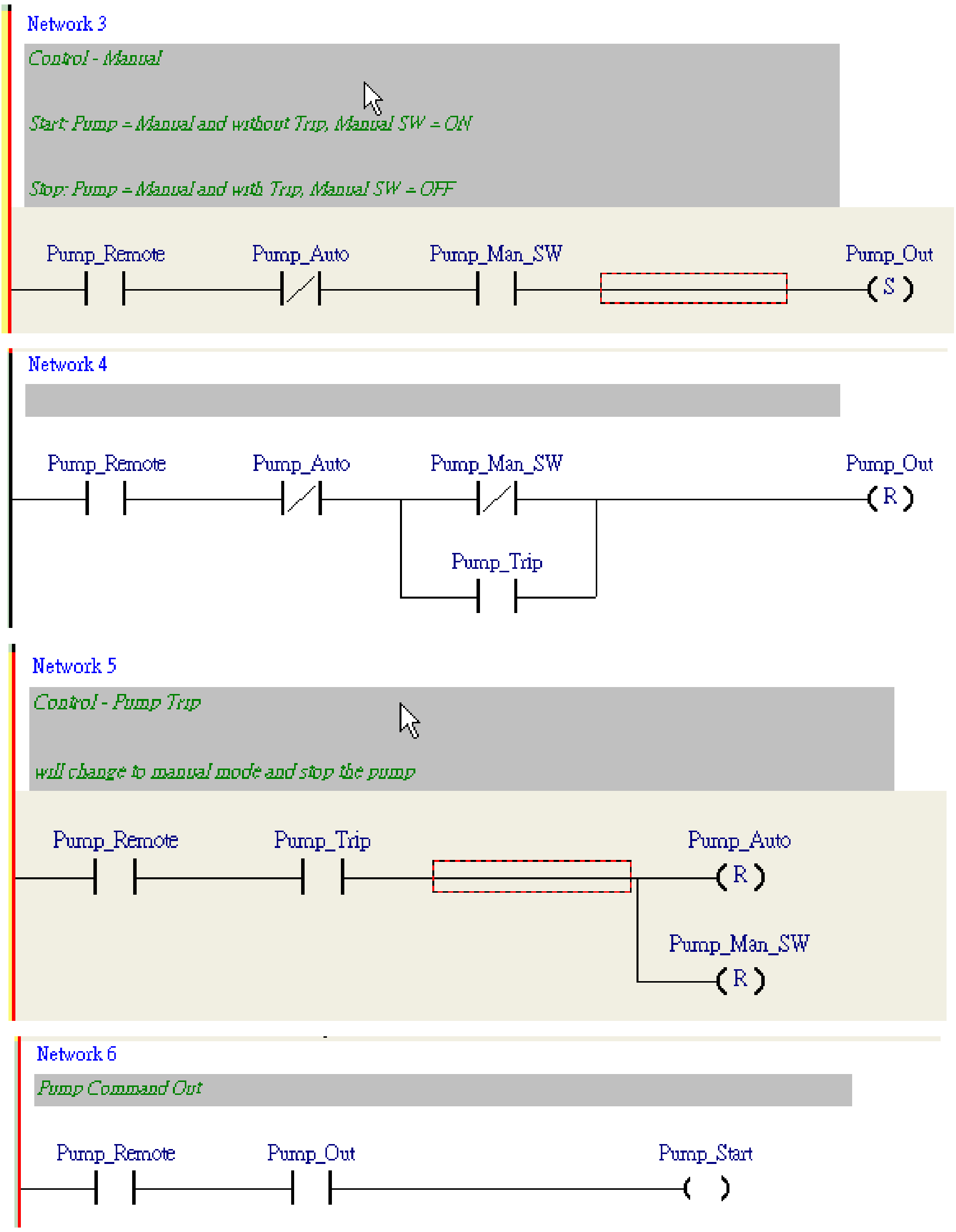
Declaration Type	Identifiers	Address	Type...	Initial Value
VAR_INPUT	Tank_B_LSW	N/A [Auto]	BO	N/A
VAR_INPUT	Tank_R_LSW	N/A [Auto]	BOOL	N/A
VAR_INPUT	Tank_R_HSW	N/A [Auto]	BOOL	N/A



Users can write the following program in the same way.



So far the automatic program control is complete. Please write the following program in the same way.



Network 3

Control - Manual

Start: Pump = Manual and without Trip, Manual SW = ON

Stop: Pump = Manual and with Trip, Manual SW = OFF

Network 4

Network 5

Control - Pump Trip

will change to manual mode and stop the pump

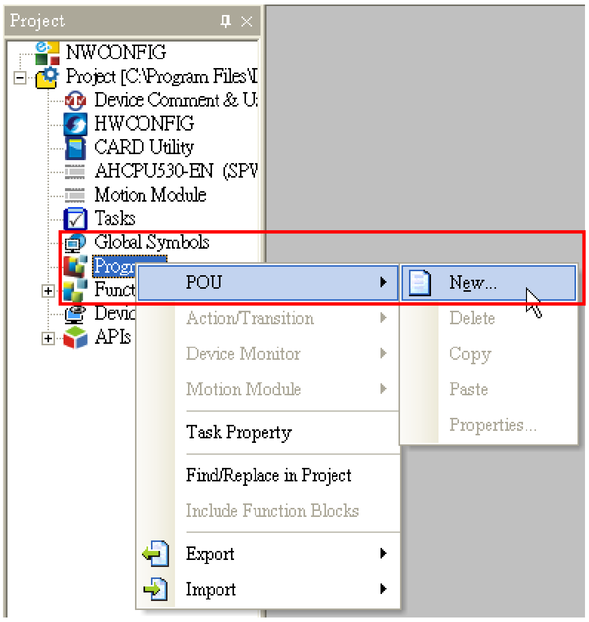
Network 6

Pump Command Out

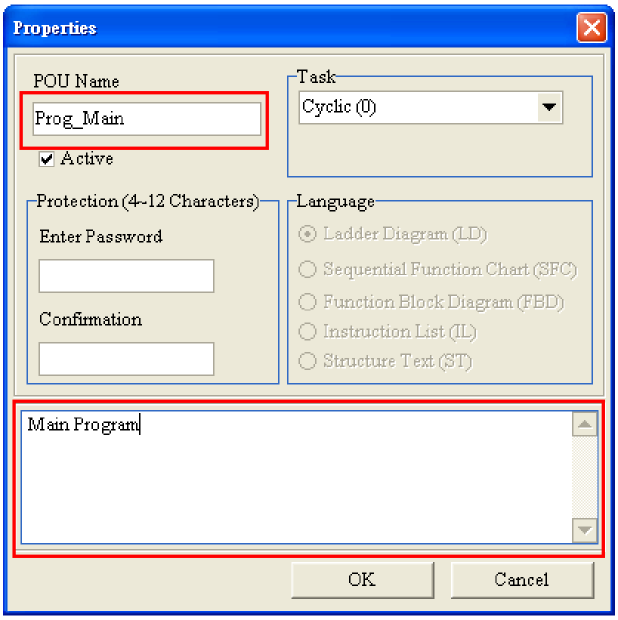
Finally, users need to save the file, and the function block is created.

2.9 Creating Main Programs

After the function block is created, users can create the main program. First, find Programs in the window at the left side of the main screen of the software. Then, click the right mouse button to add a new POU.



Type "Prog_Main" in the **POU Name** box, and then type "Main program" in the **POU Comment** box. Finally, click **OK**.


Properties

POU Name: Prog_Main

☑ Active

Task: Cyclic (0)

Protection (4~12 Characters)

Enter Password

Confirmation

Language

● Ladder Diagram (LD)

○ Sequential Function Chart (SFC)

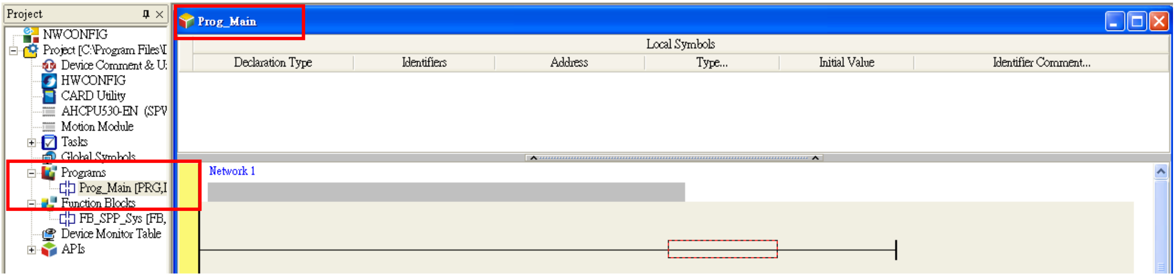
○ Function Block Diagram (FBD)

○ Instruction List (IL)

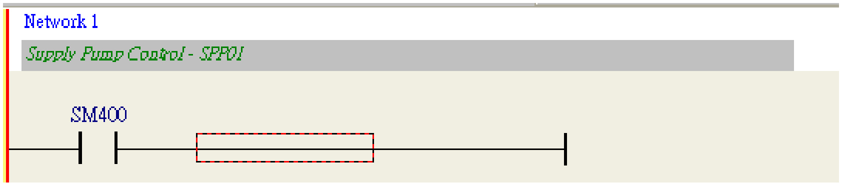
○ Structure Text (ST)

Main Program

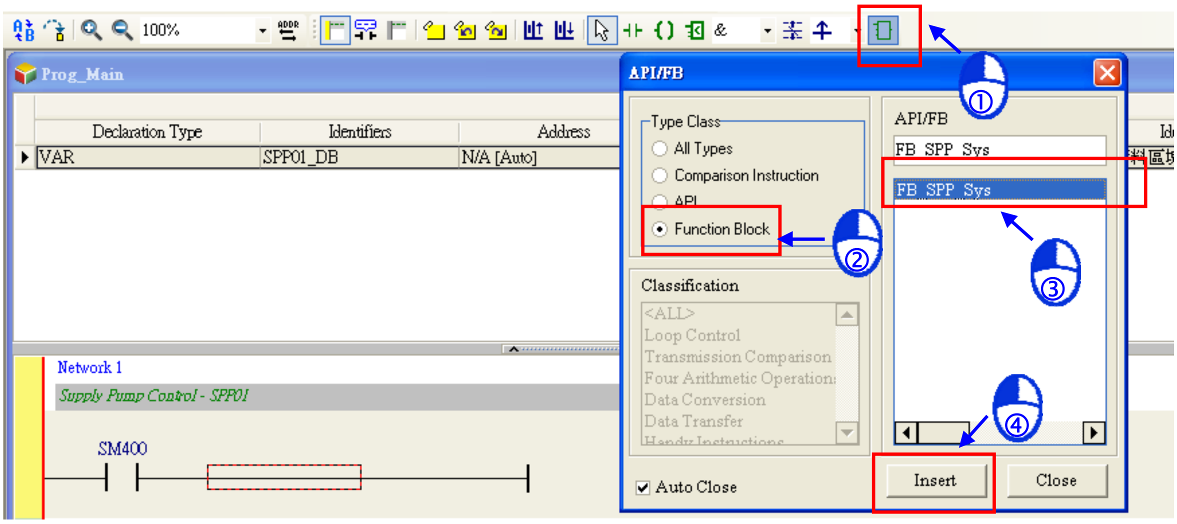
Users can see the **Prog_Main** window on the main screen.



The difference between the function block and the main program is that the function block has to be called before it is executed while the main program is executed directly. The creation of the local symbols for the main program is omitted in this example. Please write the following program by means of the skills learned previously.

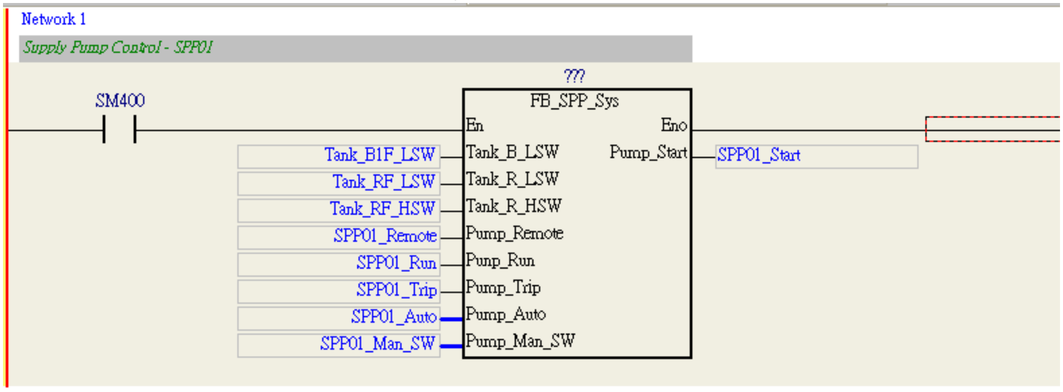


Users can prepare to call the function block. First, click to open the **API/FB** window. Then, select **Function Block** in the **Type Class** box, and select “FB_SPP_Sys” in the **API/FB** box. Finally, click **Insert**.

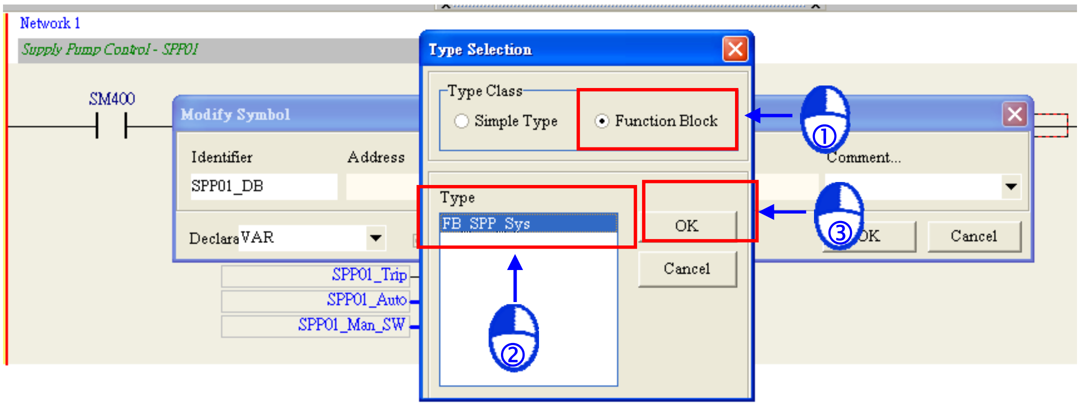


Declaration Type	Identifiers	Address
VAR	SPP01_DB	N/A [Auto]

After the function block interface appears, type the identifiers as shown below.



Users need to create a data block for the function block. The data block can be created as a global symbol or a local symbol, but it is created as a local symbol in the main program here. Notice that users need to select **Function Block** in the **Type Class** box, and select “FB_SPP_Sys” in the **Type** box. Please type “SPP01_DB” in the **Identifier** box, and type “SPP01 Data Block” in the **Comment…** box, as show below.


Network 1

Supply Pump Control - SPP01

Modify Symbol

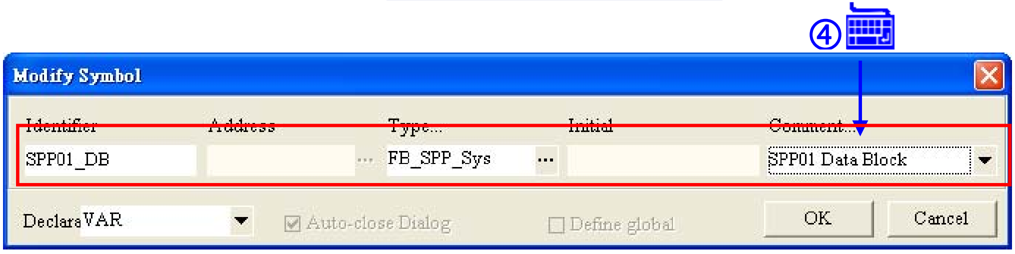
Identifier: SPP01_DB

Declare: VAR

Type Selection

Type Class: ○ Simple Type ● Function Block

Type: FB_SPP_Sys

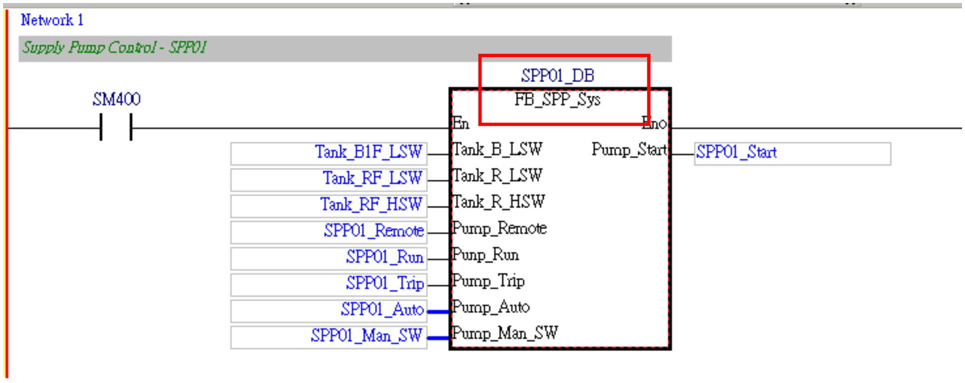

Modify Symbol

Identifier	Address	Type...	Initial	Comment...
SPP01_DB		FB_SPP_Sys		SPP01 Data Block

Declare: VAR

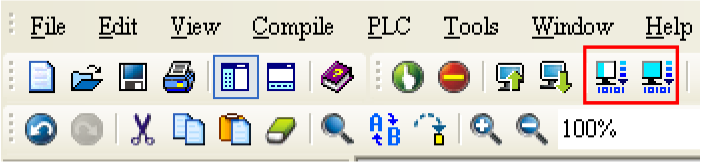
☑ Auto-close Dialog □ Define global

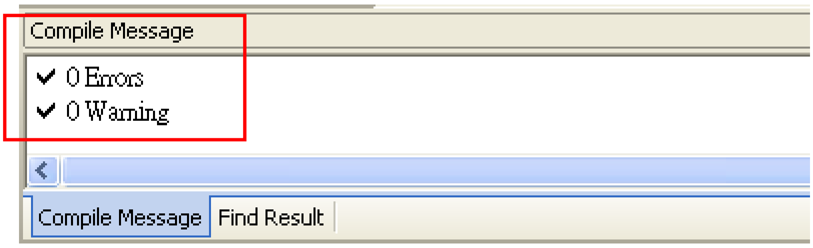
After the local symbol is created, the identifier can be given to the function block. (The data type should be the same with the name of the function block.)



After the file is saved, the writing of the program is complete.

Users can click to check the syntax of the current program, and click to compile the project. The compiling of the project is different from the checking of the syntax in that the former involves not only the function of checking the syntax, but also the function of compiling all programs and the contents of the function block.



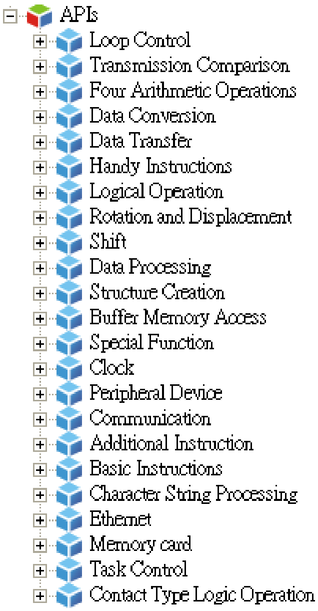


Compile Message

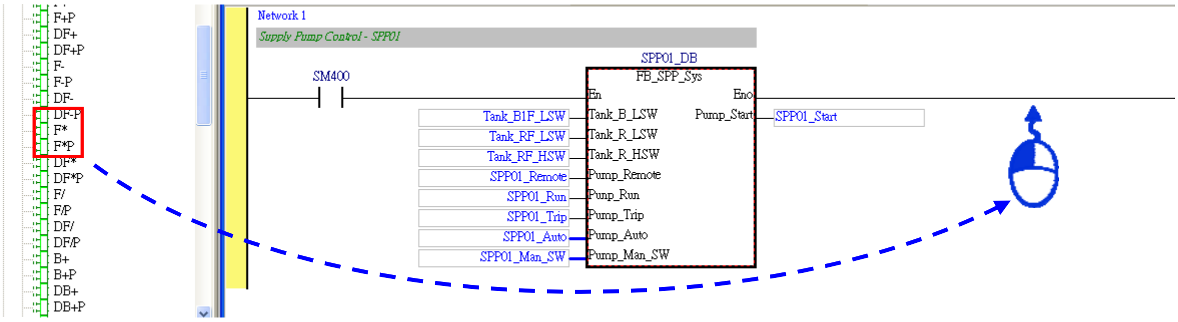
✓ 0 Errors

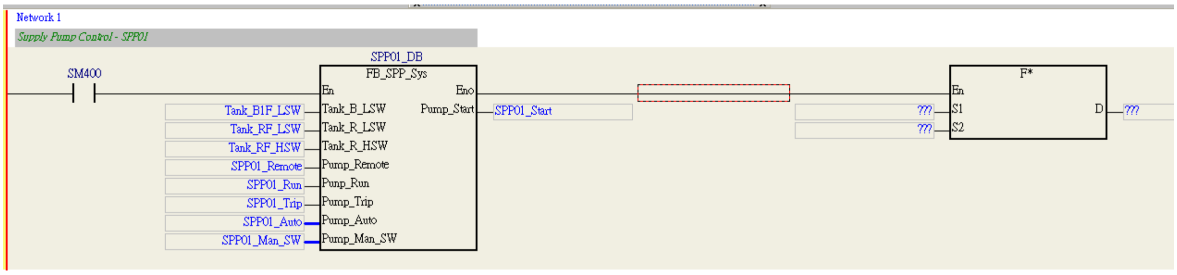
✓ 0 Warning

Note: In order to help users understand the control program, the example provided here is simple. The actual control program is much more complex because the functions related to judging the start failure and the conversion from the local control to the remote control have to be taken into account. Users need to use the applied instructions in the window at the left side of the main screen to write a program for these functions. (Please refer to ISPSoft User Manual for more information.)

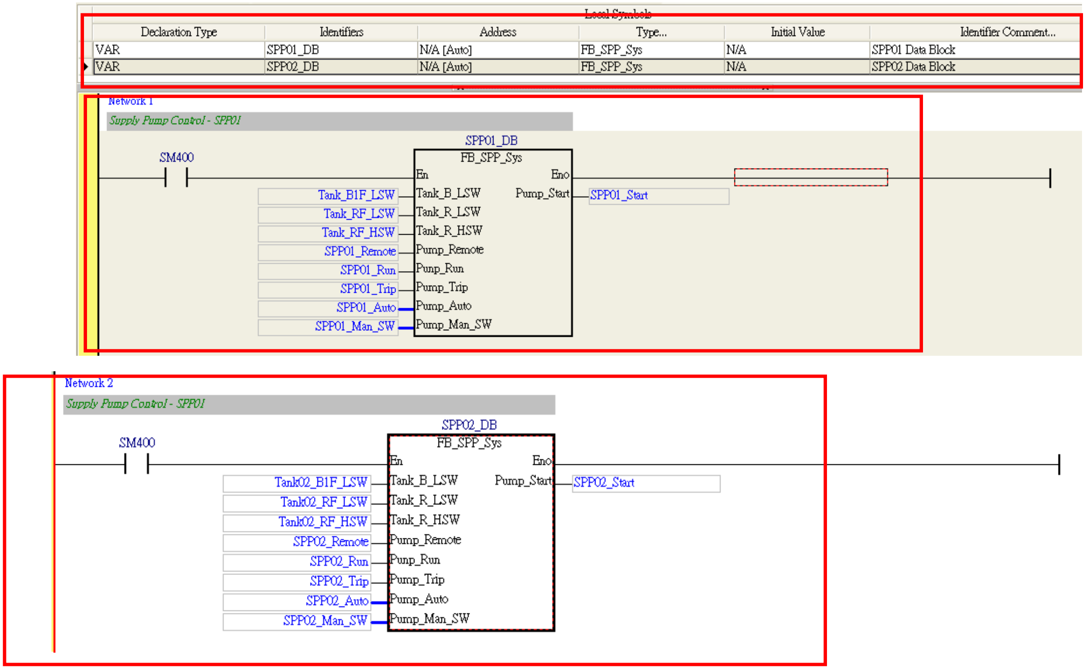


APIs
- Loop Control
- Transmission Comparison
- Four Arithmetic Operations
- Data Conversion
- Data Transfer
- Handy Instructions
- Logical Operation
- Rotation and Displacement
- Shift
- Data Processing
- Structure Creation
- Buffer Memory Access
- Special Function
- Clock
- Peripheral Device
- Communication
- Additional Instruction
- Basic Instructions
- Character String Processing
- Ethernet
- Memory card
- Task Control
- Contact Type Logic Operation





If users need to create the second system, they only need to call “FB_SPP_Sys” again, and type the identifier. Please notice that the same data block can not be used in the two systems unless user consider that the value of VAR_OUTPUT and that of VAR of in the first system does not affect those in the second system. Otherwise, please create a new identifier for the second system, as shown below.



Local Symbols

Declaration Type	Identifiers	Address	Type...	Initial Value	Identifier Comment...
VAR	SPP01_DB	N/A [Auto]	FB_SPP_Sys	N/A	SPP01 Data Block
VAR	SPP02_DB	N/A [Auto]	FB_SPP_Sys	N/A	SPP02 Data Block

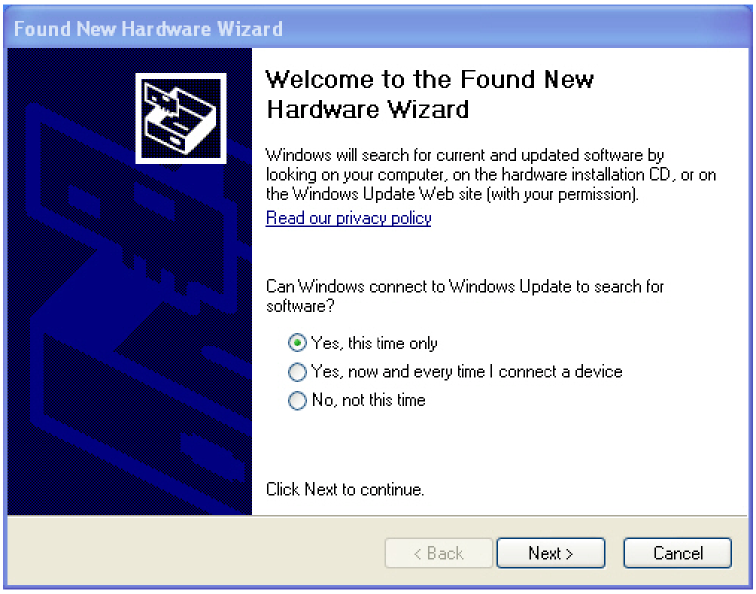
MEMO

Chapter 3 Downloading and Monitoring the Program

Table of Contents

After the writing of the program is complete, users can prepare to download the program. In order to increase the functions and the convenience, the latest software package is divided into COMMGR and ISPSoft. In addition to preparing the hardware, supplying power to the CPU module, and installing the USB driver, users need to set the communication software to communicate with the CPU module.

If the computer is not connected to the USB port on the CPU module, please connect the CPU module to the computer with an USB cable. The following window will appear.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

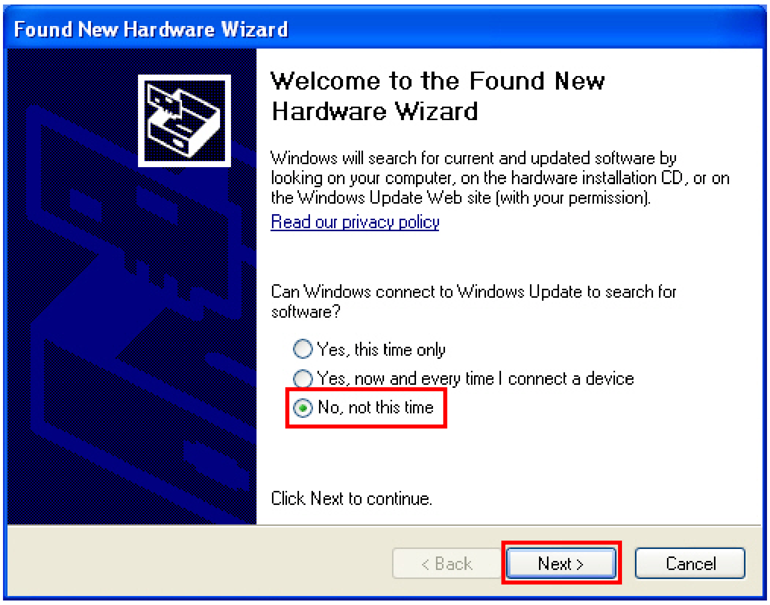
Read our privacy policy

Can Windows connect to Windows Update to search for software?

● Yes, this time only
○ Yes, now and every time I connect a device
○ No, not this time

Click Next to continue.

Please complete the setting according to the steps below.



Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

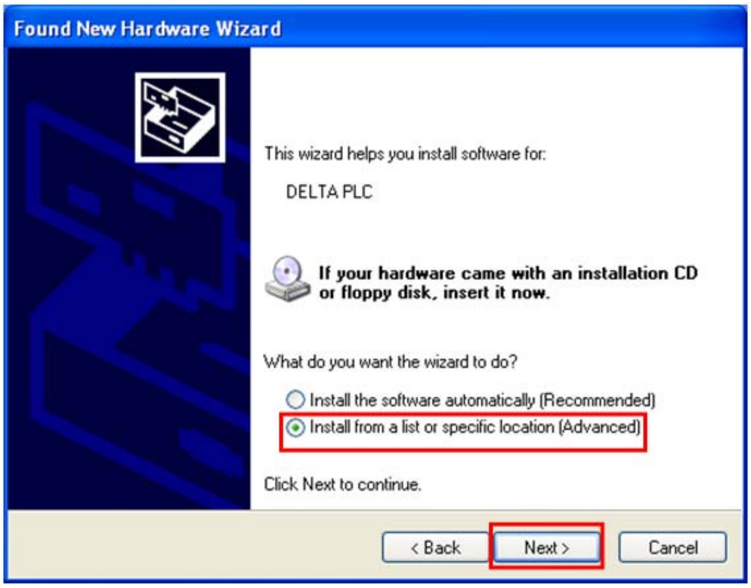
Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

Read our privacy policy

Can Windows connect to Windows Update to search for software?

○ Yes, this time only
○ Yes, now and every time I connect a device
● No, not this time

Click Next to continue.



Found New Hardware Wizard

This wizard helps you install software for:

DELTA PLC

If your hardware came with an installation CD or floppy disk, insert it now.

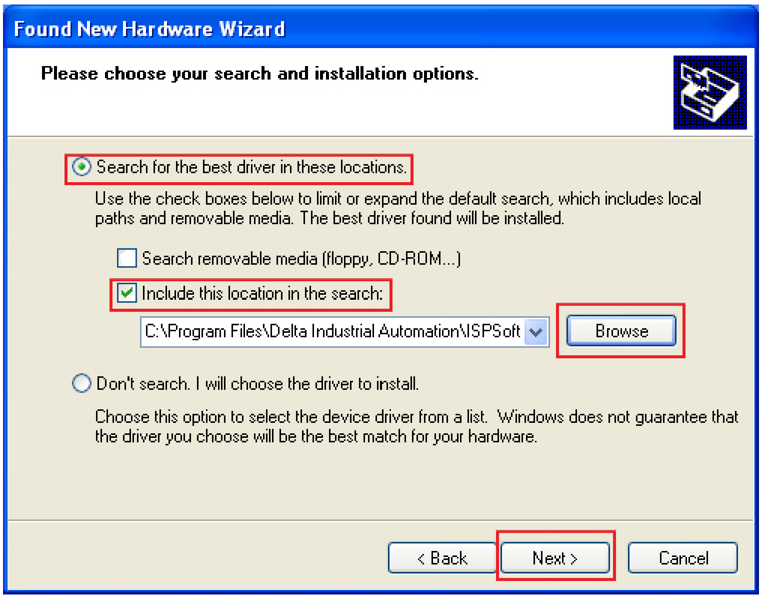
What do you want the wizard to do?

○ Install the software automatically (Recommended)

● Install from a list or specific location (Advanced)

Click Next to continue.

The path C:\Program Files\Delta Industrial Automation\ISPSoftx.xx\drivers\USB shown in the picture below is the folder where the software is installed. If the path is modified, please select the folder where the software will be installed.



Found New Hardware Wizard

Please choose your search and installation options.

● Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

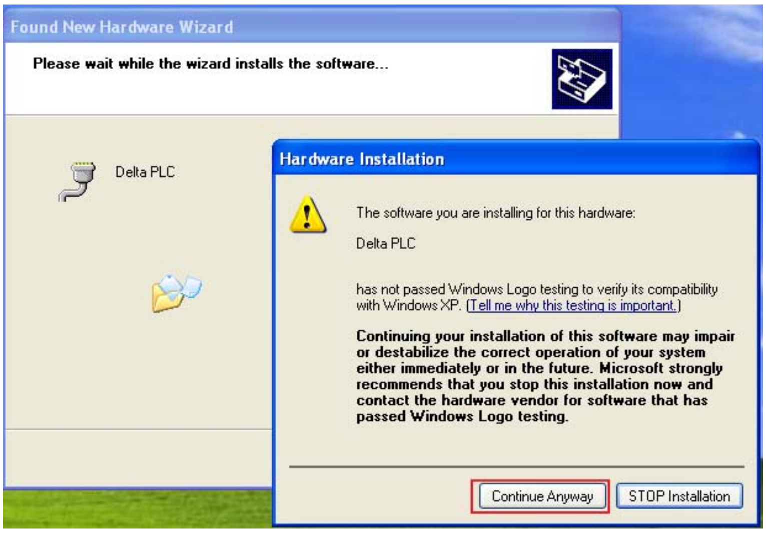
□ Search removable media (floppy, CD-ROM...)

☑ Include this location in the search:

C:\Program Files\Delta Industrial Automation\ISPSoft

○ Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.



Found New Hardware Wizard

Please wait while the wizard installs the software...

Delta PLC

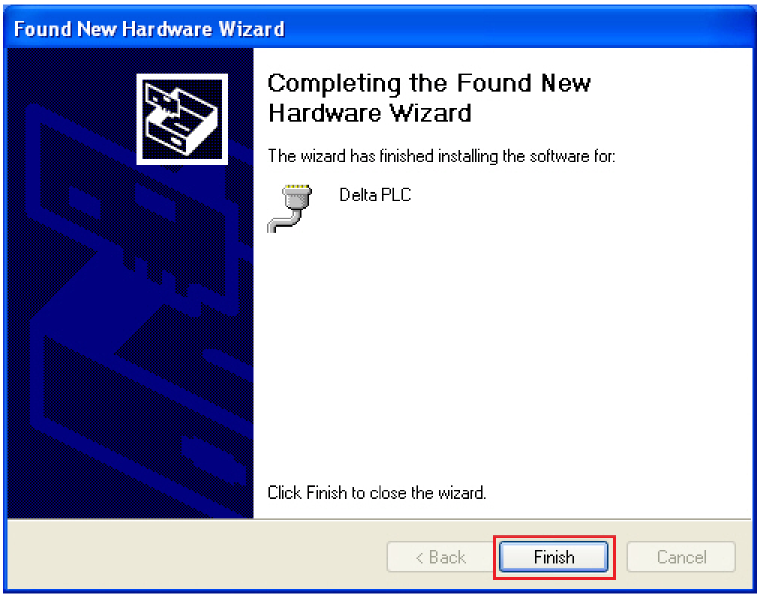
Hardware Installation

The software you are installing for this hardware:

Delta PLC

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.



Found New Hardware Wizard

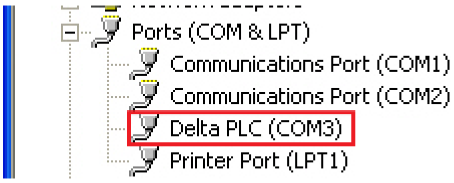
Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

Delta PLC

Click Finish to close the wizard.

After the software is installed, users can see the USB driver in the **Device Manager** window. (The COM number of the USB may vary from computer to computer.)



Ports (COM & LPT)
- Communications Port (COM1)
- Communications Port (COM2)
- Delta PLC (COM3)
- Printer Port (LPT1)

3.1 Setting COMMGR

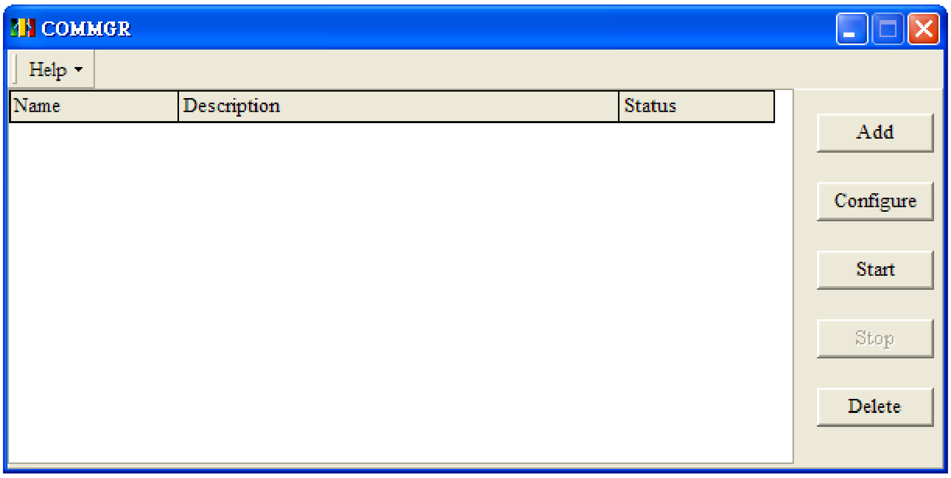
3.1.1 Enabling COMMGR

After the software is installed, the shortcut for COMMGR automatically appears in the notification area in the operating system. Whenever the computer is turned on, users can see located at the far right of the taskbar. If users do not see the icon, they can enable the software manually. The shortcut to enabling the software is Start→Programs→ Delta Industrial

Automation→Communication→COMMGR→COMMGR.

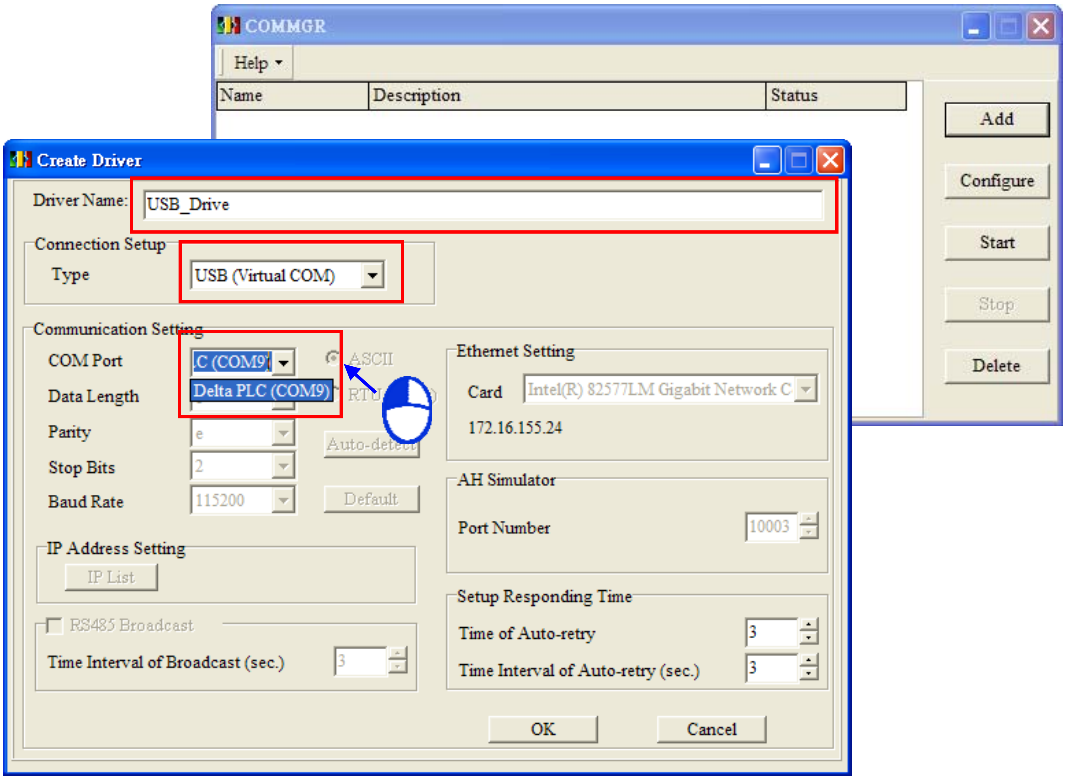
3.1.2 Opening COMMGR

After COMMGR is enabled, users can double-click to open the setting interface below.



3.1.3 Setting COMMGR

After the interface is opened, users can begin to set the communication. Owing to the fact that the computer is connected to the CPU module with an USB cable, an USB driver is created. First, click **Add**. Then, complete the following setting.



Driver Name: USB_Drive

Type: USB (Virtual COM)

COM Port: Delta PLC (COM9)

Data Length

Parity: e

Stop Bits: 2

Baud Rate: 115200

□ RS485 Broadcast

Time Interval of Broadcast (sec.): 3

Card: Intel(R) 82577LM Gigabit Network C

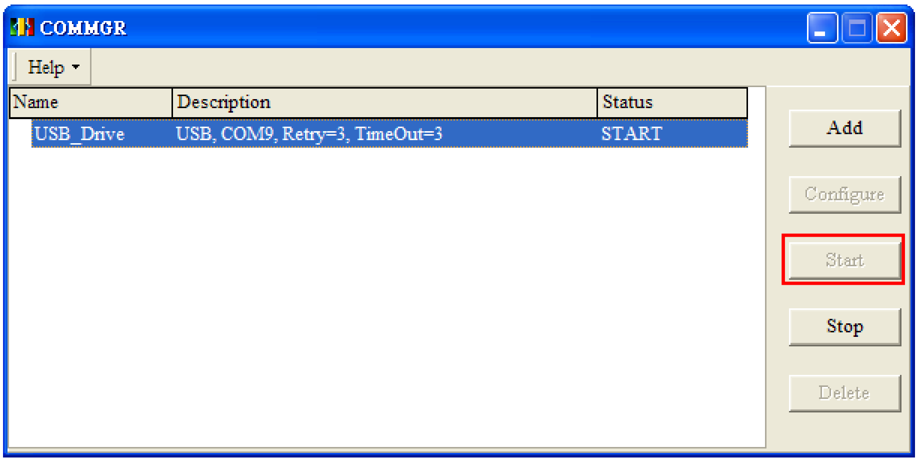
172.16.155.24

Port Number: 10003

Time of Auto-retry: 3

Time Interval of Auto-retry (sec.): 3

After the setting is complete, the driver appears in the **COMMGR** window. Please click **Start** to start the driver.



COMMGR

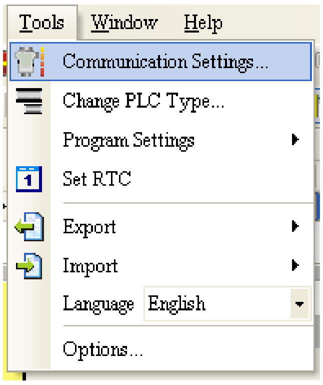
Name	Description	Status
USB_Drive	USB, COM9, Retry=3, TimeOut=3	START

3.2 Downloading the Program

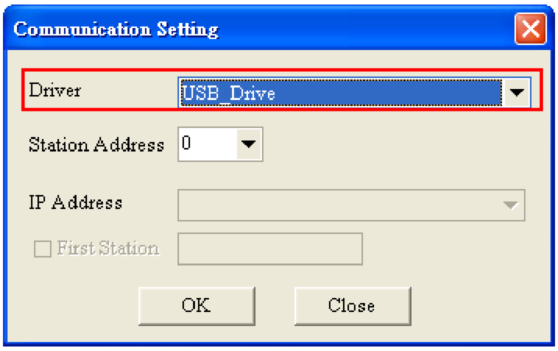
After the editing above is complete, users need to download the parameter and the program to the CPU module. In this example, the data which is downloaded to the PLC is the hardware configuration, the CPU parameter, and the program.

3.2.1 Setting the Communication

After the setting of COMMGR is complete, users can set the communication in ISPSoft. First, find **Communication Settings…** on the Tools menu.



After the Communication Setting window appears, please select **USB_Drive** in the **Driver** box, and click **OK**.



Communication Setting

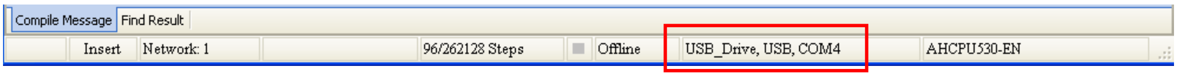
Driver: USB_Drive

Station Address: 0

IP Address:

□ First Station

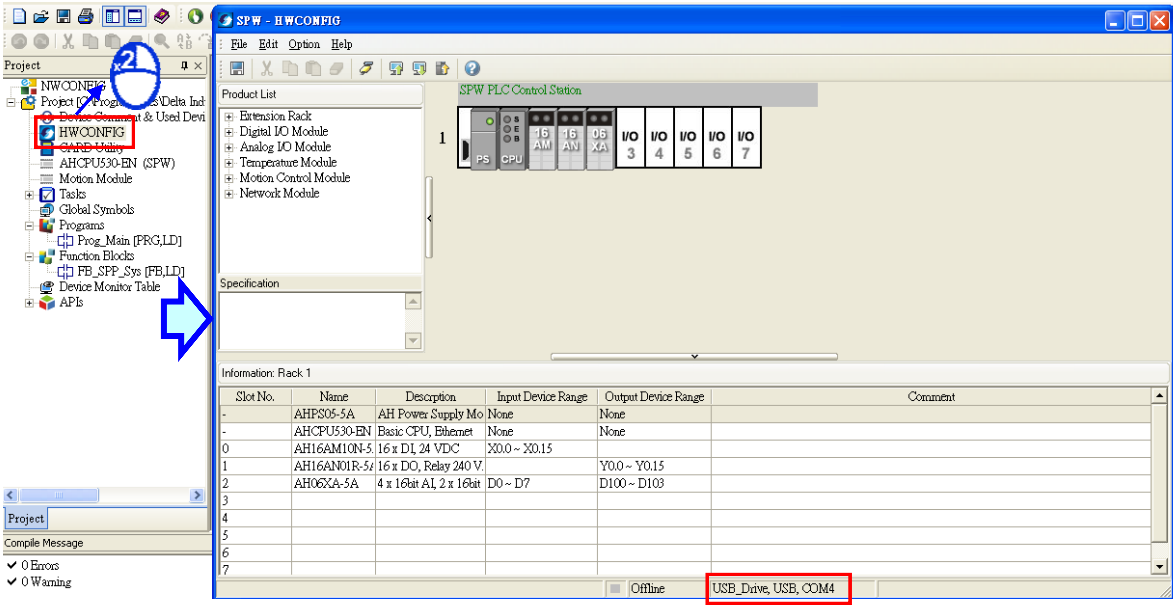
Users can see the information about the communication setting in the lower right corner of the screen.



Insert | Network: 1 | 96/262128 Steps | Offline | USB_Drive, USB, COM4 | AHCPU530-EN

3.2.2 Downloading the Hardware Configuration

The hardware configuration is downloaded first. Open the **SPW-HWCONFIG** window as shown below.

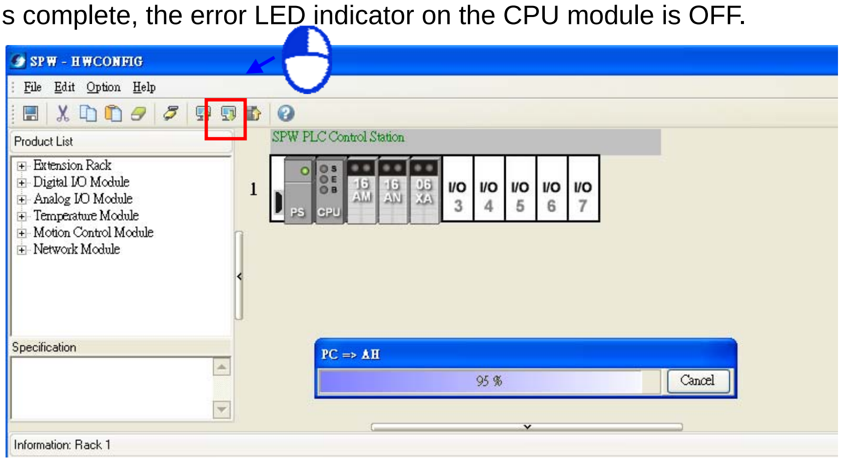


Information: Rack 1

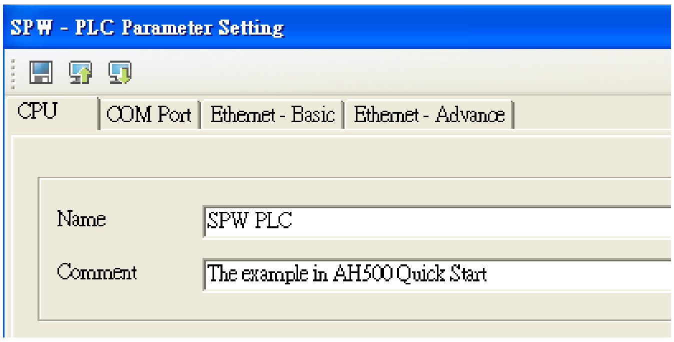
Slot No.	Name	Description	Input Device Range	Output Device Range	Comment
-	AHPS05-5A	AH Power Supply Mo	None	None	
-	AHCPU530-EN	Basic CPU, Ethernet	None	None	
0	AH16AM10N-5.	16 x DI, 24 VDC	X0.0 ~ X0.15		
1	AH16AN01R-5/	16 x DO, Relay 240 V.		Y0.0 ~ Y0.15	
2	AH06XA-5A	4 x 16bit AI, 2 x 16bit	D0 ~ D7	D100 ~ D103	
3					
4					
5					
6					
7					

Offline | USB_Drive, USB, COM4

Click to download the hardware configuration. After the downloading of hardware configuration is complete, the error LED indicator on the CPU module is OFF.



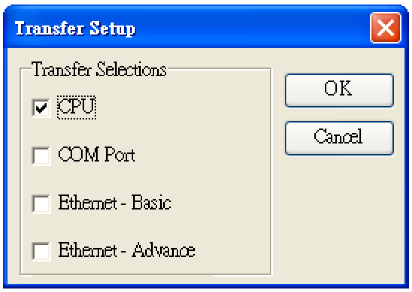
Double-click the CPU module to open the **SPW-PLC Parameter Setting** window.



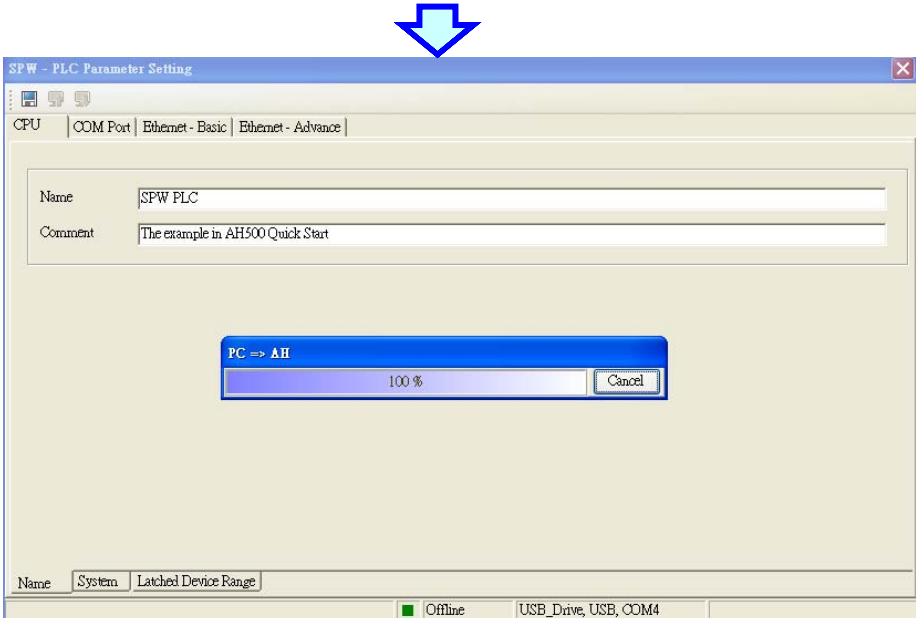
Name: SPW PLC

Comment: The example in AH500 Quick Start

3 Click to open the **Transfer Setup** window. Check the **CPU** box, and click **OK** to download the parameter.



☑ CPU
□ COM Port
□ Ethernet - Basic
□ Ethernet - Advance



Name: SPW PLC

Comment: The example in AH500 Quick Start

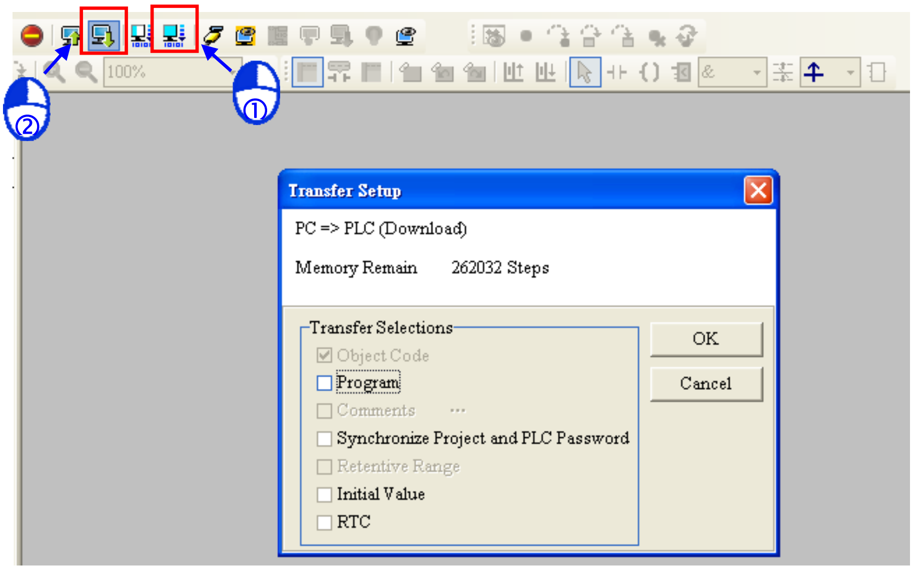
100 %

Offline | USB_Drive, USB, COM4

After the downloading of the parameter is complete, please close the **SPW-HWCONFIG** window. (If users use a network module or a motion control module, they have to open the corresponding software to download the related parameter.)

3.2.3 Downloading the Program

After the hardware configuration is downloaded, users can begin to download the program. The project has been compiled previously. If the program is modified, users can click to check the syntax. After the compiling of the program is successful, users can click to download the program.



Transfer Setup

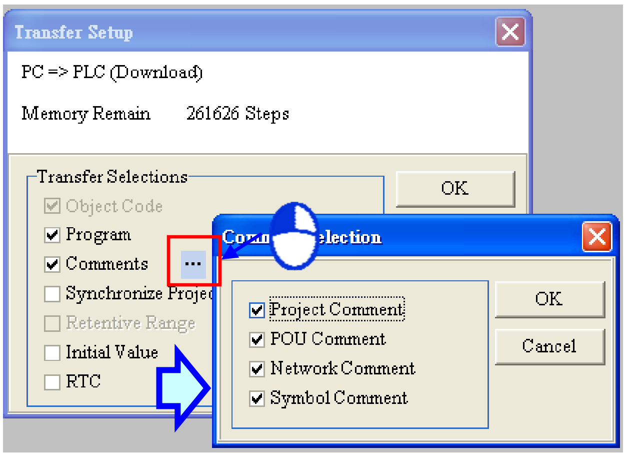
PC => PLC (Download)

Memory Remain 262032 Steps

Transfer Selections

- ☑ Object Code
- □ Program
- □ Comments ...
- □ Synchronize Project and PLC Password
- □ Retentive Range
- □ Initial Value
- □ RTC

There are several transfer options in the **Transfer Setup** window. The object code is the necessary condition for the operation of the CPU module, and therefore the **Object Code** box is checked. In order to maintain the program, the **Program** box and the **Comment**s box are checked in this example. The current project will be backupped in the CPU module, and the program will not be lost. Besides, if users want to modify the program on the spot, and forget to bring the original program, they can upload the original program from the CPU module to the computer to modify the program.



Transfer Setup

PC => PLC (Download)

Memory Remain 261626 Steps

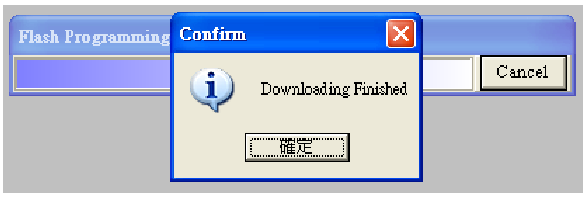
Transfer Selections

- ☑ Object Code
- ☑ Program
- ☑ Comments ...
- □ Synchronize Projec
- □ Retentive Range
- □ Initial Value
- □ RTC

Comm election

- ☑ Project Comment
- ☑ POU Comment
- ☑ Network Comment
- ☑ Symbol Comment

After users click **OK**, the downloading of the program is complete.



Flash Programming

Confirm

Downloading Finished

Finally, check the I/O LED indicator on the module and the status of the equipment. After users make sure that the whole system can be tested, they have to switch the CPU module on so that the CPU module starts to run.

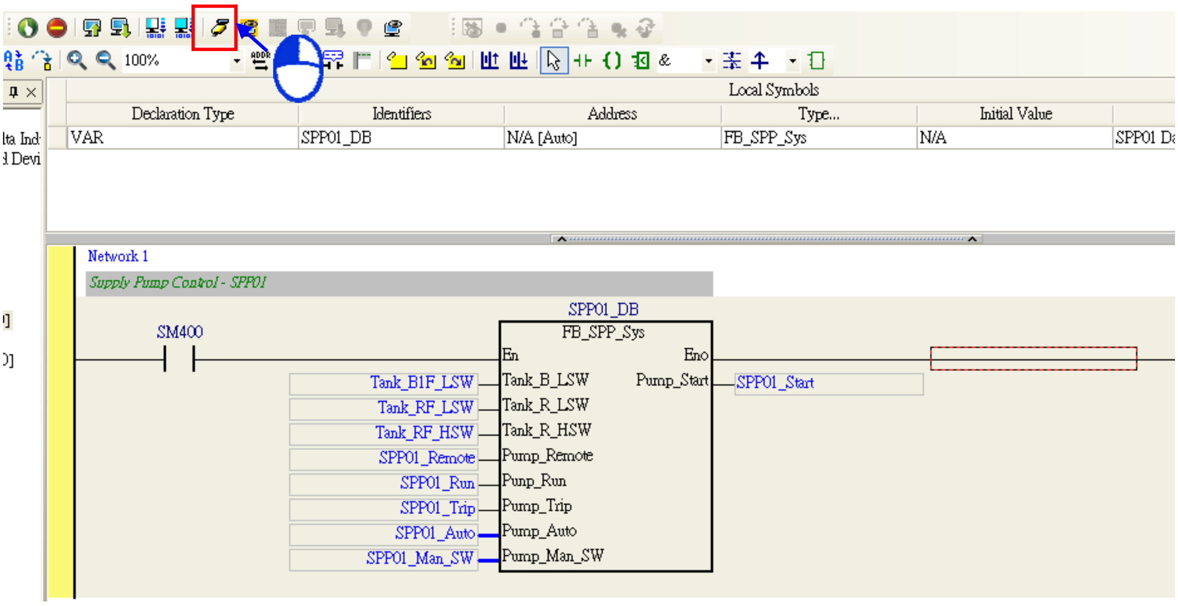
3.3 Monitoring and Debugging the Program

3.3.1 Monitoring the Program

When the program is executed, users can monitor the program to understand the current logical control state, or modify the values in some devices to test the system. The operation of the program monitoring is as follows.

- **Monitoring the program**

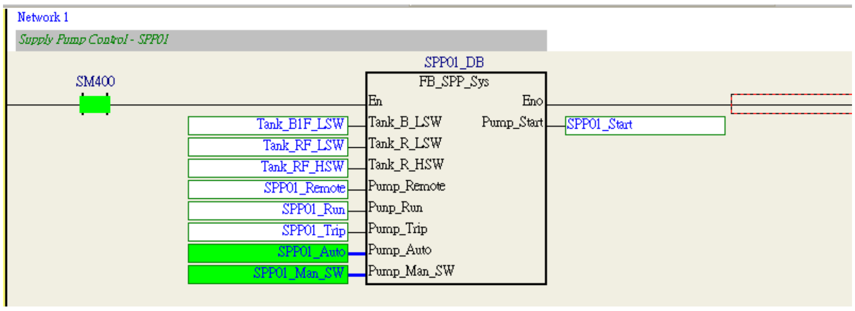
  Open the **Prog_Main** window, and click .



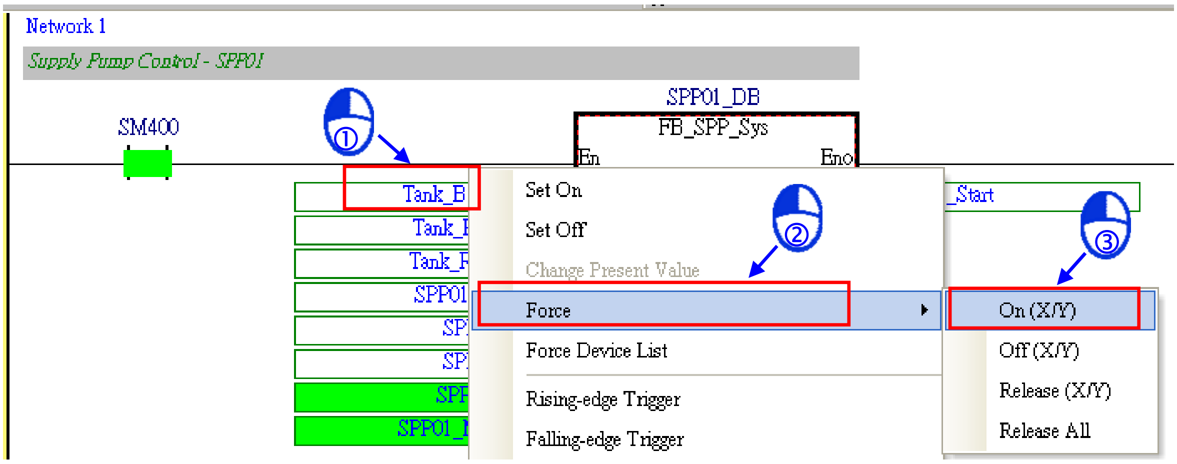
Local Symbols

Declaration Type	Identifiers	Address	Type...	Initial Value	
VAR	SPP01_DB	N/A [Auto]	FB_SPP_Sys	N/A	SPP01 D

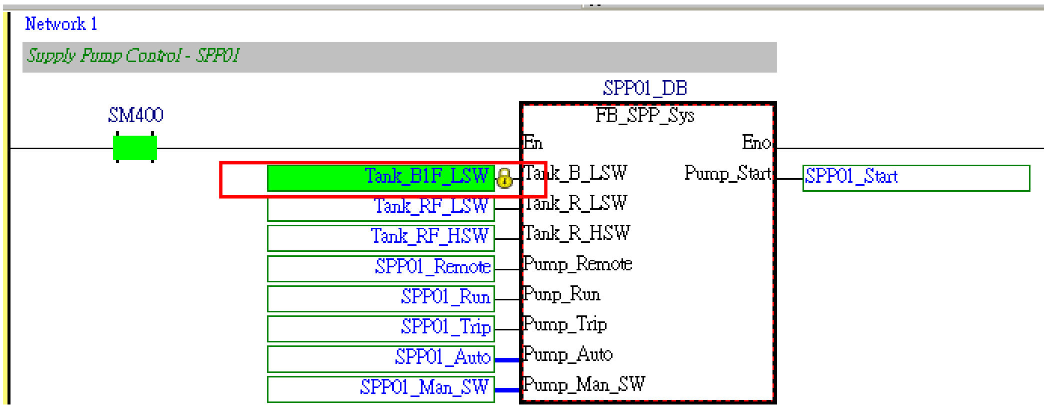
Not only the logic program, but also the information related to the devices is monitored. (If the data type is a Boolean value, the state of the device is represented by green (ON) or white (OFF). If the data type is not a Boolean value, the value in the device is shown.)



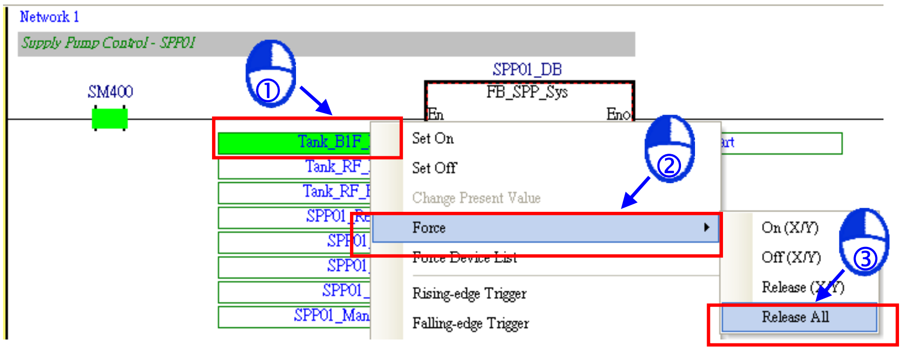
If users want to alter the value in the device, they can click the right mouse button. Tank_B1F_LSW is the device which is modified here. Please notice that the device corresponds to an actual I/O device. Setting the device to ON or OFF is meaningless because the actual I/O value covers the setting value immediately. To alter the value in the device, users need to force the device ON or OFF.



The value in the device is altered, and a lock symbol indicating that the device is forced ON appears.



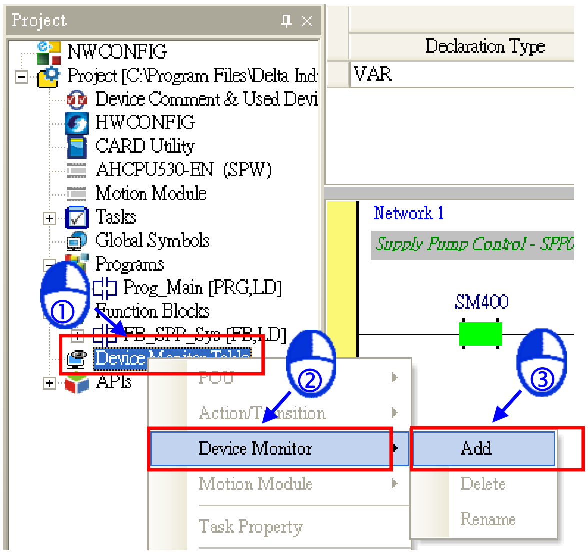
After the system is tested, users have to cancel the setting so that the CPU module can return to normal.



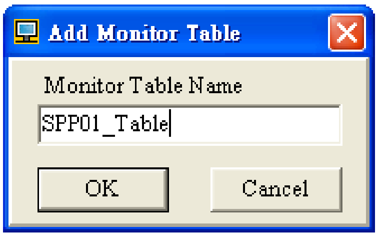
- **Monitoring the device**

  When the logical control is tested, the devices which have to be modified may be used in different programs. It is not convenient to find these devices in the programs and monitor them. Besides, users sometimes do not alter the value in a device for debugging. They alter the value for testing the external device. If users want to alter the value in a device by means of monitoring the program, they will have difficulty in finding the device, and the program has to be uploaded from the CPU module to the computer. In order to solve the difficulty, users can alter the value in a device by means of a device monitoring table. User even can alter the value in a device and monitor the device without the program.

Find Device Monitor Table in the window at the left side of the main screen of the software, and click the right mouse button to add a new device monitoring table.



Type “SPP01_Table” in the **Monitor Table Name** box, and click **OK**.



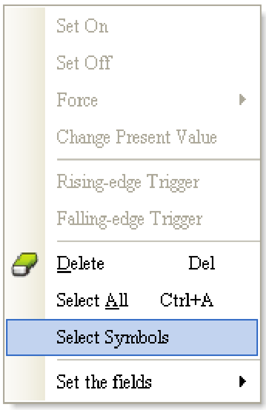
Add Monitor Table

Monitor Table Name

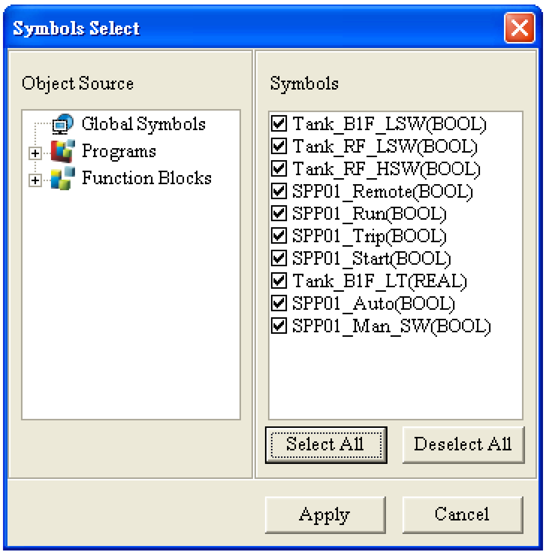
SPP01_Table

OK Cancel

Double-click the blank to open the **Device Monitor Input** window, or click the right mouse button to select the symbols.

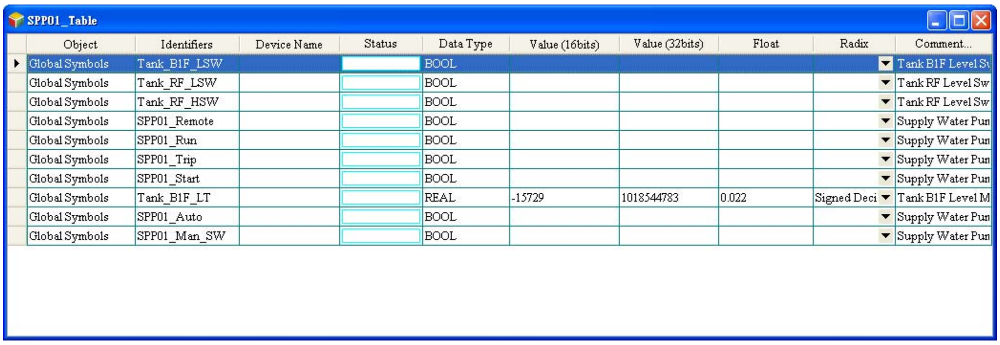


Click **Select All**, and then click **Apply**.



☑ Tank_B1F_LSW(BOOL)
☑ Tank_RF_LSW(BOOL)
☑ Tank_RF_HSW(BOOL)
☑ SPP01_Remote(BOOL)
☑ SPP01_Run(BOOL)
☑ SPP01_Trip(BOOL)
☑ SPP01_Start(BOOL)
☑ Tank_B1F_LT(REAL)
☑ SPP01_Auto(BOOL)
☑ SPP01_Man_SW(BOOL)

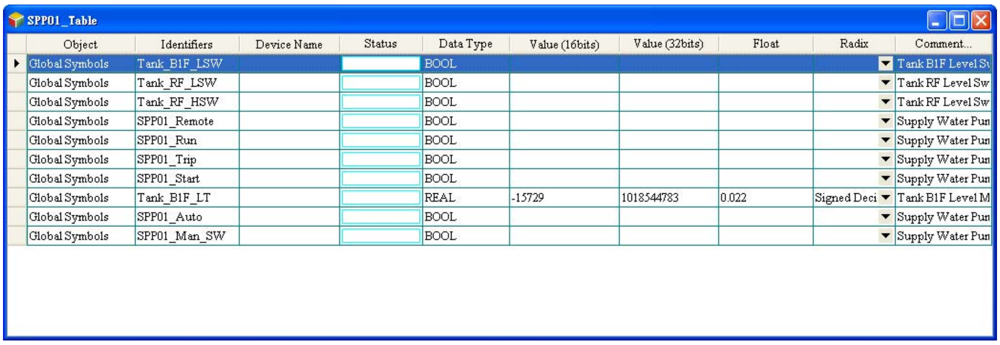
Click **Cancel**, and then save the file. The device monitoring table is as follows.



SPP01_Table

Object	Identifiers	Device Name	Status	Data Type	Value (16bits)	Value (32bits)	Float	Radix	Comment...
Global Symbols	Tank_B1F_LSW			BOOL					Tank B1F Level S
Global Symbols	Tank_RF_LSW			BOOL					Tank RF Level Sw
Global Symbols	Tank_RF_HSW			BOOL					Tank RF Level Sw
Global Symbols	SPP01_Remote			BOOL					Supply Water Pun
Global Symbols	SPP01_Run			BOOL					Supply Water Pun
Global Symbols	SPP01_Trip			BOOL					Supply Water Pun
Global Symbols	SPP01_Start			BOOL					Supply Water Pun
Global Symbols	Tank_B1F_LT			REAL	-15729	1018544783	0.022	Signed Deci	Tank B1F Level M
Global Symbols	SPP01_Auto			BOOL					Supply Water Pun
Global Symbols	SPP01_Man_SW			BOOL					Supply Water Pun

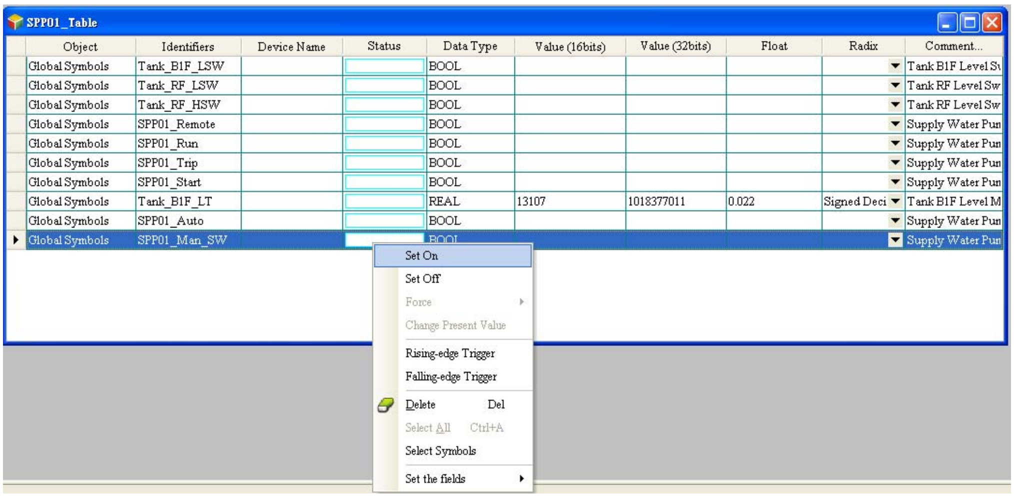
Click . Owing to the fact that only devices are monitored and modified, users do not need a copy of the program.



SPP01_Table

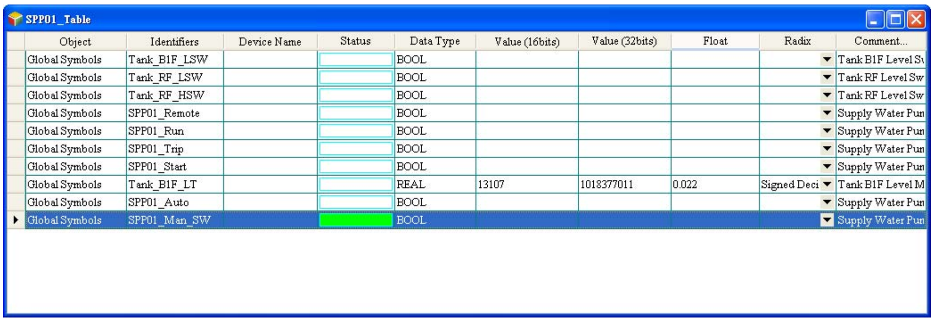
Object	Identifiers	Device Name	Status	Data Type	Value (16bits)	Value (32bits)	Float	Radix	Comment...
Global Symbols	Tank_B1F_LSW			BOOL					Tank B1F Level S
Global Symbols	Tank_RF_LSW			BOOL					Tank RF Level Sw
Global Symbols	Tank_RF_HSW			BOOL					Tank RF Level Sw
Global Symbols	SPP01_Remote			BOOL					Supply Water Pun
Global Symbols	SPP01_Run			BOOL					Supply Water Pun
Global Symbols	SPP01_Trip			BOOL					Supply Water Pun
Global Symbols	SPP01_Start			BOOL					Supply Water Pun
Global Symbols	Tank_B1F_LT			REAL	-15729	1018544783	0.022	Signed Deci	Tank B1F Level M
Global Symbols	SPP01_Auto			BOOL					Supply Water Pun
Global Symbols	SPP01_Man_SW			BOOL					Supply Water Pun

If users want to alter the value in the device, they can click the right mouse button. The state of SPP01_Man_SW is altered here. Owing to the fact that the device does not correspond to an actual I/O device, users can set the device to ON or OFF.


SPP01_Table

	Object	Identifiers	Device Name	Status	Data Type	Value (16bits)	Value (32bits)	Float	Radix	Comment...
	Global Symbols	Tank_B1F_LSW			BOOL					Tank B1F Level S\
	Global Symbols	Tank_RF_LSW			BOOL					Tank RF Level Sw
	Global Symbols	Tank_RF_HSW			BOOL					Tank RF Level Sw
	Global Symbols	SPP01_Remote			BOOL					Supply Water Pun
	Global Symbols	SPP01_Run			BOOL					Supply Water Pun
	Global Symbols	SPP01_Trip			BOOL					Supply Water Pun
	Global Symbols	SPP01_Start			BOOL					Supply Water Pun
	Global Symbols	Tank_B1F_LT			REAL	13107	1018377011	0.022	Signed Deci	Tank B1F Level M
	Global Symbols	SPP01_Auto			BOOL					Supply Water Pun
▸	Global Symbols	SPP01_Man_SW			BOOL					Supply Water Pun

After the state of SPP01_Man_SW is altered, users can get the following device monitoring table.


SPP01_Table

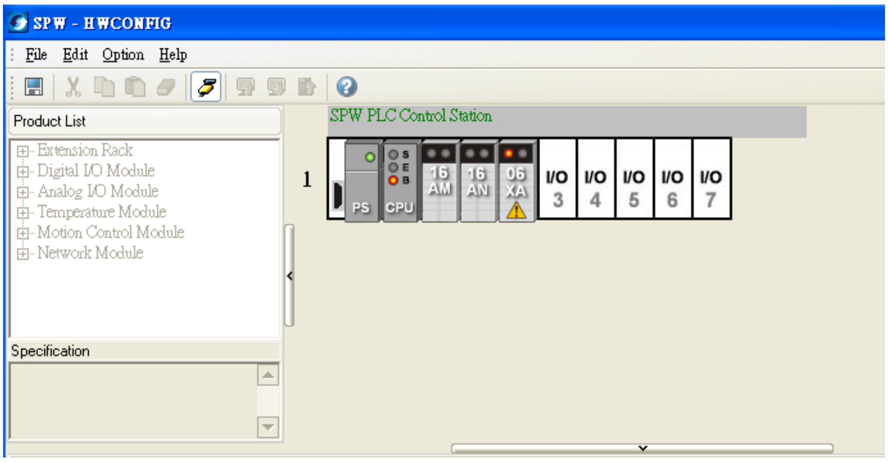
	Object	Identifiers	Device Name	Status	Data Type	Value (16bits)	Value (32bits)	Float	Radix	Comment...
	Global Symbols	Tank_B1F_LSW			BOOL					Tank B1F Level S\
	Global Symbols	Tank_RF_LSW			BOOL					Tank RF Level Sw
	Global Symbols	Tank_RF_HSW			BOOL					Tank RF Level Sw
	Global Symbols	SPP01_Remote			BOOL					Supply Water Pun
	Global Symbols	SPP01_Run			BOOL					Supply Water Pun
	Global Symbols	SPP01_Trip			BOOL					Supply Water Pun
	Global Symbols	SPP01_Start			BOOL					Supply Water Pun
	Global Symbols	Tank_B1F_LT			REAL	13107	1018377011	0.022	Signed Deci	Tank B1F Level M
	Global Symbols	SPP01_Auto			BOOL					Supply Water Pun
▸	Global Symbols	SPP01_Man_SW		🟩	BOOL					Supply Water Pun

If users do not want to monitor the devices, they can click ⚡ and close the device monitoring table.

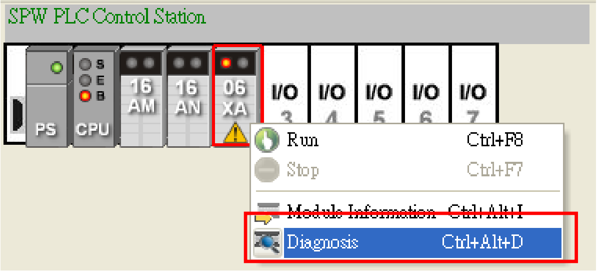
3.3.2 Debugging the Program and the System

When the system operates, an error may occur, and the error LED indicator on a module may be ON. If an error occurs, and the error LED indicator is not ON, the error is a logic program error. If users follow the steps in the example, no error will occur. In order to demonstrate a system error, users are asked to turn off the 24 V DC power supply. After the 24 V DC power supply is turned off, the bus fault LED indicator on the CPU module and the error LED indicator on AH06XA-5A are ON.

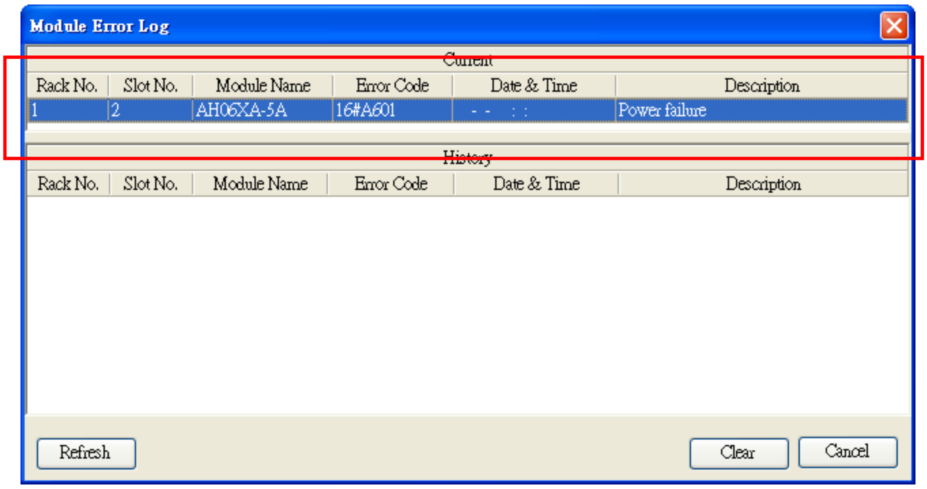
Find HWCONFIG in the window at the left side of the main screen of the software, and click . The actual statuses of the modules are as follows.



Generally speaking, a hardware configuration error occurs if the bus fault LED indicator on the CPU module is ON. Users can select AH06XA-5A, click the right mouse button, and select **Diagnosis**.



The information about the error is shown in the following picture.



Module Error Log

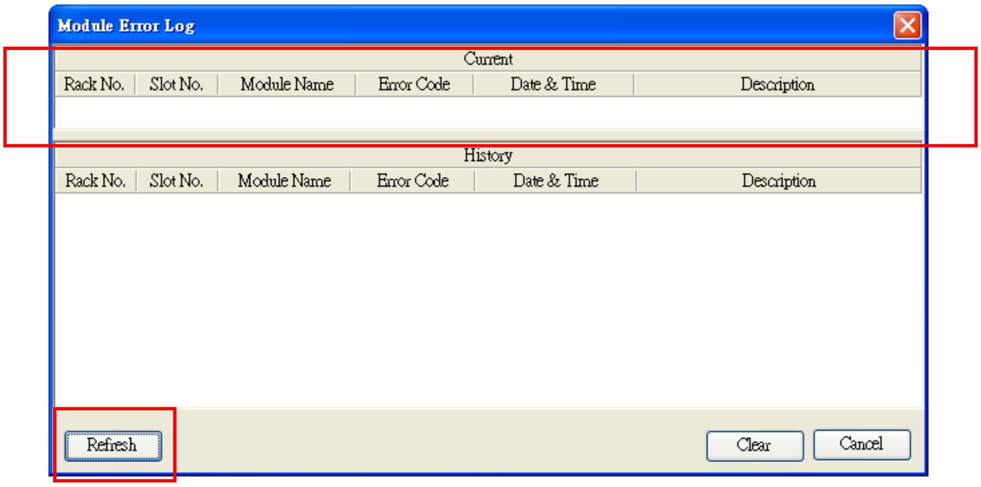
Current

Rack No.	Slot No.	Module Name	Error Code	Date & Time	Description
1	2	AH06XA-5A	16#A601	- - : :	Power failure

History

Rack No.	Slot No.	Module Name	Error Code	Date & Time	Description

As the picture above shows, the module is not supplied with power. Please turn on the power supply, and click **Refresh**. The system will check the status of the module again. In this example, the module has been supplied with power. Therefore, the system does not detect any error, and there is no error log in the **Current** box, as shown below. If the system detects a new error, the new error log appears in the **Current** box. Besides, the error log in the **History** box is cleared if users click **Clear**. However, due to the fact that the modules are designed differently, the error logs are not maintained in all modules. In other words, for some modules, there may be no error log in the **History** box.



Module Error Log

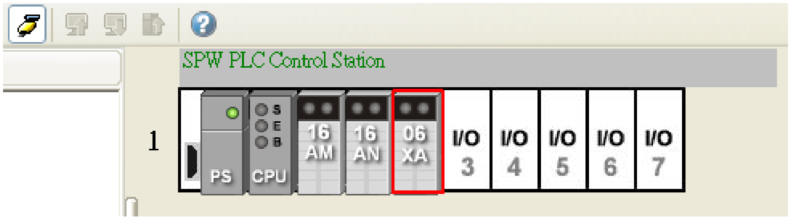
Current

Rack No.	Slot No.	Module Name	Error Code	Date & Time	Description

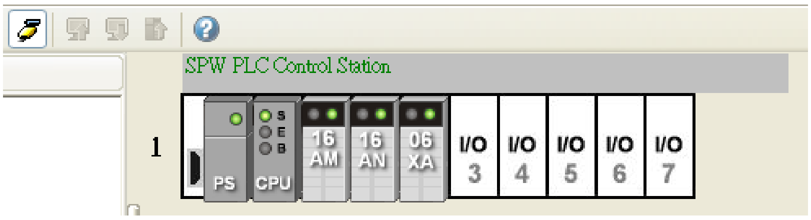
History

Rack No.	Slot No.	Module Name	Error Code	Date & Time	Description

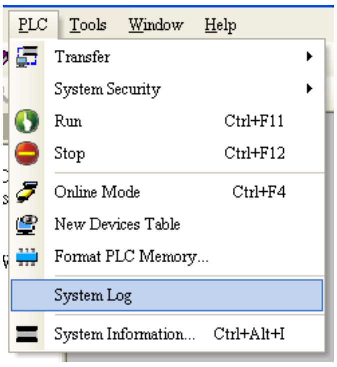
Since no error occurs, users can close the **Module Error Log** window. Besides, users can find that the whole system returns to normal.



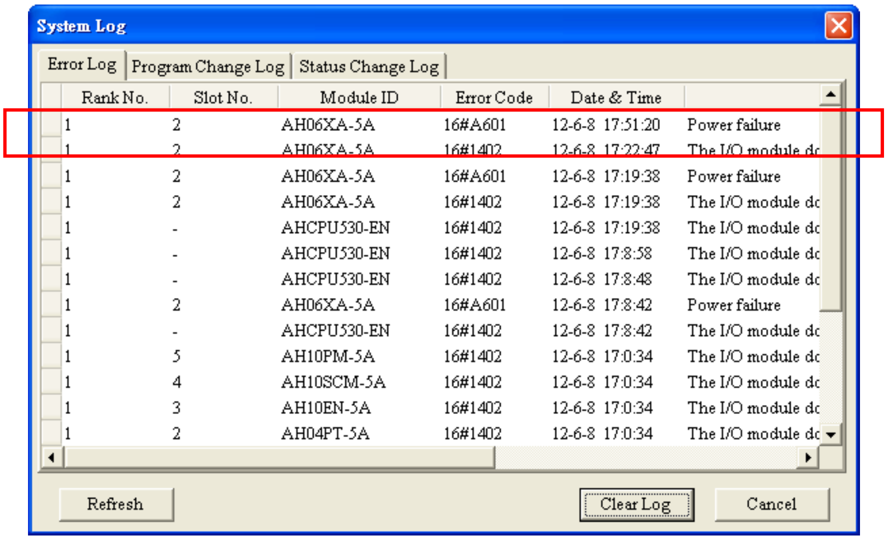
Users can set the CPU module to STOP, and then set it to RUN. When the CPU module runs, the system and the screen of the software return to normal.



Close the **SPW-HWCONFIG** window, and return to the main screen of ISPSoft. Users can see the error log by selecting **PLC Error Log** on the **PLC** menu.



As the picture below shows, the power supply is abnormal, and an error occurs when the analog data is converted into the digital data.



Rank No.	Slot No.	Module ID	Error Code	Date & Time	
1	2	AH06XA-5A	16#A601	12-6-8 17:51:20	Power failure
1	2	AH06XA-5A	16#1402	12-6-8 17:22:47	The I/O module dc
1	2	AH06XA-5A	16#A601	12-6-8 17:19:38	Power failure
1	2	AH06XA-5A	16#1402	12-6-8 17:19:38	The I/O module dc
1	-	AHCPU530-EN	16#1402	12-6-8 17:19:38	The I/O module dc
1	-	AHCPU530-EN	16#1402	12-6-8 17:8:58	The I/O module dc
1	-	AHCPU530-EN	16#1402	12-6-8 17:8:48	The I/O module dc
1	2	AH06XA-5A	16#A601	12-6-8 17:8:42	Power failure
1	-	AHCPU530-EN	16#1402	12-6-8 17:8:42	The I/O module dc
1	5	AH10PM-5A	16#1402	12-6-8 17:0:34	The I/O module dc
1	4	AH10SCM-5A	16#1402	12-6-8 17:0:34	The I/O module dc
1	3	AH10EN-5A	16#1402	12-6-8 17:0:34	The I/O module dc
1	2	AH04PT-5A	16#1402	12-6-8 17:0:34	The I/O module dc

If users want to delete the old error log, they can click **Clear Log**. After closing the software, users complete the operation illustrated in this example.

MEMO