

innOCTRL



INN-A3-MODS04COM serial communication module

2026-05-06 | V1.0.2

Contents

1. Document Description 1.....	1
1.1 Compliance with standards 1.....	1
1.2 Version Revision History 1.....	1
1.3 Warranty Description 1.....	1
1.4 Safety Precautions 2.....	2
1.4.1 Safety Instructions 2.....	2
1.4.2 Security Level 2.....	3
2. Product Information 5.....	8
2.1 Naming rules and nameplate description 5.....	8
2.2 Component Description 7.....	9
2.3 Technical Specifications 8.....	10
2.3.1 Power supply specifications 8.....	10
2.3.2 Serial port specifications 9.....	11
2.4 Interface Definition and terminal wiring 11.....	13
2.4.1 Interface Definition 11.....	13
2.4.2 Terminal Wiring 13.....	15
2.5 Environmental scope13.....	15
3. Dimensions and installation 14.....	16
3.1 Module Dimensions 14.....	16
3.2 Equipment installation 14.....	16
3.2.1 Installation 14.....	16
3.2.2 Disassembly 16.....	18
3.3 Electrical Installation 17.....	19
4. Programming Example 18	21
4.1 Use with EtherCAT Coupler18.....	21
4.2 Use with PROFINET Coupler 27.....	32

1 Document Description

INN-A3-MODS04COM 4-channel serial communication module, this product has 2 RS232 and 2 RS485/RS422 channels, and is suitable for INNOCTRL A-class series Coupler modules.

1.1 Applicable Standards

Table 1 Certification standards1

Certification name	Command name		Applicable Standards
CE certification	EMC directive	2014/30/EU	• EN 61131-2
	LVD instruction	2014/35/EU	• EN 61010-1 • EN 61010-2-201

1.2 Version Revision History

Table 2 version revision record2

Revision date	release version	Change history
2025/09/14	V1.0.0	initial version
2025/11/04	V1.0.1	Updated programming example parameter descriptions
2026/05/06	V1.0.2	Added programming example for use with PROFINET Coupler

1.3 Warranty Description

Under normal use, if the product fails or is damaged, Zhejiang Visual Intelligent Technology Co., Ltd. provides warranty services

within the warranty period (please refer to the order form for details of the product warranty period). After the warranty period, repair fees will be charged.

During the warranty period, if the product is damaged due to the following conditions, repair fees will be charged.

- 1) Failure to operate this product according to the regulations in this manual may result in damage to the product.
- 2) Product damage caused by fire, flood, voltage abnormality.
- 3) Product damage caused by using this product for abnormal functions.
- 4) Product damage caused by exceeding the specified use range of the product.
- 5) Secondary damage to the product caused by force majeure (natural disasters, earthquakes, lightning strikes).

The relevant service fees are calculated according to the manufacturer's unified standards. If there is a contract, the contract priority will be used.

Please refer to the "Product Warranty Card" for detailed warranty instructions.

1.4 Safety Precautions

1.4.1 Safety Instructions

When installing, operating, and maintaining the product, please read and comply with the following safety precautions:

- To ensure personal and equipment safety, please follow all safety precautions marked on the product and described in this manual when installing, operating, and maintaining the product.
- The "Caution", "Warning" and "Danger" items in this manual do not represent all the safety precautions that should be observed, but are only supplementary to all safety precautions.
- This product should be used in an environment that meets the design specifications, otherwise it may cause malfunctions.

Functional abnormalities or component damage caused by failure to comply with relevant regulations are not covered by the product quality guarantee.

- Our company will not bear any legal responsibility for personal safety accidents, property losses, etc. caused by illegal operation of products.

1.4.2 Security level

This manual will indicate the user's safety level into three categories: danger, warning, and caution.

DANGER

Indicates that death or serious bodily injury may result if instructions are not followed.

WARNING

Indicates that death or serious bodily injury may result if instructions are not followed.

CAUTION

Indicates that minor bodily injury or equipment damage may result if operations are not carried out as specified.

Please keep this manual in a safe place for reading when necessary, and be sure to give this manual to the end user.

1.4.2.1 When designing the control system

When designing a control system, please read and follow the following safety precautions:

DANGER

- Please be sure to design a safety circuit to ensure that the control system can still work safely when the external power supply fails or the programmable controller fails;
- When the rated load current is exceeded or the load is short-circuited, causing overcurrent for a long time, the module may smoke or catch fire. Safety devices such as fuses or circuit breakers should be installed externally.

 WARNING

- Please be sure to set up emergency braking circuits, protection circuits, interlock circuits for forward and reverse operations, and position upper and lower limit interlock switches to prevent machine damage in the external circuit of the programmable controller;
- In order to ensure the safe operation of the equipment, please design external protection circuits and safety mechanisms for output signals related to major accidents; the programmable controller CPU may shut down all outputs after detecting an abnormality in its own system; when part of the controller's circuit fails, its output may be uncontrolled. To ensure normal operation, it is necessary to design an appropriate external control circuit;
- When the output units such as relays and transistors of the programmable controller are damaged, the output cannot be controlled to ON or OFF state;
- Programmable controllers are designed to be used in indoor, overvoltage level II electrical environments. The power supply system level should have lightning protection devices to ensure that lightning overvoltage is not applied to the power input end or signal input end, control output end and other ports of the programmable controller to avoid damage to the equipment.

1.4.2.2 When installing

When installing the device, please read and follow the following safety precautions:

 WARNING

- Only professional maintenance personnel who have received relevant training on electrical equipment and have sufficient electrical knowledge can install this product;
- When disassembling or installing a module, all external power supplies used by the system must be disconnected before proceeding. If the power is not completely disconnected, it may cause electric shock or module failure or malfunction;
- Do not use the programmable controller in the following places: places with dust, oil fumes, conductive dust, corrosive gases, and flammable gases; places exposed to high temperatures, condensation, wind and rain; places with vibrations and shocks. Electric shock, fire, and misoperation can also cause product damage and deterioration;
- The programmable controller is an Open type device. Please install it in a control cabinet with a door lock (control cabinet shell protection > IP20). Only operators who have received relevant training on electrical equipment and have sufficient electrical knowledge can open the control cabinet.

 CAUTION

- When installing, avoid metal shavings and wire ends from falling into the ventilation holes of the controller, which may cause fire, malfunction, and misoperation;
- After installation, make sure there are no foreign objects on the ventilation surface, otherwise it may cause poor heat dissipation, causing fire, malfunction, and misoperation;
- When installing, they should be tightly connected to their respective connectors and the module connection hooks should be locked firmly. If the module is installed improperly, it may cause malfunction, malfunction, and falling off.

1.4.2.3 When wiring

When wiring, please read and follow the following safety precautions:

DANGER

- Only professional maintenance personnel who have received relevant training on electrical equipment and have sufficient electrical knowledge can perform wiring on this product;
- During wiring work, all external power supplies used by the system must be disconnected before proceeding. If all are not disconnected, it may cause electric shock or equipment failure or malfunction;
- Cable terminals should be well insulated to ensure that the insulation distance between cables will not be reduced after the cables are installed on the terminal block. Otherwise it may cause electric shock or equipment damage.

CAUTION

- To avoid electric shock, please cut off the power supply before connecting this product;
- Please refer to the data in the technical specifications for the input power supply specifications of this product. Please provide the power supply strictly in accordance with the data in the technical specifications. If the power supplied is not within the required range, the product will be seriously damaged. Therefore, please regularly check whether the DC power provided by the switching power supply is stable.

1.4.2.4 During operation and maintenance

When operating and maintaining the equipment, please read and comply with the following safety precautions:

⚠ CAUTION

- Only professional maintenance personnel who have received relevant training on electrical equipment and have sufficient electrical knowledge can operate and maintain the product;
- Do not touch the terminals when the power is on, otherwise it may cause electric shock or malfunction;
- When disassembling modules or connecting or removing communication cables, all external power supplies used by the system must be disconnected first. Failure to disconnect them all may result in electric shock or malfunction.

1.4.2.5 When scrapped

When scrapping your equipment, please read and follow the following safety precautions:

⚠ CAUTION

- Please treat it as industrial waste; when discarding batteries, they should be disposed of separately in accordance with the laws and regulations established by each region;
- Scrapped equipment and products should be processed and recycled according to industrial waste treatment standards to avoid environmental pollution.

2 product information

2.1 Naming rules and nameplate instructions

INN-A 3-MODS 04 COM

① ② ③ ④ ⑤ ⑥

Table 3 Naming rules table3

No.	Description
①	Product range INNOCTRL
②	Module classification level: A-series
③	Product generation: 3, third generation product
④	Module classification: • MODD, digital module • CPL, Coupler • MODA, analog module • MODS, serial port module • ASSY, attachment
⑤	Number of channels: 04, 4 channels
⑥	COM: serial communication module

2.2 Component Description

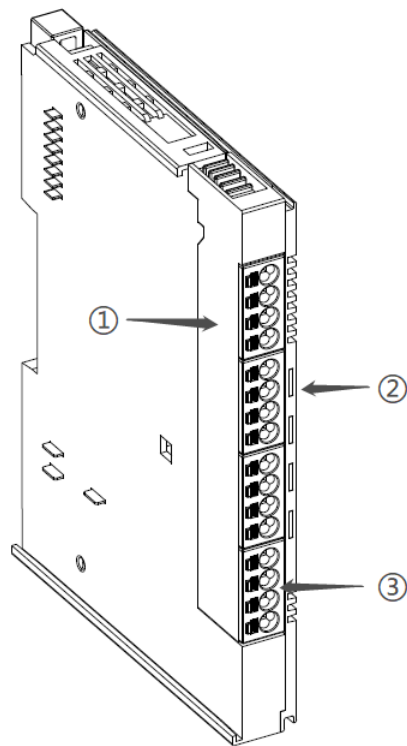


Figure 1 Component diagram1

As shown in the figure above, the component mainly consists of three parts. See the table below for detailed description:

Table 4 Component description table4

No.	Description
①	Plugable terminal box, including 2 RS232 and 2 RS485/RS422 channels
②	Module indicator light
③	Replaceable integrated light guide terminal

2.3 Technical Specifications

2.3.1 Power specifications

Table 5 Power supply specification table5

Item		Specification
bus powered	Voltage	Typical value 5V DC
	Current	Typical value 200mA
On-site power supply	Voltage	24V DC
	Current	Typical value 50mA+load

2.3.2 Serial port specifications

2.3.2.1 RS232 specifications

The specification table of 2-way RS232 serial port is as follows:

Table 6 RS232 specification table6

Item	Specification
Communication interface	RS232
Number of channels	2 way
baud rate	9600bps, 14400bps, 19200bps, 38400bps, 57600bps and 115200bps
data bits	7 or 8 bits, the default is 8 bits (the 7-bit data bits cannot be without parity)
Stop bit	1 digit, 1.5 digit or 2 digit, default is 1 digit
Check digit	Odd parity, even parity, or no parity, the default is no parity
Interface protection	RS232 interface has 600W lightning surge protection and ± 15 KV electrostatic protection per line
Isolate or not	Yes

2.3.2.2 RS485/RS422 specifications

The specification table of 2-way RS485/RS422 serial port is as follows:

Table 7 RS485/RS422 specification table7

Item	Specification
------	---------------

Communication interface	RS485/RS422
Number of channels	• SLOT3 can be equipped with one RS485 or RS422 • SLOT4 can be equipped with one RS485 or RS422
baud rate	9600bps, 14400bps, 19200bps, 38400bps, 57600bps and 115200bps
data bits	7 or 8 bits, the default is 8 bits (the 7-bit data bits cannot be without parity)
Stop bit	1 digit, 1.5 digit or 2 digit, default is 1 digit
Check digit	Odd parity, even parity, or no parity, the default is no parity
Interface protection	RS485/RS422 interface has 600W lightning surge protection and $\pm 15\text{KV}$ electrostatic protection per line
Isolate or not	Yes

2.4 Interface Definition and terminal wiring

2.4.1 Interface Definition

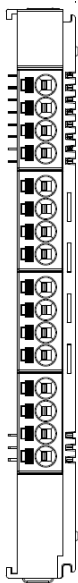


Figure 2 Interface diagram2

The interface is shown in the figure above, and the interface slots and definitions are shown in the table below:

Table 8 Interface Definition table8

Serial port type	slot	No.	definition
RS232	SLOT1	1	TX
		2	RX
		3	GND
		4	—
	SLOT2	1	TX
		2	RX

RS458/RS422		3	GND
		4	–
	SLOT3	1	● RS458: B ● RS422: Z
		2	● RS458: A ● RS422: Y
		3	● RS458: – ● RS422: B
		4	● RS458: – ● RS422: A
	SLOT4	1	● RS458: B ● RS422: Z
		2	● RS458: A ● RS422: Y
		3	● RS458: – ● RS422: B
		4	● RS458: – ● RS422: A

2.4.2 Terminal Wiring



Figure 3 terminal wiring diagram3

2.5 Environmental scope

Table 9 Environmental requirements table9

Item	Specification
Working environment temperature	-20° C~55° C
Working environment humidity	10%~90%RH, no condensation
Usage environment	No corrosive, flammable gas, conductive dust (dust), not serious occasions
Ambient storage temperature	-40° C~70° C

altitude	≤2000m
Immunity	Power cord 2kV (IEC 61000-4-4)
Overvoltage category	I
EMC anti-interference level	Zone B, IEC61131-2

3 Dimensions and installation

3.1 Module Dimensions

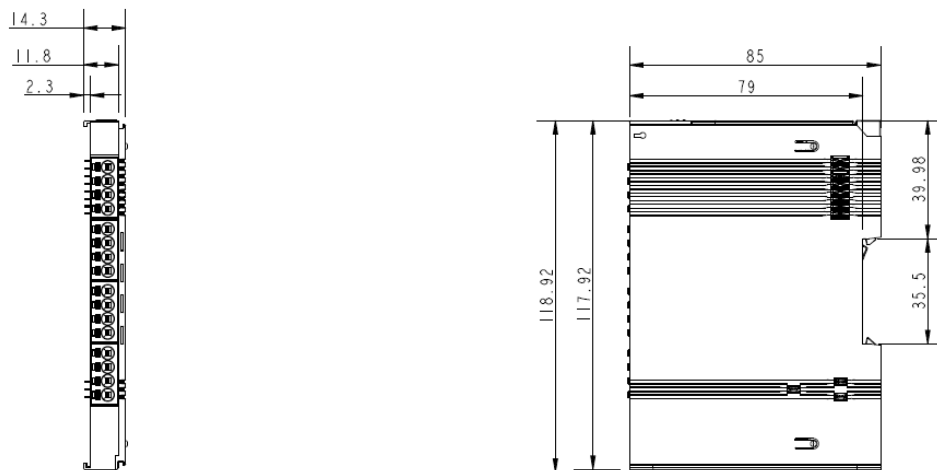


Figure 4 Module Dimensions diagram4

3.2 Equipment installation

3.2.1 Installation

The module is installed on a DIN rail. The DIN rail must comply with the IEC 60715 standard (35mm wide, 1mm thick). The size information is as shown in the figure below, in millimeters (mm):

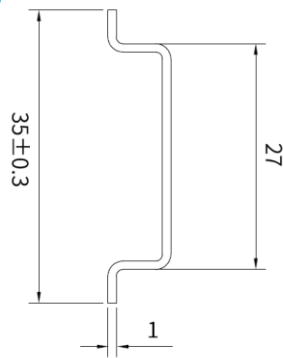


Figure 5 DIN rail diagram5

⚠ CAUTION

If the module is installed on a DIN rail other than those recommended above, the DIN rail buckle may not lock properly.

Module-to-module assembly is performed by sliding the top and bottom rails of the module as shown in the figure below.

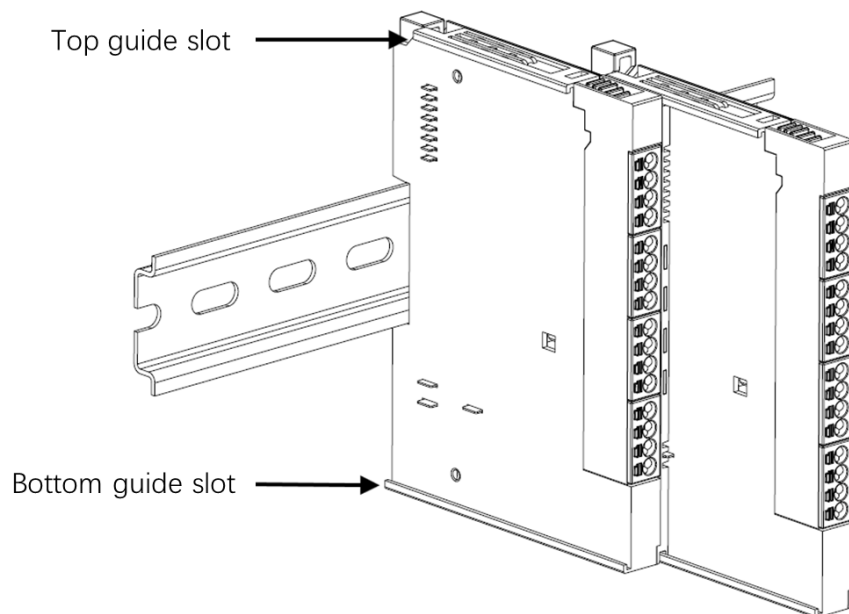


Figure 6 Schematic diagram of module guide slot6

As shown in the figure below, when installing, align the module with the DIN rail and press the module in the direction shown by arrow ① so that the module snaps onto the DIN rail. After the installation is in

place, press the protruding guide rail part at the top in the direction shown by arrow ②. When you hear a snapping sound, the installation is complete.

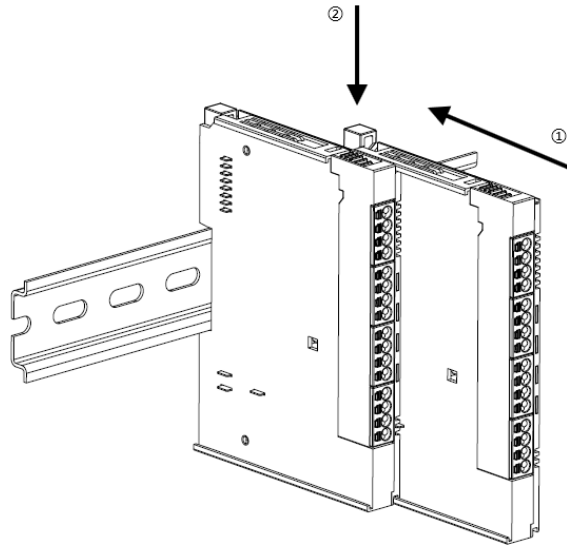


Figure 7 Module installation diagram7

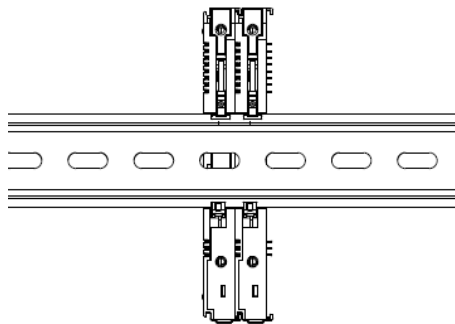


Figure 8 Schematic diagram of DIN rail buckle8

3. 2. 2 Disassembly

As shown in the figure below, use a flat-blade screwdriver or similar tool to pry up the rail lock, and pull the module straight forward at the buckle position (the raised part). When completed, press down on the top of the lock.

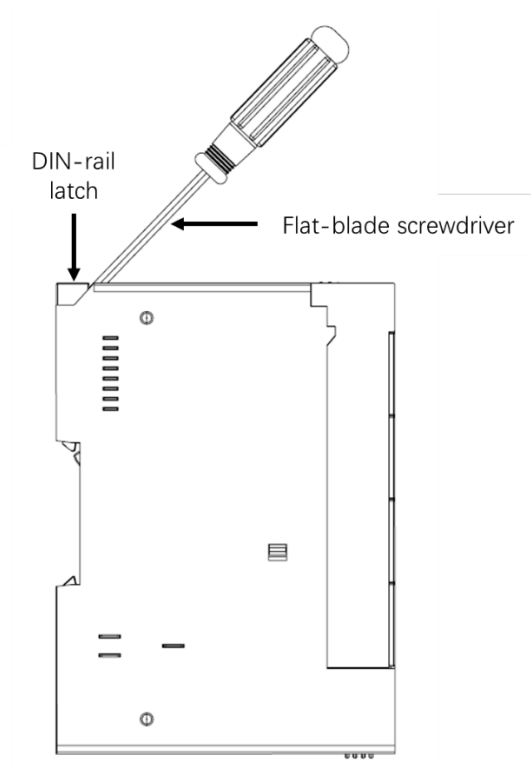


Figure 9 Schematic diagram of prying the guide rail lock

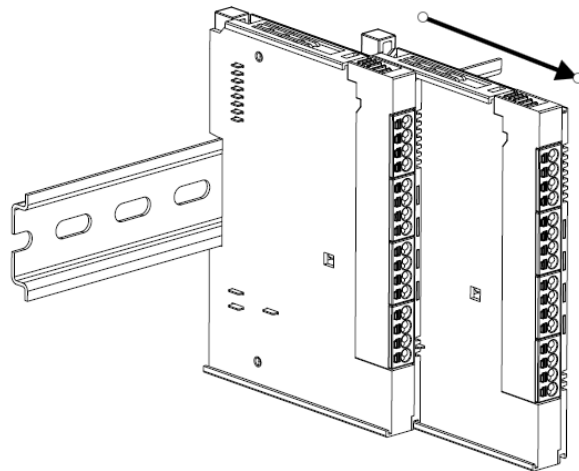


Figure 10 Schematic diagram of pulling out module

3.3 Electrical Installation

The following table shows the selection and installation specifications of cables and tube terminals:

Table 10 Cable and tube terminal specifications10

Supporting material name	Adaptable wire diameter		Tube type terminal
	mm ²	AWG	Specification
Tubular wire lug	0.75	18	E7508
	1.0	17	E1008
	1.5	15	E1508

4 Programming Examples

4.1 For use with EtherCAT Coupler

This section provides a CODESYS programming example for using the INN-A3-MODS04COM 4-channel serial communication module with the INN-A3-CPL-ETC32 EtherCAT Coupler. The steps are as follows:

- 1) In the CODESYS menu bar, click "Tools" and select "Device Repository".

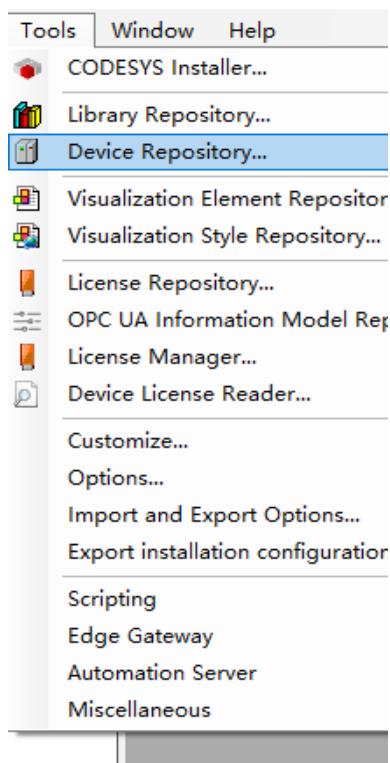


Figure 11 Tool menu drop-down box11

- 2) In the "Device Repository" window interface, click "Install", select the provided .xml format file and open it. After receiving the prompt "The relevant device has been installed in the device repository", click "Close".

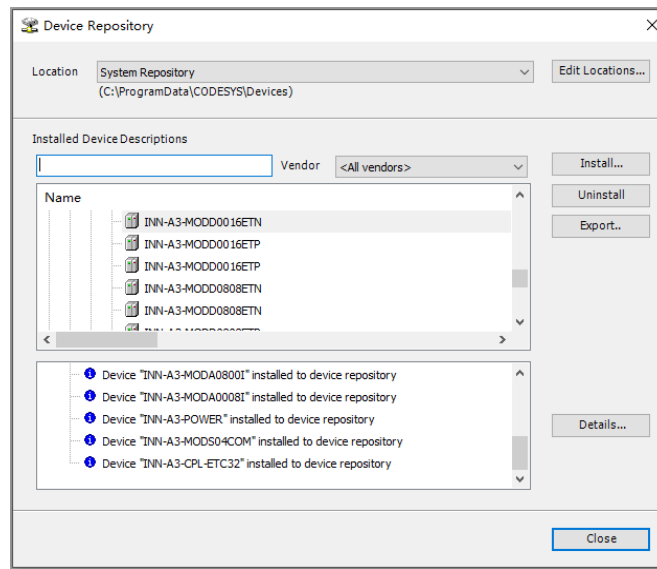


Figure 12 Installing the device into the device repository12

- 3) In the "Device" window on the left side of CODESYS, right-click the name of the PLC master station ("CODESYS_Control_Win_v3_x64" in the example), and then click "Add Device". In the Add Device window, click "EtherCAT Master" and "Add Device" in sequence to successfully add the device. When finished, just close the window.

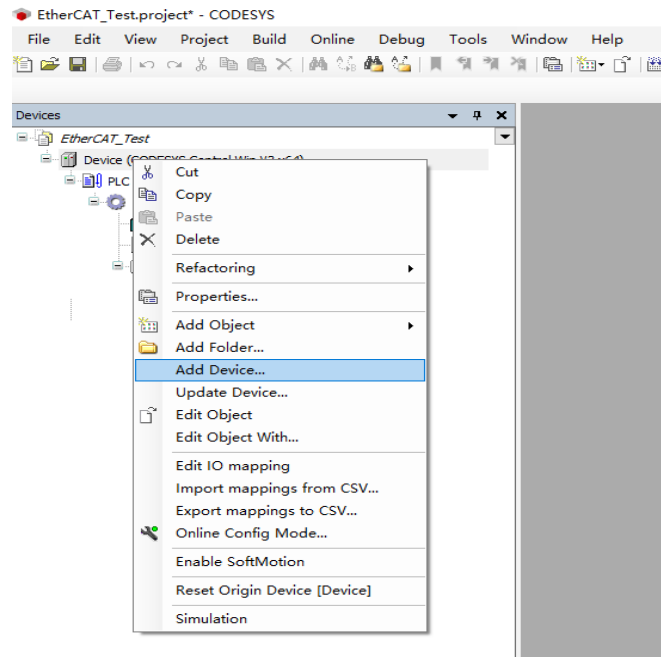


Figure 13 Device window13

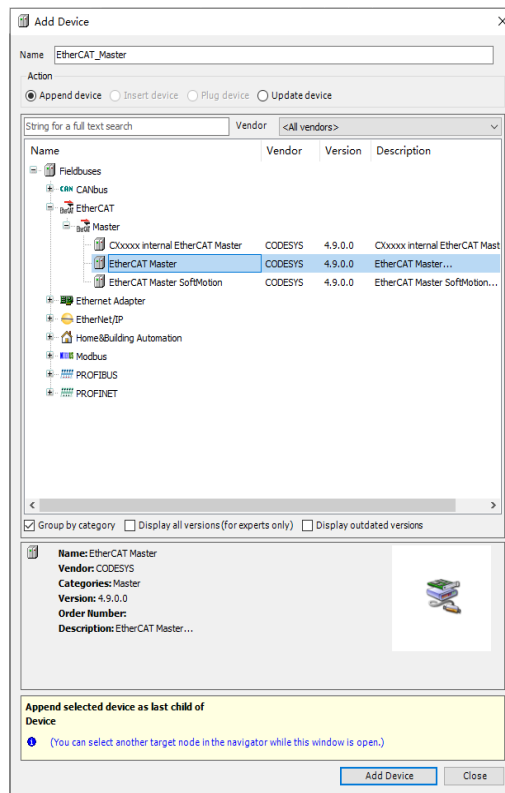


Figure 14 Add device14

- 4) Add corresponding Couplers and I/O modules based on existing modules.

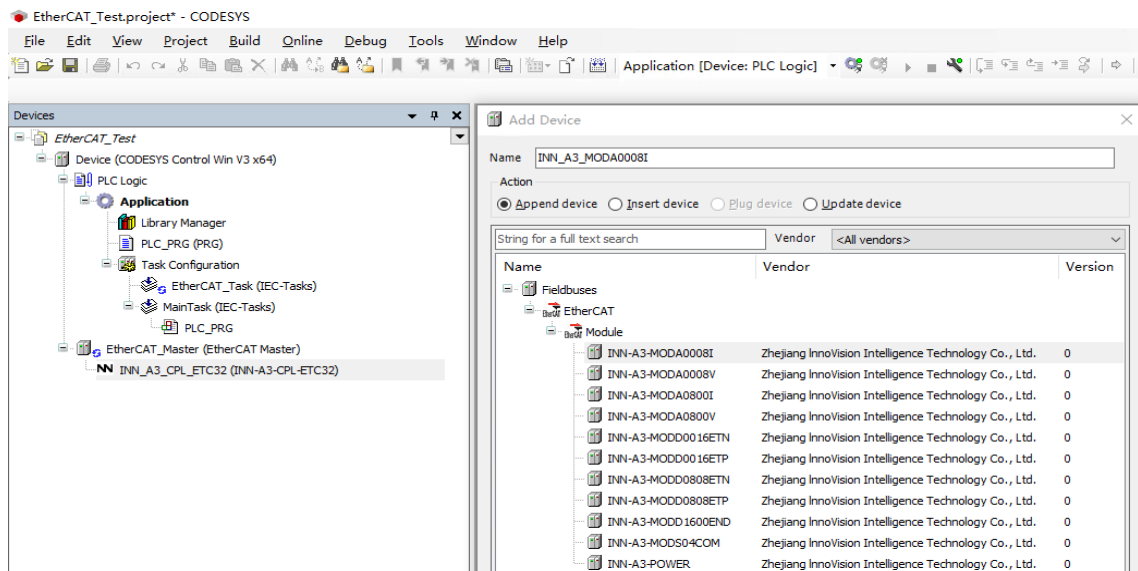


Figure 15 Add Coupler and I/O module15

5) Write code and create variables.

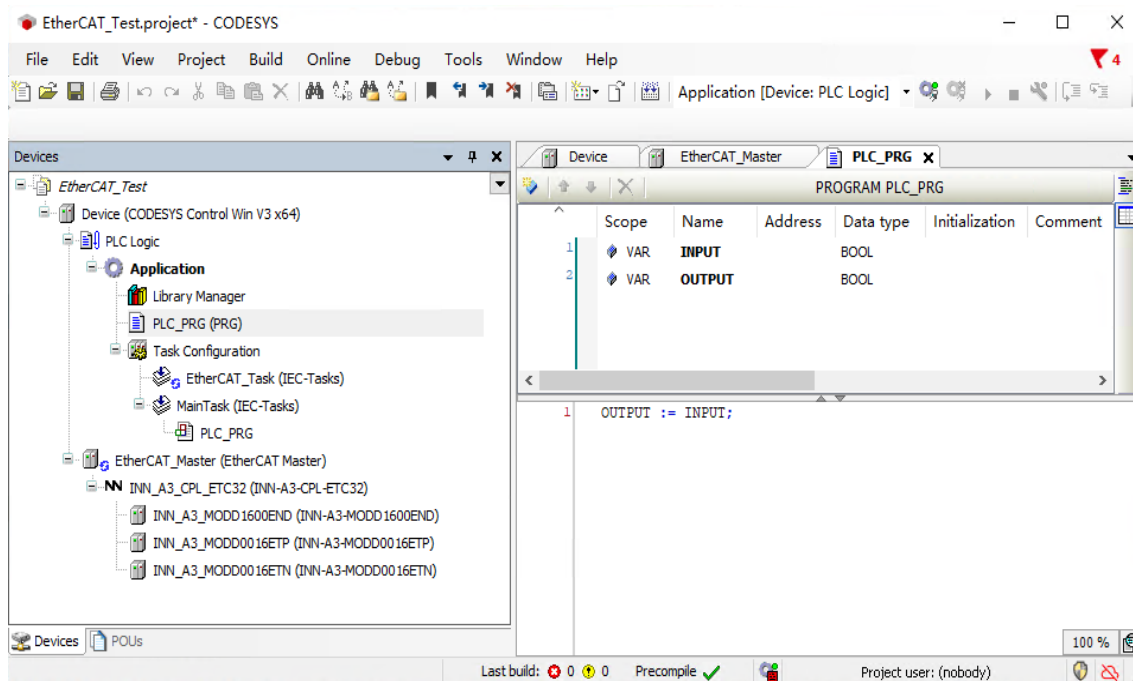


Figure 16 Code window16

6) To associate variables with I/O modules, users need to fill in each byte in the parameter in order according to little-endian byte order.

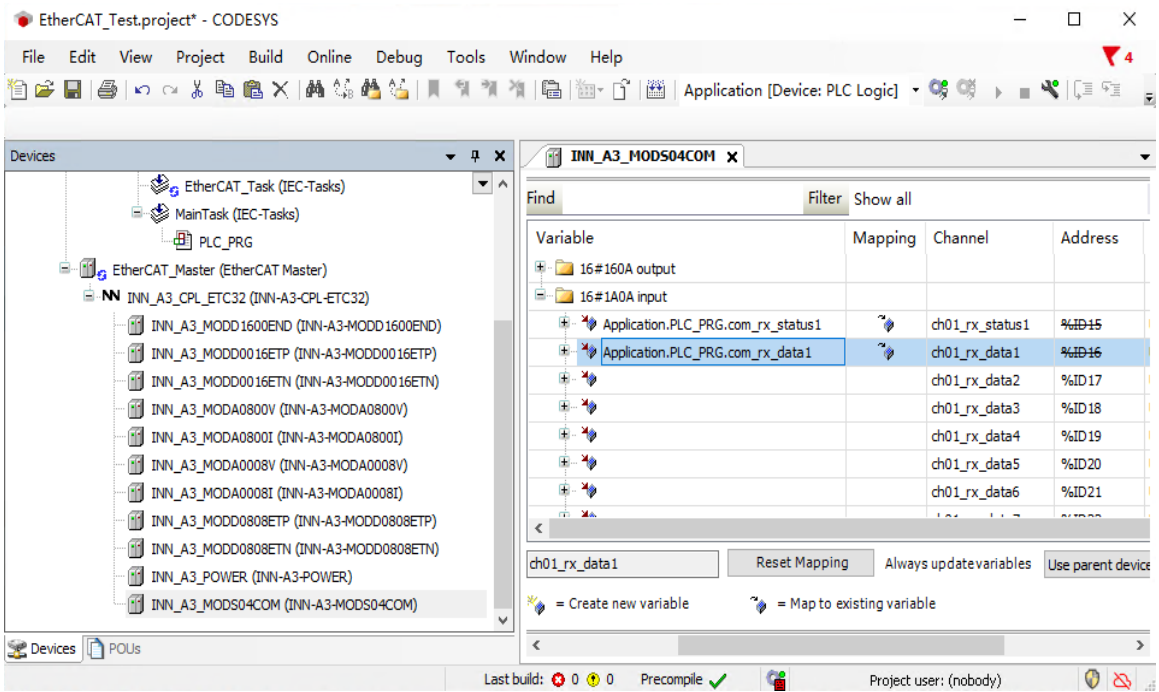


Figure 17 Set input parameters17

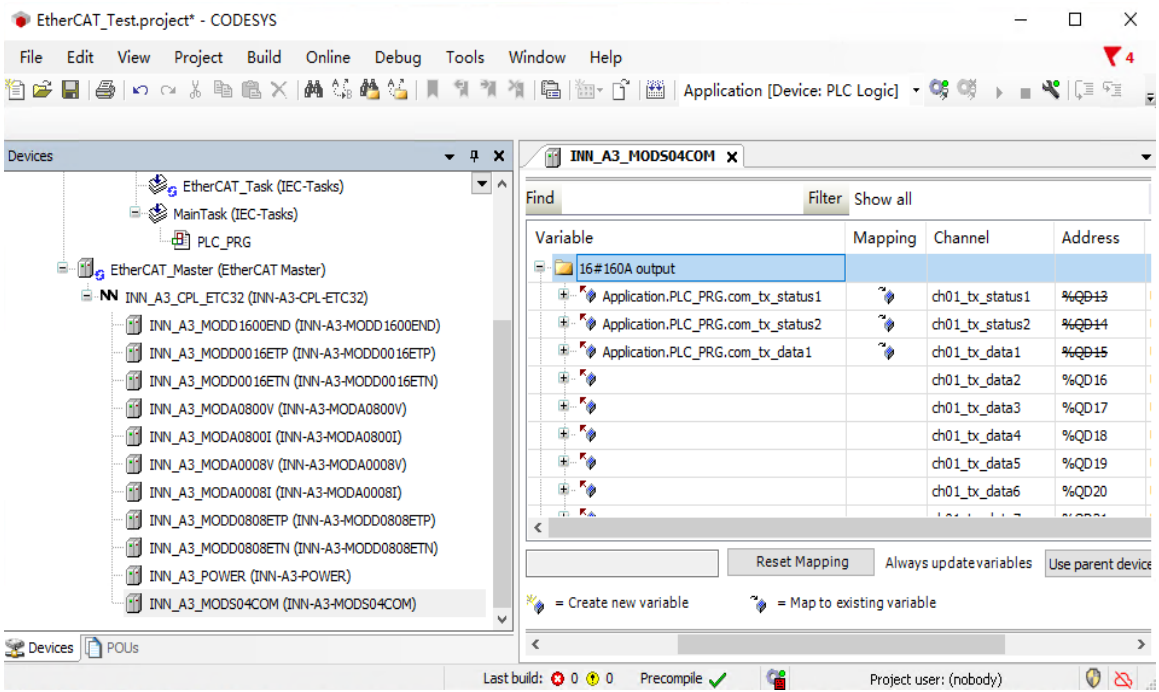


Figure 18 Set output parameters18

Please see the following table for detailed description of input

and output parameters:

 CAUTION

In the output parameters, you need to fill in the ch0x_tx_status1 and ch0x_tx_datax parameters first, and then fill in the ch0x_tx_status2 parameter, otherwise the output data will be wrong.

Table 11 Input and output parameter table11

parameters		Byte content description	
Output parameters	ch0x_tx_status1 Arranged in 4-byte little-endian order	Control byte 1: 1 byte, 485/422 (1b) Reserved (2b) Data bits (1b) Reserved (2b) Stop bits (2b)	485/422 (1b):
			0: 485 1: 422 Only channels 3 and 4 are valid
			Reserve (2b): reserve
			Data bits (1b): 0: 8b 1: 7b (only available when the parity bit is odd parity or even parity 7b)
			Reserve (2b): reserve
			Stop bit (2b): 0: 1b 1: 1.5b

			2: 2b
		Control byte 2: 1 byte, baud rate (4b) reserved (2b) parity digit (2b)	Baud rate (4b): 0: 9600 1: 14400 2: 19200 3: 38400 4: 57600 5: 115200
			Reserved (2b)
			Check digit (2b): 0: No verification (none) 1: Even parity (even) 2: Odd parity (odd)
		Reserved byte: 1 byte, not used yet	–
		plc rx rolling request number: 1 byte, from 0x00 to 0xff, +1 to 0xff, and then return to 0x00 to accumulate in a loop	–
	ch0x_tx_status2 Arranged in 4-byte little-endian order	plc tx rolling request number: 1 byte, from 0x00 + 1 to 0xff, and then return to 0x00 to accumulate in a loop	–

		Total number of frames: 1 byte, indicating the total number of frames that need to be split for a complete data	-
		Current frame number: 1 byte, indicating the current frame number where the current data is located in the total number of frames split by a complete data	-
		Valid data length: 1 byte, indicating the effective data length in the current ch0x_tx_datax field, the maximum length is 32 bytes	-
	ch0x_tx_data1/ ch0x_tx_data2/ ch0x_tx_data3/ ch0x_tx_data4/ ch0x_tx_data5/ ch0x_tx_data6/ ch0x_tx_data7/ ch0x_tx_data8	Data is sent in order from data1 to data8. The data in each data is sent in 4-byte little-endian order. The effective data length of the current frame needs to be split according to the total number of frames.	-
input	ch0x_rx_status1	plc rx rolling number: 1 byte, sequentially +1 from 0x00 to 0xff, and circularly accumulated back to 0x00. This parameter must be consistent with the	-

a r a m e t e r s		plc rx rolling request number of the sending end.	
		Total number of frames: 1 byte, indicating the total number of frames that need to be split for a complete data	—
		Current frame number: 1 byte, indicating the current frame number where the current data is located in the total number of frames split by a complete data	—
		Valid data length: 1 byte, indicating the effective data length in the current ch0x_rx_datax field, the maximum length is 32 bytes	—
	ch0x_rx_data1/ ch0x_rx_data2/ ch0x_rx_data3/ ch0x_rx_data4/ ch0x_rx_data5/ ch0x_rx_data6/ ch0x_rx_data7/ ch0x_rx_data8	Receive data in the order of data1~data8. The data in each data is received in 4- byte little-endian order. The effective data length of the current frame needs to be split according to the total number of frames.	—

The sample code and description of the output parameters are as follows:

```
// In this example, the fields of com_tx_status1 are described as follows:

// Control byte 1: serial port type is 422 (0b1), reserved field (0b00), data bits are 8b (0b0), reserved field (0b00), stop bits are 2b (0b10). The binary value of control byte 1 is 0b10000010, which is 0x82 in hexadecimal.

// Control byte 2: baud rate is 38400 (0b0011), reserved field (0b00), parity bit is even parity (0b01). The binary value of control byte 2 is 0b00110001, which is 0x31 in hexadecimal.

// The hexadecimal value of the reserved byte is 0x00.

// The hexadecimal value of the PLC RX rolling request number is 0x00, indicating the first data input request.

// The above 4 bytes are arranged in little-endian byte order and written in the code as 16#00003182.

com_tx_status1 := 16#00003182;

// The character data output in this example is "TEST01", which is 6 bytes of data. One data parameter can contain at most 4 bytes, so data1 and data2 are used to output the data.

// The data in com_tx_data1 is "T", "E", "S", and "T", corresponding to hexadecimal values 0x54, 0x45, 0x53, and 0x54. Arranged in little-endian byte order, it is written in the code as 16#54534554.

com_tx_data1 := 16#54534554;

// The data in com_tx_data2 is "0" and "1", corresponding to hexadecimal values 0x30 and 0x31. Arranged in little-endian byte order, it is written in the code as 16#00003130, where 0000 is zero-padding for com_tx_data2.

com_tx_data2 := 16#00003130;

// In this example, the fields of ch0x_rx_status2 are described as follows:

// The hexadecimal value of the PLC TX rolling request number is 0x01, indicating the first data output request.

// The character data output this time is "TEST01", which is 6 bytes of data.

// Total number of frames: one frame can contain up to 32 bytes of data. Therefore, outputting 6 bytes of data requires only 1 frame, and the hexadecimal value of the total number of frames is 0x01.

// Current frame number: since the total number of frames is 1, the current frame number is 1, corresponding to hexadecimal value 0x01.

// The valid data length is 6 bytes, corresponding to hexadecimal value 0x06.

// The above 4 bytes are arranged in little-endian byte order and written in the code as 16#06010101.

com_tx_status2 := 16#06010101;
```

Examples and descriptions of input parameters are as follows:

```
// Assume that the character data input this time is "HELLOWORLD",
// which is 10 bytes of data. One data parameter can contain at most 4
// bytes, so the data will be input into data1, data2, and data3.

// The 4 bytes "H", "E", "L", and "L" will be input into com_rx_data1,
// corresponding to hexadecimal values 0x48, 0x45, 0x4C, and 0x4C.
// Arranged in little-endian byte order, this is 16#4C4C4548.
// The com_rx_data1 parameter will display 16#4C4C4548.

// The 4 bytes "O", "W", "O", and "R" will be input into com_rx_data2,
// corresponding to hexadecimal values 0x4F, 0x57, 0x4F, and 0x52.
// Arranged in little-endian byte order, this is 16#524F574F.
// The com_rx_data2 parameter will display 16#524F574F.

// The 2 bytes "L" and "D" will be input into com_rx_data3,
// corresponding to hexadecimal values 0x4C and 0x44. Arranged in little-
// endian byte order, this is 16#0000444C, where 0000 is zero-padding
// for com_rx_data3.
// The com_rx_data3 parameter will display 16#0000444C.

// In this example, the fields of com_rx_status1 are described as
// follows:

// The PLC RX rolling number of the serial port module must be
// consistent with the user's PLC RX rolling request number. The user
// requests RX data from the serial port module through the PLC RX rolling
// request number, and the serial port module outputs RX data to the user
// according to the corresponding PLC RX rolling number. Assume that the
// user requests the first output data from the serial port module this
// time; the hexadecimal value of the PLC RX rolling number is 0x00.

// The character data input this time is "HELLOWORLD", which is 10
// bytes of data.

// Total number of frames: one frame can contain up to 32 bytes of
// data. Therefore, inputting 10 bytes of data requires only 1 frame,
// and the hexadecimal value of the total number of frames is 0x01.

// Current frame number: since the total number of frames is 1, the
// current frame number is 1, corresponding to hexadecimal value 0x01.

// The valid data length is 10 bytes, corresponding to hexadecimal
// value 0x0A.

// The above 4 bytes are arranged in little-endian byte order as
// 16#0A010101.
// The com_rx_status1 parameter will display 16#0A010101.
```

4.2 Used with PROFINET Coupler

This section provides a CODESYS programming example for using the INN-A3-MODS04COM 4-channel serial communication module with the INN-A3-CPL-PN PROFINET Coupler. The steps are as follows:

- 1) In the CODESYS menu bar, click "Tools" and select "Device Repository".

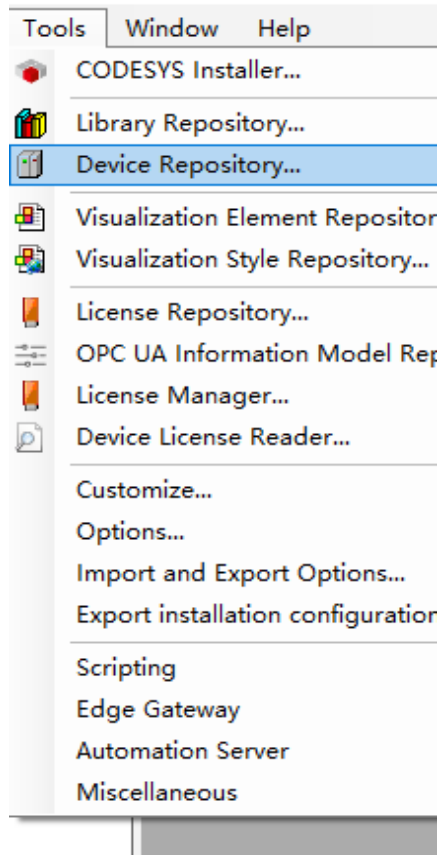


Figure 19 Tools menu drop-down box19

- 2) In the "Device Repository" window interface, click "Install", select the provided GSDML*.xml format file and open it. After receiving the prompt "The relevant device has been installed in the device repository", click "Close".



Figure 20 Installing the device into the device repository20

- 3) In the "Device" window on the left side of CODESYS, right-click the name of the PLC master station ("CODESYS_Control_Win_v3_x64" in the example), and then click "Add Device". In the Add Device window, click "Ethernet Adapter", "Ethernet", and "Add Device" in order to successfully add the device. When finished, just close the window.

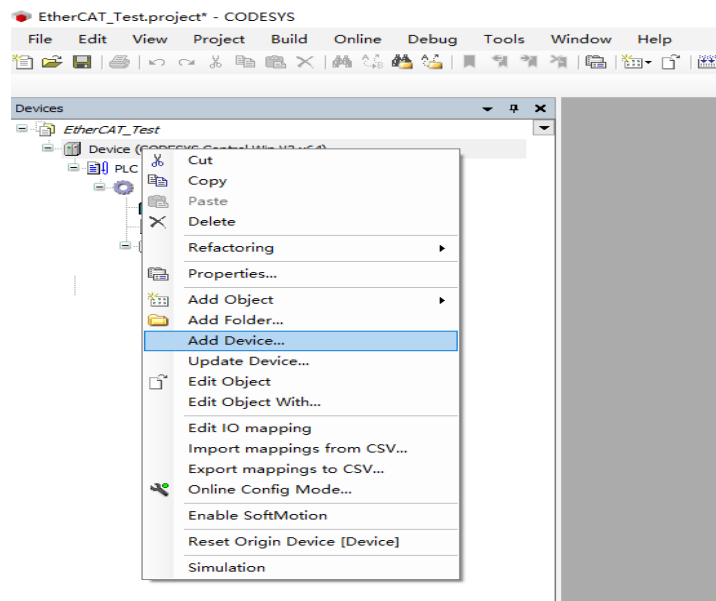


Figure 21 Device window21

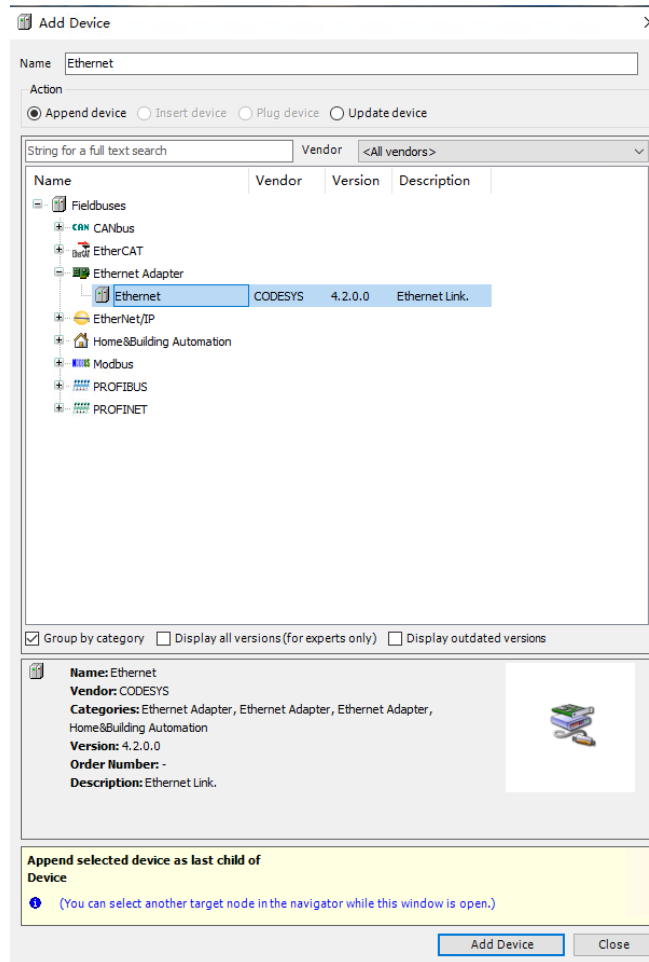


Figure 22 Add device22

- 4) In the network settings of the computer system, set the network interface of the machine that is actually connected to the Coupler to a static IP (use an IPv4 address). In the device window of CODESYS, double-click the Ethernet device, then click the "Browse" button after "Network Interface" in the right window, and select the network interface with a static IP set to ensure normal communication. After completing the settings, click "Confirm" to save the network configuration. The interface will automatically synchronize the IP address, subnet mask and default gateway of the network interface.

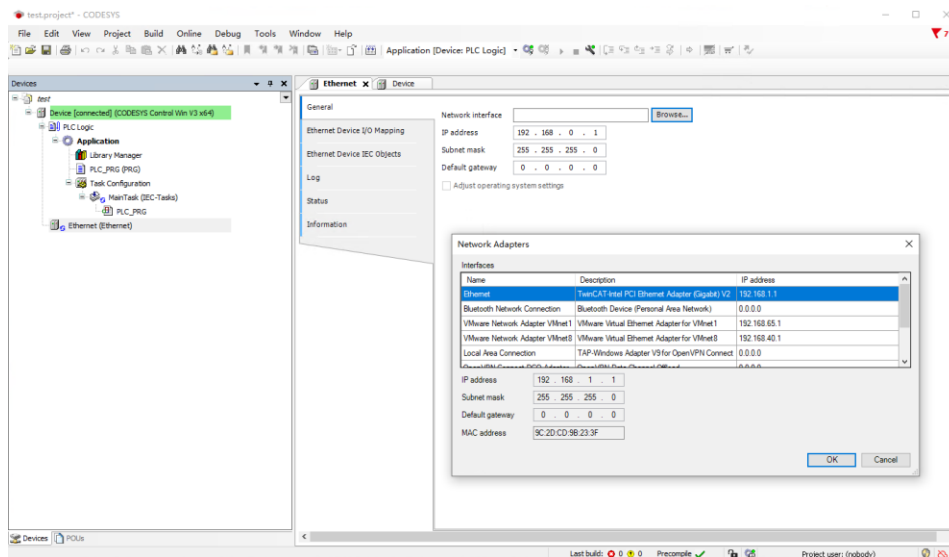


Figure 23 Select network interface

- 5) In the "Devices" window on the left side of CODESYS, right-click "Add Device" on "CODESYS_Control_Win_v3_x64", then click "Ethernet", "Add Device". When finished, just close the window.

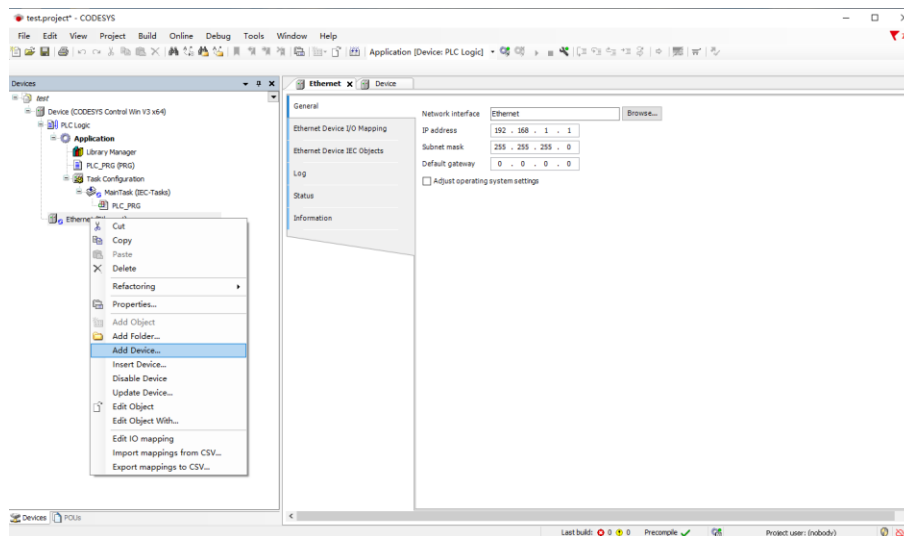


Figure 24 Device window

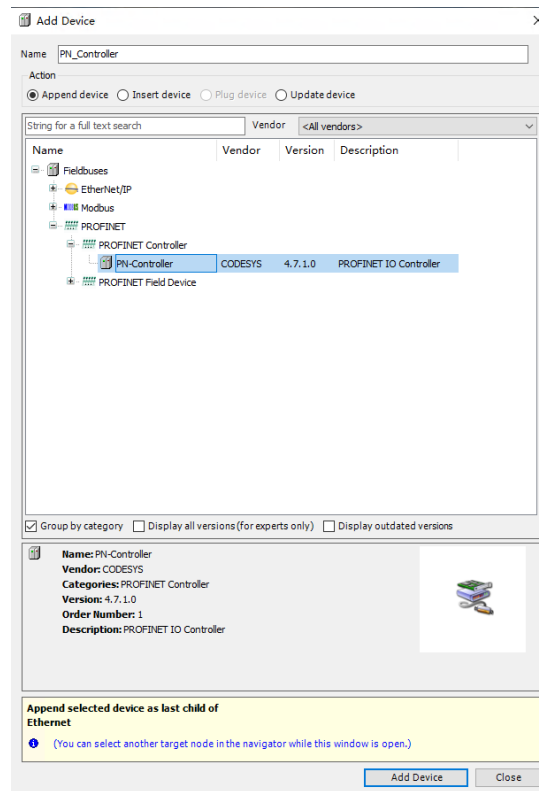


Figure 25 Add Ethernet device25

- 6) Right-click the Ethernet device and click "Add Device" in the pop-up menu. Select PN-Controller and click "Add Device" to complete adding the PROFINET master station.

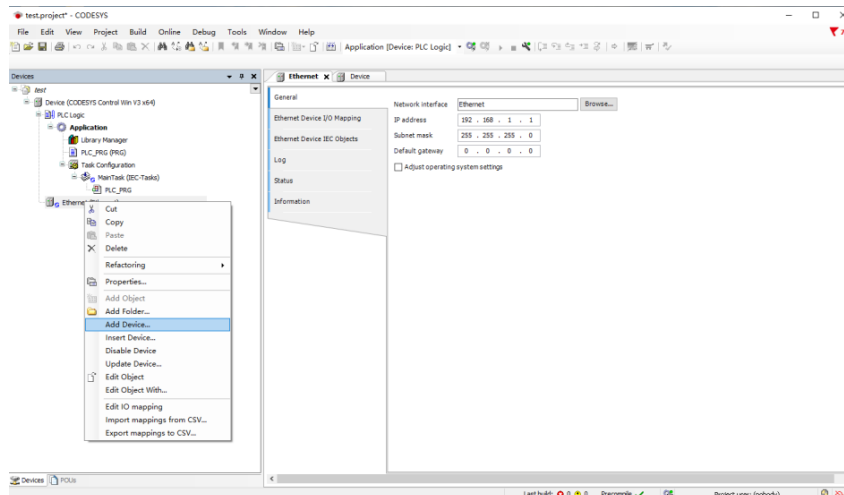


Figure 26 Add device26

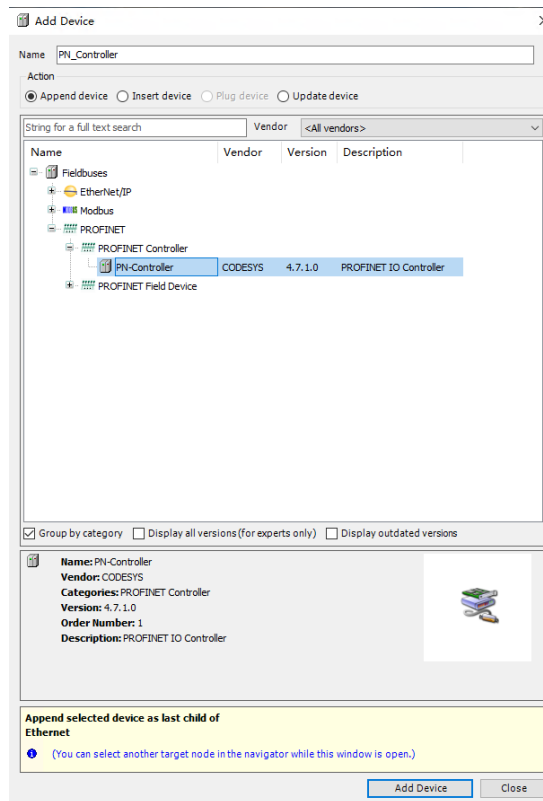


Figure 27 Add PN-Controller27

- 7) Double-click PN-Controller, click to open the "Topology" window on the right, and click the refresh button on the interface to scan the master and slave devices in the network.

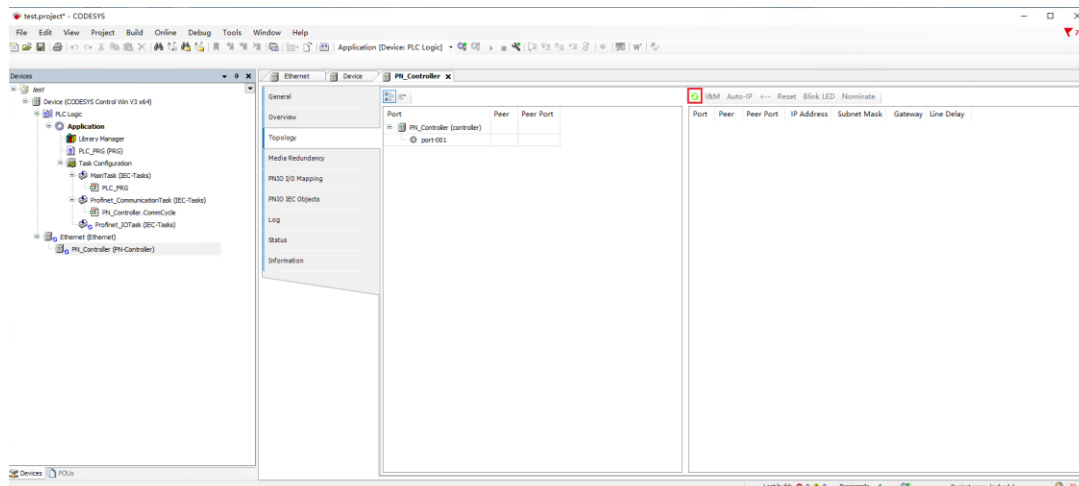


Figure 28 Scanning device28

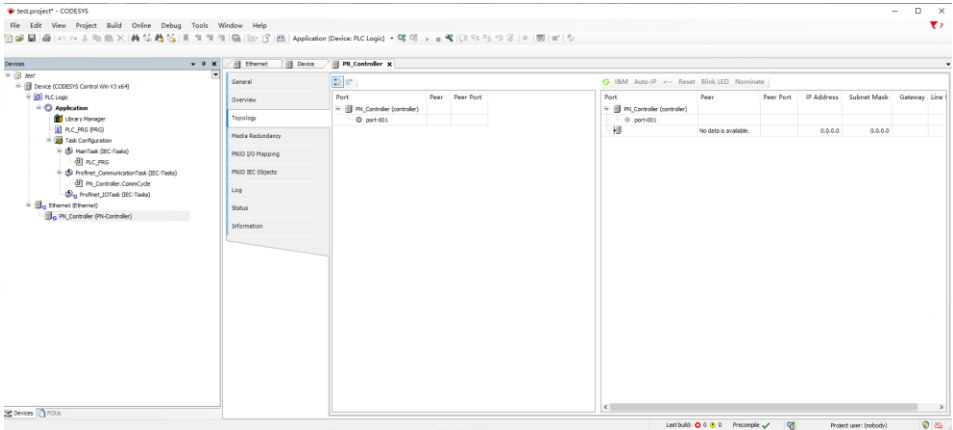


Figure 29 Scan results29

- 8) Select the PN Coupler device, configure the IP address, subnet mask and gateway, and then click the "Specify" button. After completing the configuration, check the port information to confirm that the settings have taken effect.

 CAUTION

The IP address set in this step must be in the same network segment as the network interface IP address configured in step 4.

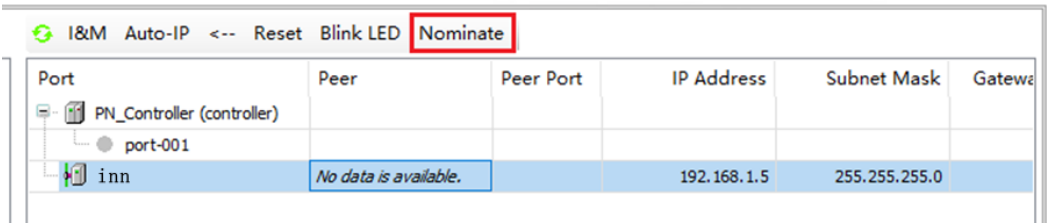


Figure 30 Configure device IP30

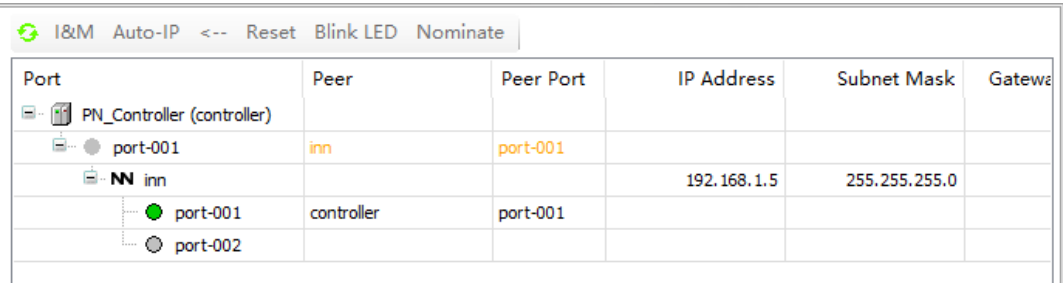


Figure 31 Configuration takes effect31

- 9) In the "Device" window on the left side of CODESYS, right-click the PN-Controller device and click "Add Device" in the pop-up menu. In the Add Device window, select "INN-A3-CPL-PN" under the PROFINET/PROFINET slave station/IO/INN/INN PROFINET IRT DEVKIT level, and then click "Add Device" to successfully add the PROFINET slave station.

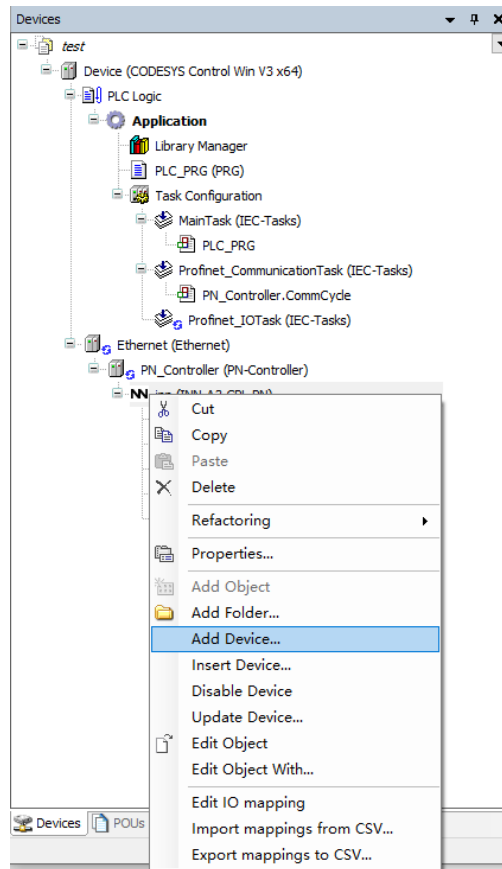


Figure 32 Click "Add Device"32

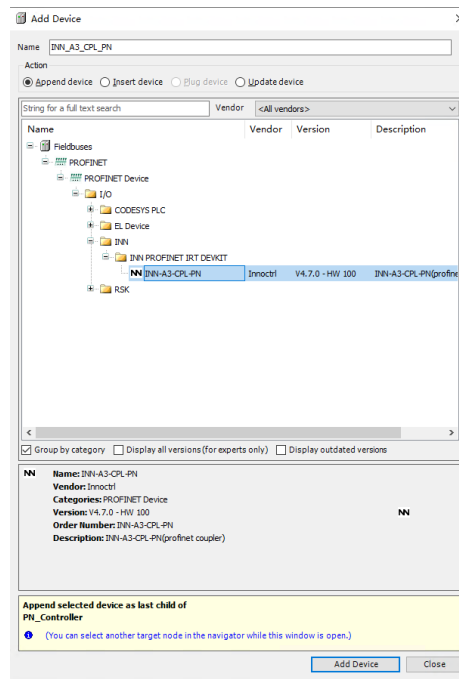


Figure 33 Click "Add Device"33

- 10) Add corresponding Couplers and I/O modules based on existing modules.

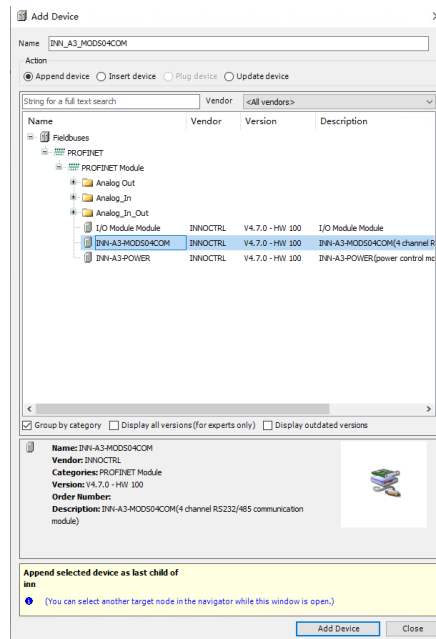


Figure 34 Add Coupler and I/O module34

- 11) In the "Device" window on the left side of CODESYS, double-click the serial port module and configure the basic configuration parameters of the serial port module in the general window.

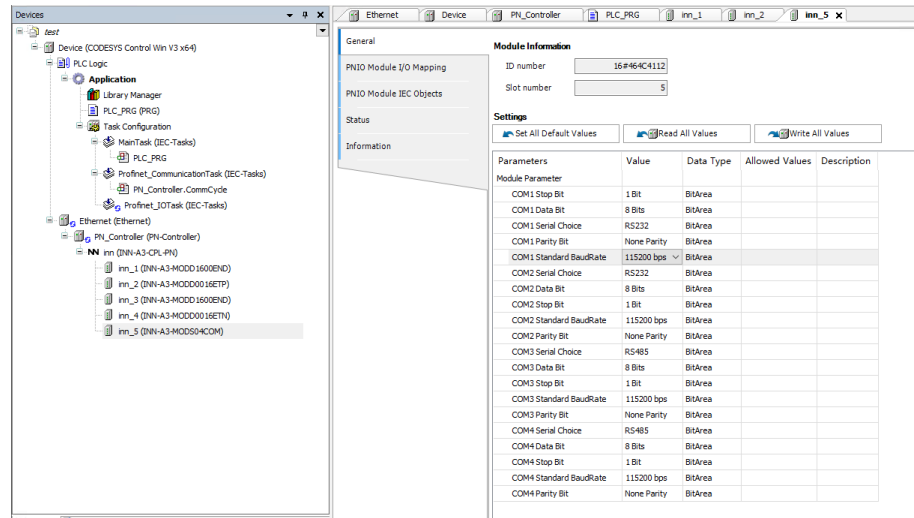


Figure 35 Configure basic parameters of the serial port module35

- 12) Write code and create variables.

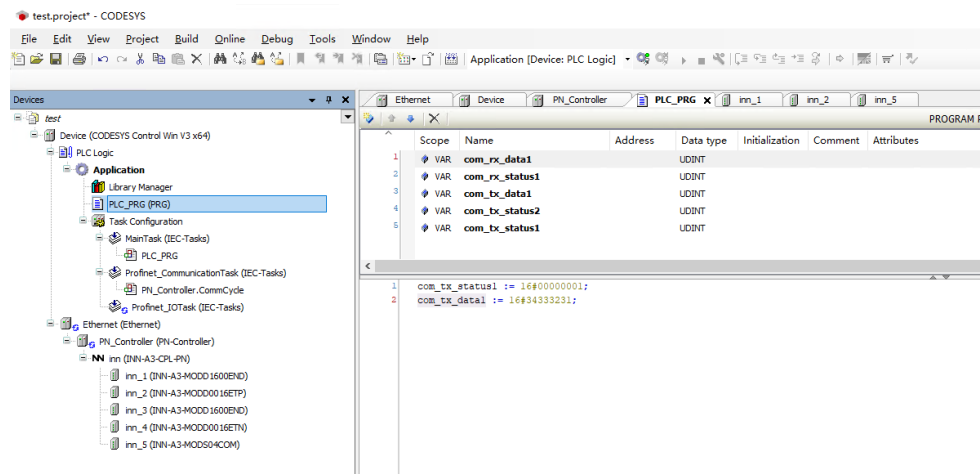


Figure 36 code window36

- 13) To associate variables with I/O modules, users need to fill in each byte in the parameter in order according to big-endian byte order.

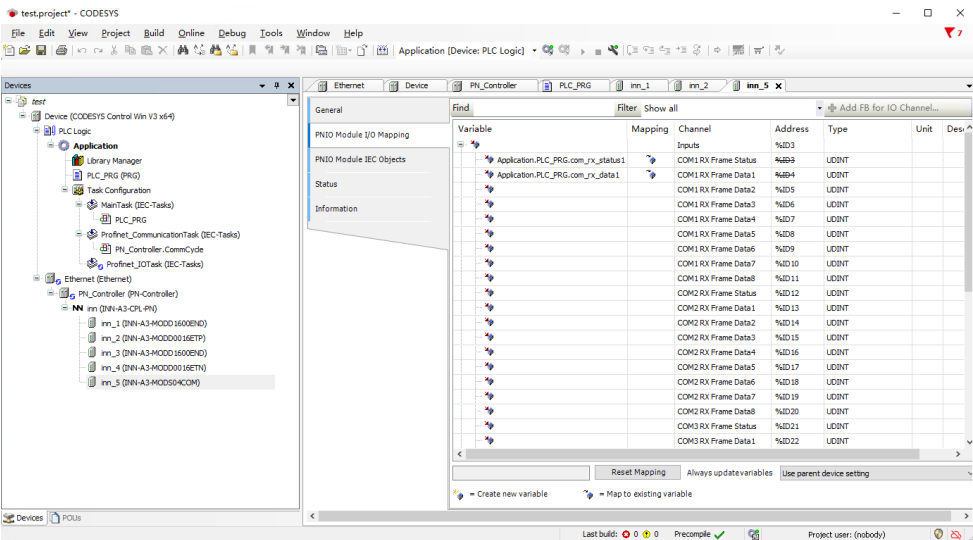


Figure 37 Set input parameters37

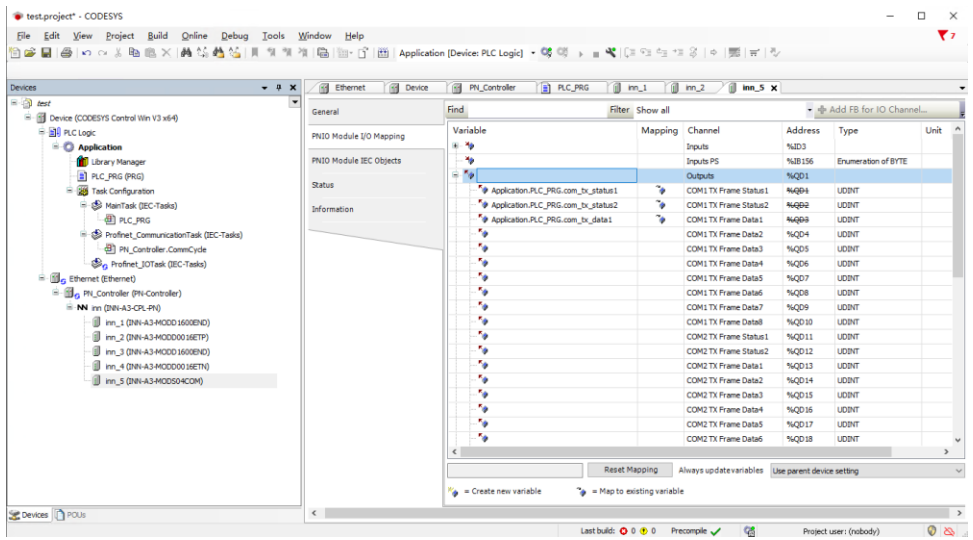


Figure 38 Set output parameters38

Please see the following table for detailed description of input and output parameters:

 CAUTION

In the output parameters, you need to fill in the COM1 TX Frame Status1 and COM1 TX Frame Datax parameters first, and then fill in the COM1 TX Frame Status2 parameter, otherwise the output data will be wrong.

Table 12 Input and output parameter table12

parameters		Byte content description	
O u t p u t p a r a m e t e r s	COM1 TX Frame Status1 Arranged in 4-byte big-endian order	Reserved byte: 1 byte, not used yet	–
		Reserved byte: 1 byte, not used yet	–
		Reserved byte: 1 byte, not used yet	–
		plc rx rolling request number: 1 byte, from 0x00 to 0xff, +1 to 0xff, and then return to 0x00 to accumulate in a loop	–
	COM1 TX Frame Status2 Arranged in 4-byte big-endian order	plc tx rolling request number: 1 byte, from 0x00 + 1 to 0xff, and then return to 0x00 to accumulate in a loop	–
		Total number of frames: 1 byte, indicating the total number of frames that need to be split for a complete data	–
		Current frame number: 1 byte, indicating the current frame number where the current data is located in the total number of frames split by a complete data	–
		Valid data length: 1 byte, indicating the effective data length in the current COM1 TX Frame Datax field, the maximum length is 32 bytes	–
	COM1 TX Frame Data1/ COM1 TX Frame Data2/	Data is sent in order from data1 to data8. The data in each data is sent in 4-byte big-endian order. The	–

	COM1 TX Frame Data3/ COM1 TX Frame Data4/ COM1 TX Frame Data5/ COM1 TX Frame Data6/ COM1 TX Frame Data7/ COM1 TX Frame Data8	effective data length of the current frame needs to be split according to the total number of frames.	
input parameters	COM1 RX Frame Status1	plc rx rolling number: 1 byte, sequentially +1 from 0x00 to 0xff, and circularly accumulated back to 0x00. This parameter must be consistent with the plc rx rolling request number of the sending end.	—
		Total number of frames: 1 byte, indicating the total number of frames that need to be split for a complete data	—
		Current frame number: 1 byte, indicating the current frame number where the current data is located in the total number of frames split by a complete data	—
		Valid data length: 1 byte, indicating the effective data length in the current COM1 RX Frame Datax field, the maximum length is 32 bytes	—
	COM1 RX Frame Data1/ COM1 RX Frame Data2/ COM1 RX Frame Data3/ COM1 RX Frame Data4/	Receive data in the order of data1~data8. The data in each data is received in 4-byte big endian order. The effective data length of the current frame needs to be split	—

	COM1 RX Frame Data5/ COM1 RX Frame Data6/ COM1 RX Frame Data7/ COM1 RX Frame Data8	according to the total number of frames.	
--	---	---	--

The sample code and description of the output parameters are as follows:

```
// In this example, the fields of com_tx_status1 are described as follows:

// The hexadecimal value of the reserved bytes is 0x000000.

// The hexadecimal value of the PLC RX rolling request number is 0x00, indicating the first data input request.

// The above 4 bytes are arranged in big-endian byte order and written in the code as 16#00000000.

com_tx_status1 := 16#00000000;

// The character data output in this example is "TEST01", which is 6 bytes of data. One data parameter can contain at most 4 bytes, so data1 and data2 are used to output the data.

// The data in com_tx_data1 is "T", "E", "S", and "T", corresponding to hexadecimal values 0x54, 0x45, 0x53, and 0x54. Arranged in big-endian byte order, it is written in the code as 16#54455354.

com_tx_data1 := 16#54455354;

// The data in com_tx_data2 is "0" and "1", corresponding to hexadecimal values 0x30 and 0x31. Arranged in big-endian byte order, it is written in the code as 16#30310000, where 0000 is zero-padding for com_tx_data2.

com_tx_data2 := 16#30310000;

// In this example, the fields of com_tx_status2 are described as follows:

// The hexadecimal value of the PLC TX rolling request number is 0x01, indicating the first data output request.

// The character data output this time is "TEST01", which is 6 bytes of data.

// Total number of frames: one frame can contain up to 32 bytes of data. Therefore, outputting 6 bytes of data requires only 1 frame, and the hexadecimal value of the total number of frames is 0x01.

// Current frame number: since the total number of frames is 1, the current frame number is 1, corresponding to hexadecimal value 0x01.

// The valid data length is 6 bytes, corresponding to hexadecimal value 0x06.

// The above 4 bytes are arranged in big-endian byte order and written in the code as 16#01010106.

com_tx_status2 := 16#01010106;
```

Examples and descriptions of input parameters are as follows:

```
// Assume that the character data input this time is "HELLOWORLD",
// which is 10 bytes of data. One data parameter can contain at most 4
// bytes, so the data will be input into data1, data2, and data3.

// The 4 bytes "H", "E", "L", and "L" will be input into com_rx_data1,
// corresponding to hexadecimal values 0x48, 0x45, 0x4C, and 0x4C.
// Arranged in big-endian byte order, this is 16#48454C4C.
The com_rx_data1 parameter will display 16#48454C4C.

// The 4 bytes "O", "W", "O", and "R" will be input into com_rx_data2,
// corresponding to hexadecimal values 0x4F, 0x57, 0x4F, and 0x52.
// Arranged in big-endian byte order, this is 16#4F574F52.
The com_rx_data2 parameter will display 16#4F574F52.

// The 2 bytes "L" and "D" will be input into com_rx_data3,
// corresponding to hexadecimal values 0x4C and 0x44. Arranged in big-
// endian byte order, this is 16#4C440000, where 0000 is zero-padding
// for com_rx_data3.
The com_rx_data3 parameter will display 16#4C440000.

// In this example, the fields of com_rx_status1 are described as
// follows:

// The PLC RX rolling number of the serial port module must be
// consistent with the user's PLC RX rolling request number. The user
// requests RX data from the serial port module through the PLC RX rolling
// request number, and the serial port module outputs RX data to the user
// according to the corresponding PLC RX rolling number. Assume that the
// user requests the first output data from the serial port module this
// time; the hexadecimal value of the PLC RX rolling number is 0x00.

// The character data input this time is "HELLOWORLD", which is 10
// bytes of data.

// Total number of frames: one frame can contain up to 32 bytes of
// data. Therefore, inputting 10 bytes of data requires only 1 frame,
// and the hexadecimal value of the total number of frames is 0x01.

// Current frame number: since the total number of frames is 1, the
// current frame number is 1, corresponding to hexadecimal value 0x01.

// The valid data length is 10 bytes, corresponding to hexadecimal
// value 0x0A.

// The above 4 bytes are arranged in big-endian byte order as
// 16#0001010A.
The com_rx_status1 parameter will display 16#0001010A.
```



For more information, please visit INNOCTRL
official website:

www.innoctrl.com

Kaihua County, Quzhou City, Zhejiang Province
Room 102, Comprehensive Building, No. 3 Taikang Road, Economic Development Zone



sales@innoctrl.com
www.innoctrl.com