

**XB6S-C01SP**

**Serial Communications Module**

**User Manual**




Nanjing Solidot Electronic Technology Co., Ltd. 2024

**Copyright © Nanjing Solidot Electronic Technology Co., Ltd. 2024. All rights reserved .**

Without the written permission of our company, no unit or individual may excerpt or copy part or all of the contents of this document, or disseminate it in any form.

### **Trademark Statement**

 and other Solidot trademarks are trademarks of Nanjing Solidot Electronic Technology Co., Ltd.

All other trademarks or registered trademarks mentioned in this document are the property of their respective owners.

### **Notice**

The products, services or features you purchase shall be subject to the commercial contracts and terms of Real Point Company. All or part of the products, services or features described in this document may not be within the scope of your purchase or use. Unless otherwise agreed in the contract, Solidot Company does not make any express or implied representations or warranties regarding the contents of this document.

Due to product version upgrades or other reasons, the content of this document will be updated from time to time. Unless otherwise agreed, this document is only a guide for use, and all statements, information and suggestions in this document do not constitute any express or implied warranty.

Nanjing Solidot Electronic Technology Co., Ltd.

11th Floor, Angying Building, No. 91 Shengli Road, Jiangning District, Nanjing City, Jiangsu Province

Postcode: 21 1106

Phone: 4007788929

Website: <http://www.solidotech.com>

# CONTENTS

|  |    |
|--|----|
| 1 Product Introduction.....                  | 1  |
| 1.1 Product Overview.....                    | 1  |
| 1.2 Product Features.....                    | 1  |
| 2 Product Parameters.....                    | 2  |
| 2.1 General Parameter.....                   | 2  |
| 3 Panel.....                                 | 3  |
| 3.1 Panel Structure.....                     | 3  |
| 3.2 Indicator light function.....            | 4  |
| 4 Installation and uninstal.....             | 5  |
| 4.1 External dimensions.....                 | 5  |
| 4.2 Installation Guide.....                  | 6  |
| 4.3 Installation and uninstal steps.....     | 9  |
| 4.4 Installation and uninstal diagram.....   | 10 |
| 5 Wiring.....                                | 17 |
| 5.1 Wiring Diagram.....                      | 17 |
| 5.2 Terminal Block Definition.....           | 18 |
| 6 Operation.....                             | 19 |
| 6.1 Configuration Parameter Definitions..... | 19 |
| 6.2 Node Code Parameters.....                | 21 |
| 6.2.1 Modbus Master Function.....            | 21 |
| 6.2.2 Modbus Master Function Node Code.....  | 21 |
| 6.2.3 Modbus Slave Function.....             | 24 |
| 6.2.4 Modbus Slave Function Node Code.....   | 24 |
| 6.2.5 ExByte Flag.....                       | 27 |
| 6.2.6 Freeport function.....                 | 27 |
| 6.2.7 Freeport Function Node Code.....       | 27 |
| 6.2.8 Control and Status Node Codes.....     | 28 |
| 6.3 Process data.....                        | 29 |
| 6.3.1 Modbus Master Process Data.....        | 30 |
| 6.3.2 Modbus Slave Process Data.....         | 31 |

---

|  |    |
|--|----|
| 6.3.3 Freeport_Request Process Data .....                      | 32 |
| 6.3.4 Freeport_Input Process Data .....                        | 33 |
| 6.3.5 Freeport_Output Process Data .....                       | 34 |
| 6.3.6 Freeport_Input_Output Process Data .....                 | 35 |
| 6.3.7 Modbus Alarm Code .....                                  | 36 |
| 6.4 Module Configuration Description .....                     | 37 |
| 6.4.1 Application in TwinCAT3 software environment .....       | 37 |
| 6.4.2 Application in Sysmac Studio software environment .....  | 51 |
| 6.4.3 Application in TIA Portal V17 software environment ..... | 66 |

# 1 Product Introduction

---

## 1.1 Product Overview

XB6S-C01SP is a plug-in 1-channel serial communication module, using X-bus backplane, adapted to our XB6S series coupler module, can realize Modbus master-slave, Freeport serial communication function, the module occupies a small space, the data interaction is simple to handle, and it can satisfy the needs of different application scenarios of serial communication.

## 1.2 Product Features

- Supports multiple communication modes  
Eight modes MRM/MRS/MAM/MAS/FP\_Request/FP\_Input\_Output/FP\_Input/FP\_Output can be set (see [6.1 Configuration Parameters](#) for details).
- Supports three communication interfaces  
RS485/RS422/RS232 interfaces.
- Supports three communication protocols  
Modbus RTU/ Modbus ASCII/Freeport.
- Small volume  
Compact structure and small space occupation.
- Easy to diagnose  
Innovative channel indicator design, close to the channel, channel status obvious, easy to detect and maintain.
- Easy to configure  
Simple configuration and support for mainstream masters.
- Easy to install  
DIN 35 mm standard rail installation  
Adopts spring-type terminals for easy and quick wiring.

# 2 Product Parameters

## 2.1 General Parameter

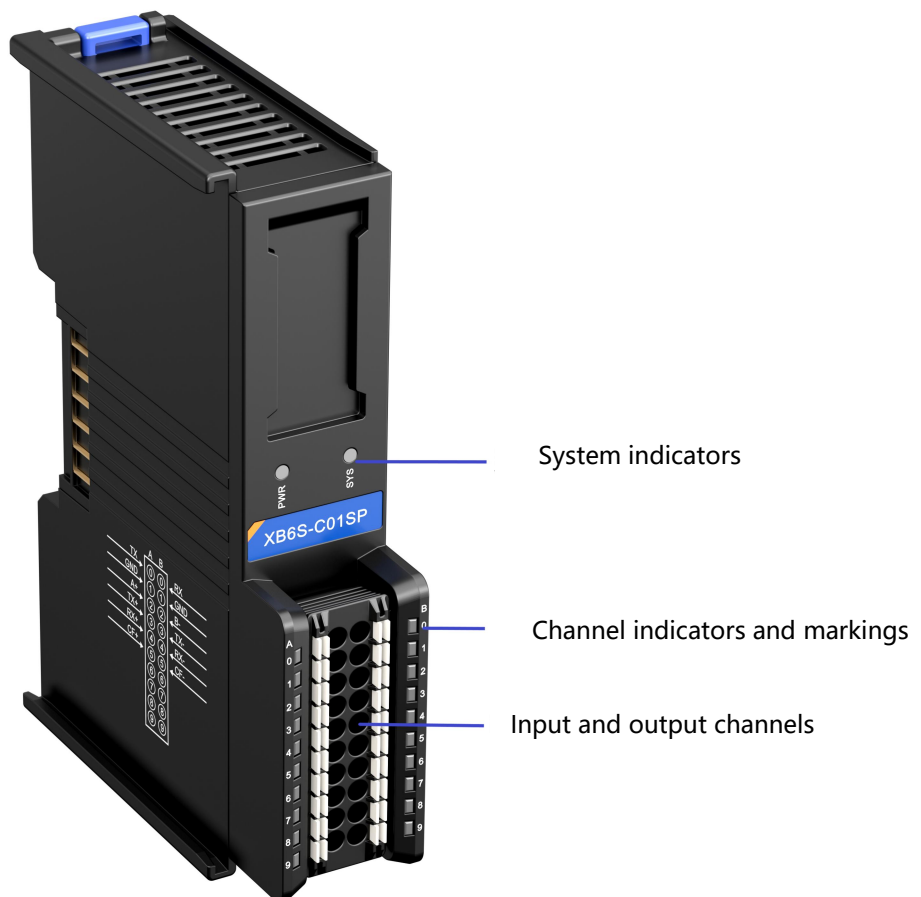
| Interface Parameter                            |                                    |
|--|------------------------------------|
| Product Model                                  | XB6S-C01SP                         |
| Bus Protocol                                   | X-bus                              |
| Bus Input Power<br>Supply Rated Voltage        | 5VDC (4.5V~5.5V)                   |
| Rated Current<br>Consumption                   | 230mA                              |
| Power Consumption                              | 1.15W                              |
| Technical Parameters                           |                                    |
| Channel Number                                 | 1 channel                          |
| Communication<br>Interface Type <sup>[1]</sup> | RS232, RS485, RS422                |
| Communication<br>Protocols                     | Modbus RTU, Modbus ASCII, Freeport |
| Baud   | 1200bps~115200bps                  |
| Weight   | 90g                                |
| Size   | 106.4 x 25.7 x 72.3mm              |
| Wiring Method                                  | Screwless Quick Plugs              |
| Installation                                   | DIN 35mm rail installation         |
| Operating<br>Temperature                       | -20°C~+60°C                        |
| Storage Temperature                            | -40°C~+80°C                        |
| Relative Humidity                              | 95%, non-condensing                |
| Protection class                               | IP20                               |

Note [1]: Only one of the RS232, RS485 and RS422 interfaces can be used at the same time.

# 3 Panel

## 3.1 Panel Structure

### Name of each part of the product



## 3.2 Indicator light function

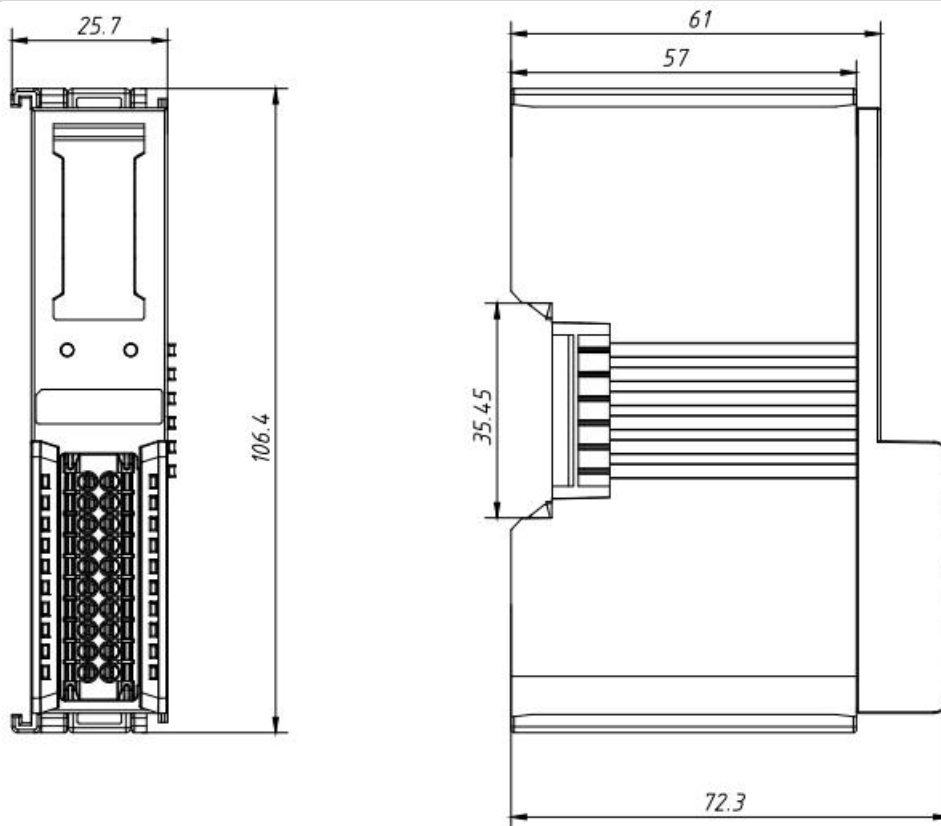
| Name                     | Markings       | Color | Status        | Status Description  |
|--------------------------|----------------|-------|---------------|---|
| Power indicator          | PWR            | GREEN | ON            | Power supply is normal  |
|                          |                |       | OFF           | The product is not powered up or the power supply is abnormal                         |
| Communication Indicator  | SYS            | GREEN | ON            | The system is functioning normally  |
|                          |                |       | Flashing 1Hz  | No business data interaction, waiting for business data interaction to be established |
|                          |                |       | Flashing 10Hz | Firmware Upgrade  |
|                          |                |       | OFF           | System not working  |
| Input Channel Indicator  | 0 (right side) | GREEN | FLASHING      | Channel has data reception  |
|                          |                |       | OFF           | No data reception on channel  |
| Output Channel Indicator | 0 (left side)  | GREEN | FLASHING      | Channel has data to send  |
|                          |                |       | OFF           | No data sent by channel   |



# 4 Installation and uninstall

## 4.1 External dimensions

### External specifications (unit mm)



## 4.2 Installation Guide

### Precautions for installation\uninstall

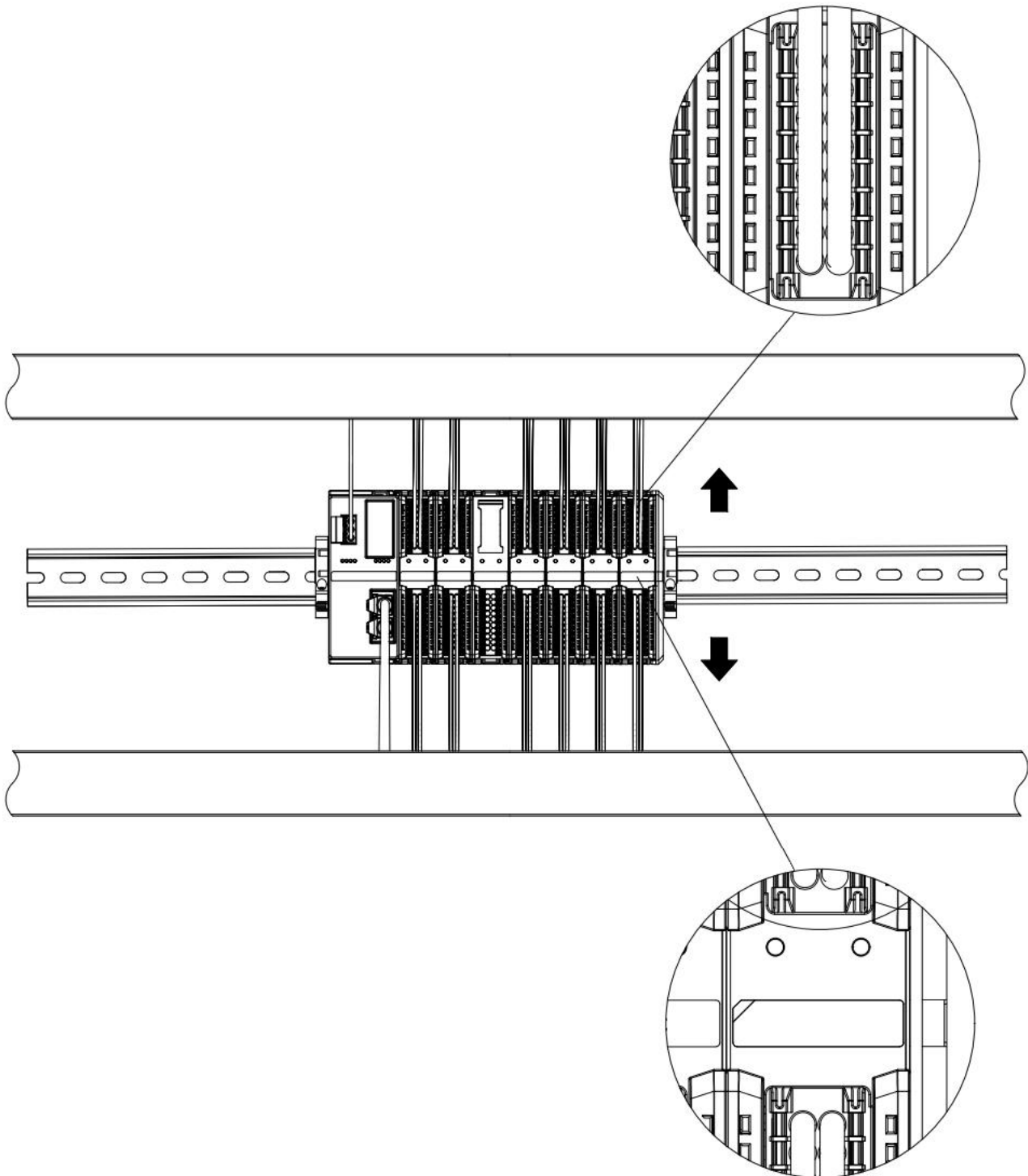
- The module protection class is IP20 and the module needs to be installed in the cabinet for indoor use.
- Ensure that the cabinet is well ventilated (e.g., the cabinet is fitted with an exhaust fan).
- Do not install this equipment next to or above equipment that may cause overheating.
- Be sure to mount the module vertically on the fixed rails and maintain air circulation around it (at least 50 mm air circulation space above and below the module).
- Once the module is installed, be sure to secure the module by installing rail mounts on both ends.
- Be sure to disconnect the power supply when installing/uninstalling.
- Once the module is installed, it is recommended that wiring and cabling be done in accordance with the top and bottom alignments.



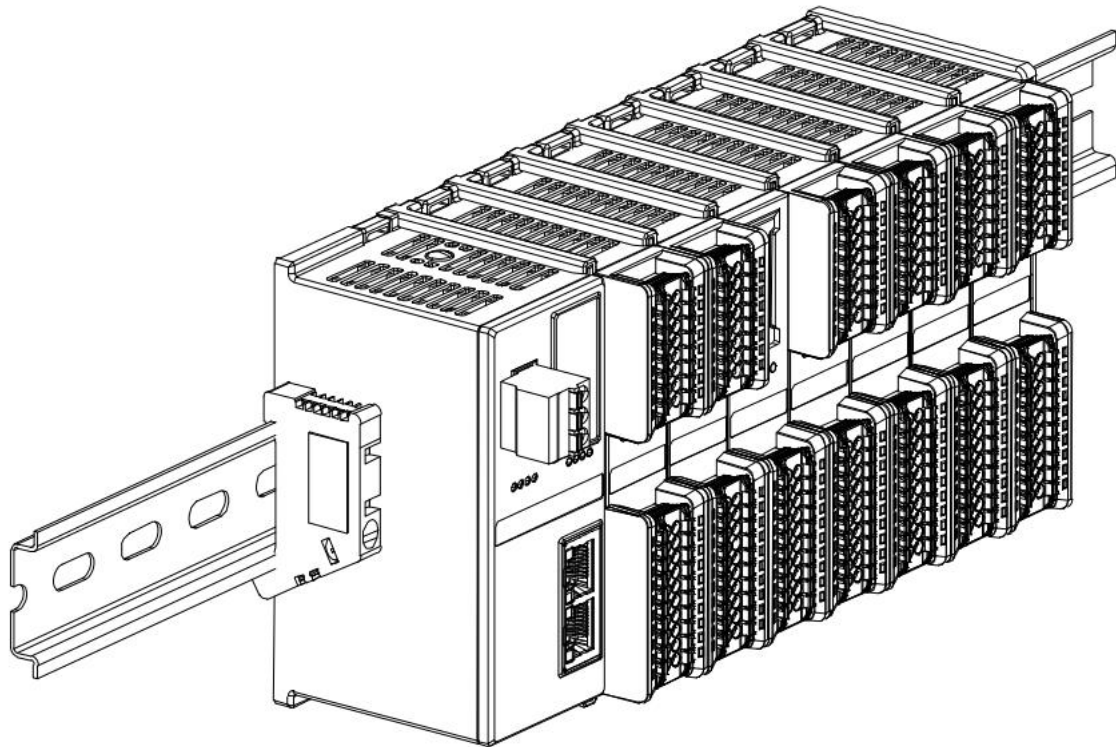
#### warnings

- The protection provided by the device may be jeopardized if it is not used in accordance with the product user manual.

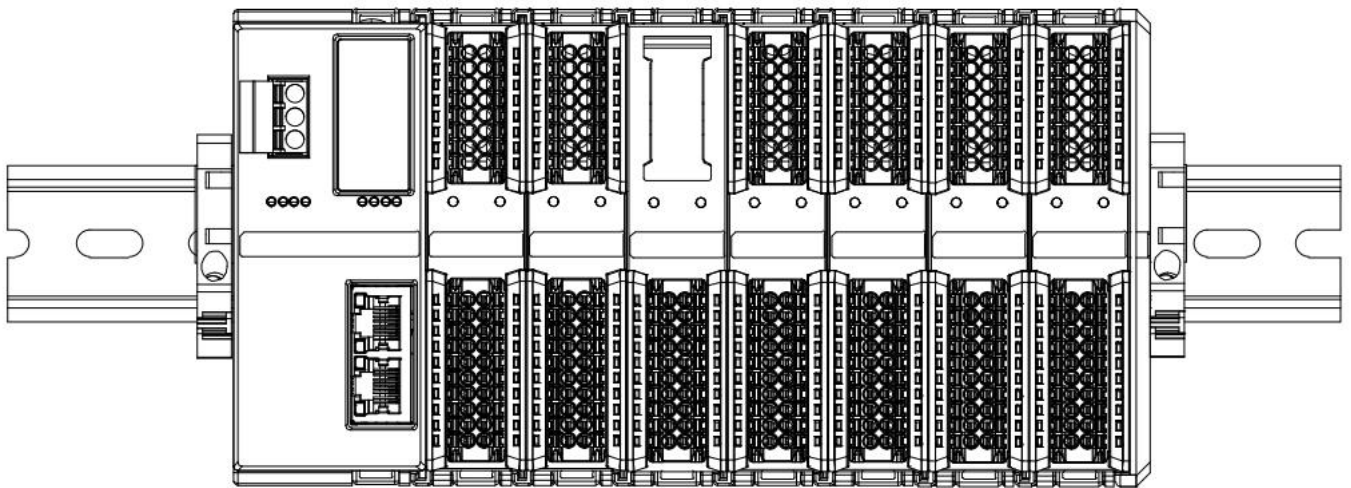
**Module installation schematic, minimum clearance above and below ( $\geq 50\text{mm}$ )**



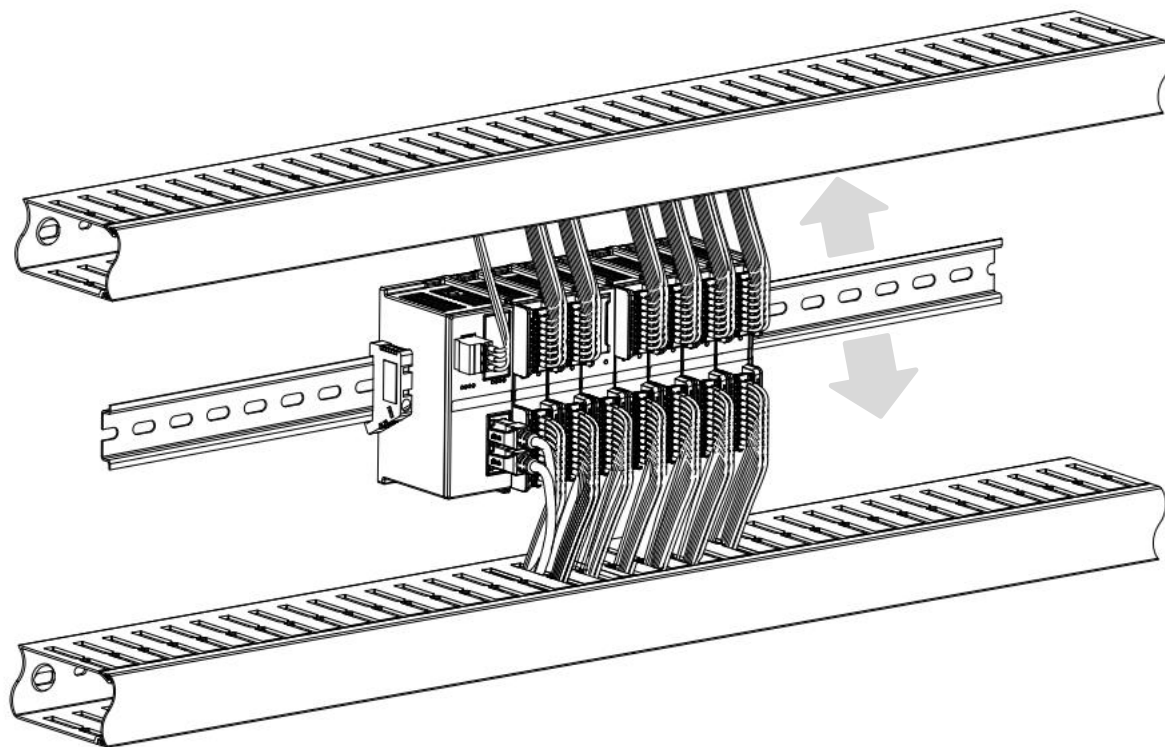
**Ensure that the module is installed vertically on the fixed rail**



**Be sure to install the rail fixings**



**Schematic diagram of upper and lower wiring of the module**



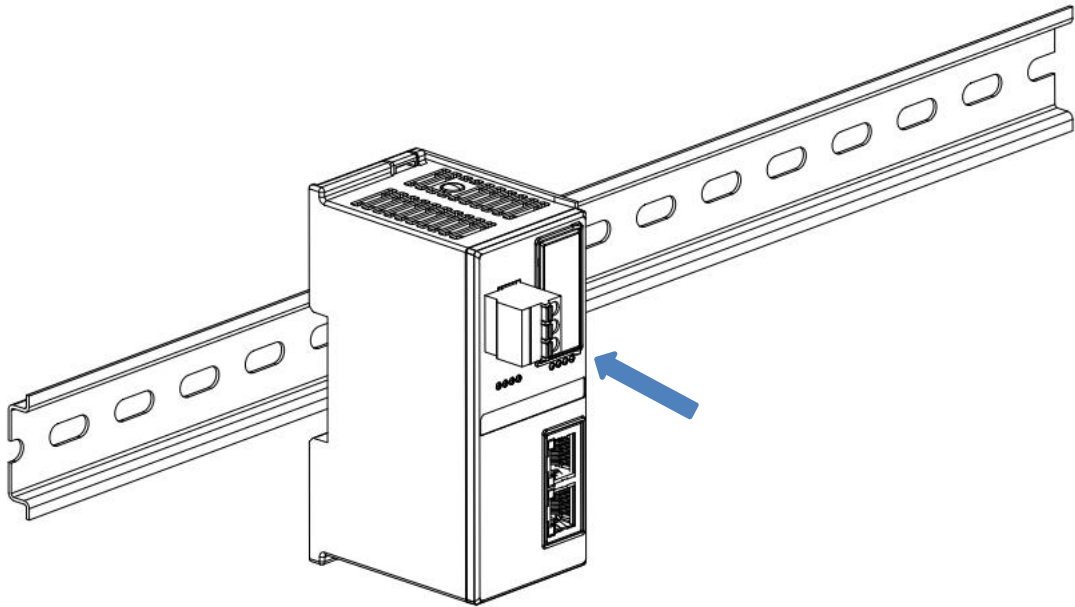
**4.3 Installation and uninstal steps**

| <b>Module Installation and uninstal</b> |   |
|---|---|
| Module Installation Steps               | 1. Install the coupler module first on the rail that has been fixed.                                      |
|   | 2. Install the required I/O modules or function modules in order to the right of the coupler module.      |
|   | 3. After installing all the required modules, install the end cap to complete the assembly of the module. |
|   | 4. Install the rail fixings at both ends of the coupler module and end cap to fix the module.             |
| Module Uninstall Steps                  | 1. Loosen the guide rail fixings at both ends of the module.  |
|   | 2. Use a slotted screwdriver to pry off the module snap.  |
|   | 3. Pull out the uninstalled module.   |

## 4.4 Installation and uninstal diagram

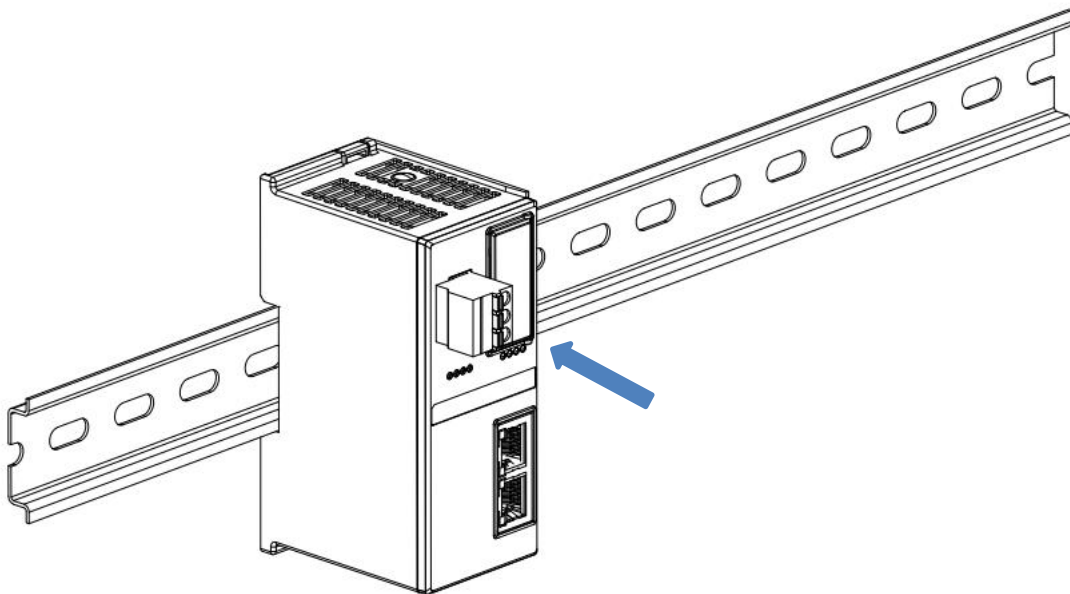
### Coupler Module Installation

- Align the coupler module vertically in the rail slot as shown in Figure ① below.



①

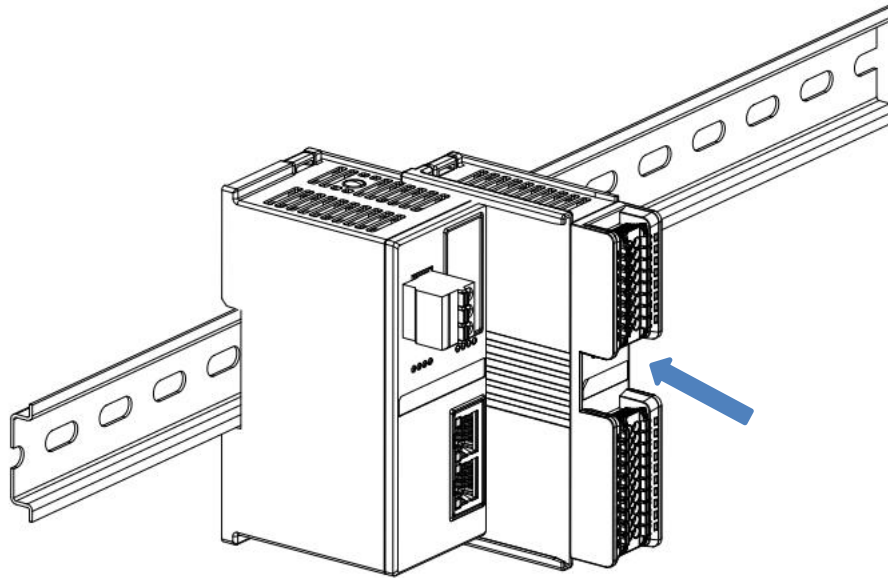
- Press the coupler module in the direction of the guide rail, and when you hear a "click" sound, the module is installed in place, as shown in Figure 2 below.



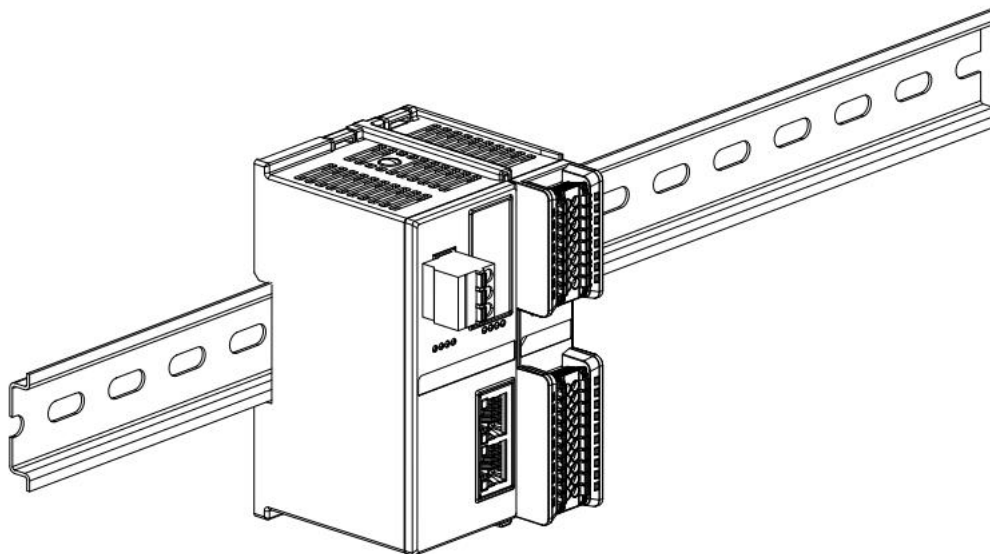
②

## I/O Module Installation

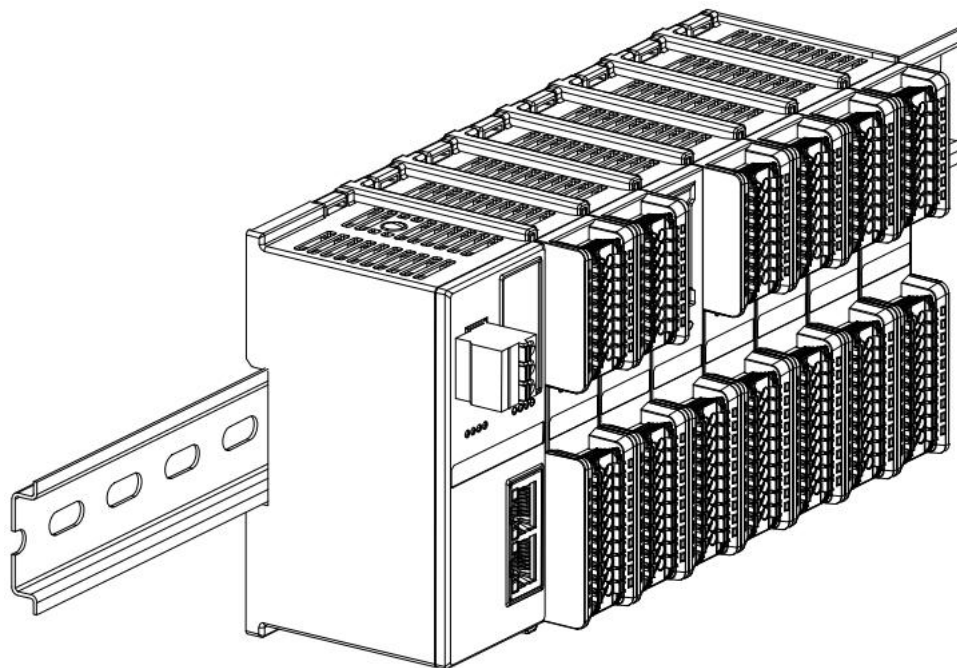
- Install the required I/O modules or function modules one by one according to the above steps of installing coupler modules as shown in Figures ③, ④ and ⑤ below, and push them in, and the modules will be installed in place when you hear the "click" sound.



③



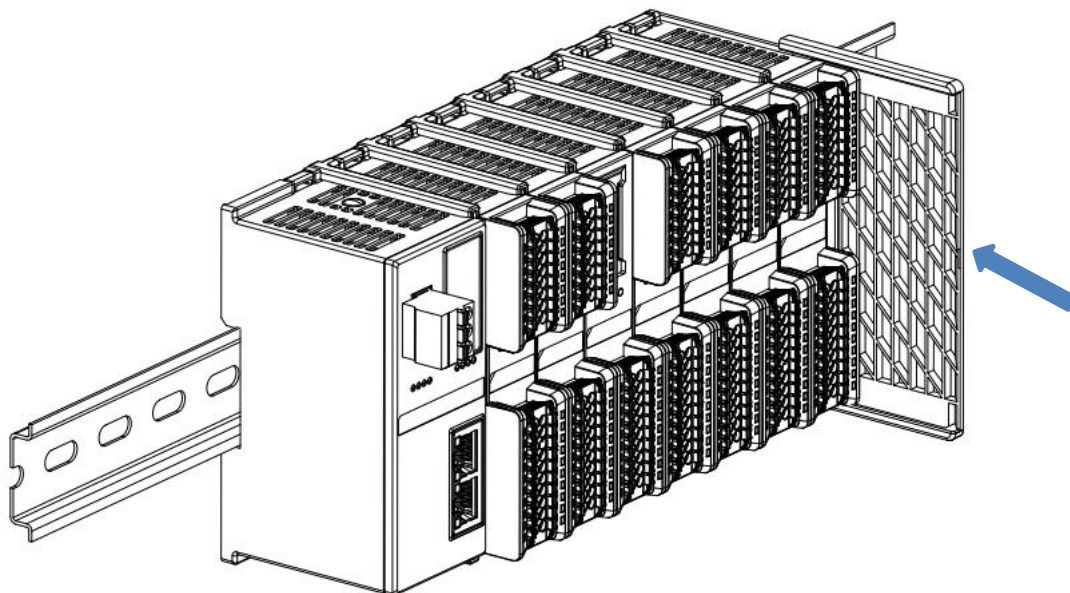
④



⑤

### End cap installation

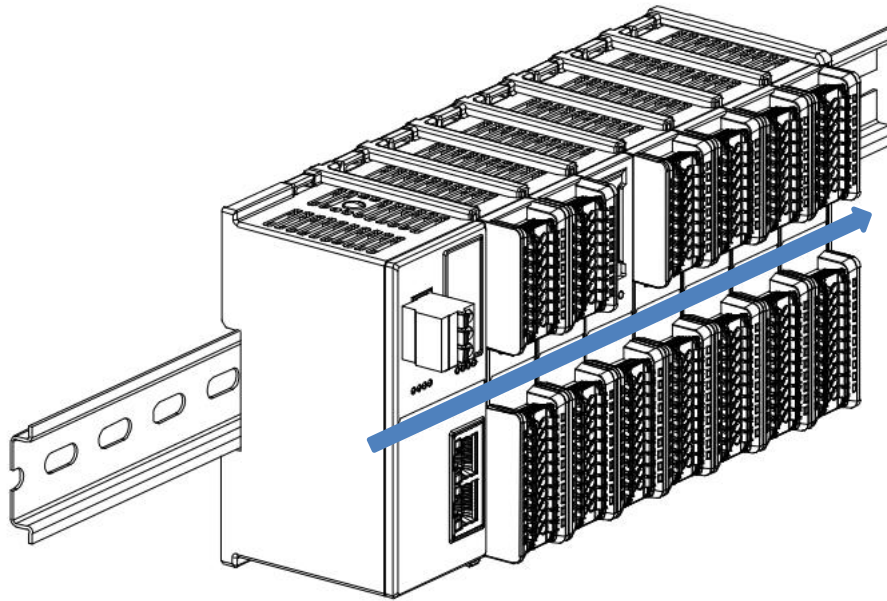
- Install the end cap on the right side of the last module, with the grooved side of the end cap aligned with the guide rail. Refer to the installation method of the I/O module for the installation method, and push the end cap inward into place, as shown in Figure ⑥ below.



⑥



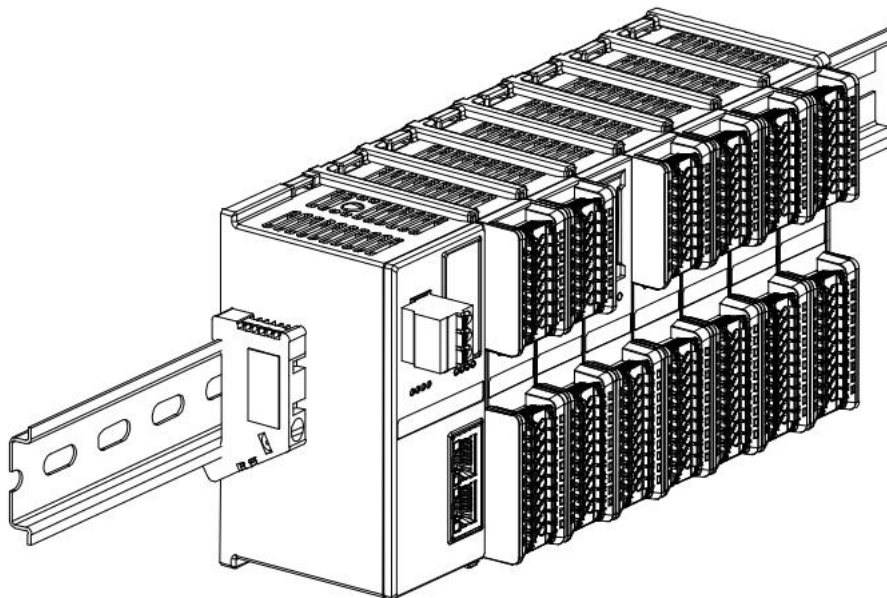
- After the end cap is installed, check whether the entire front of the module is flat to ensure that all modules and end cap are installed in place and the front is flush, as shown in Figure ⑦ below.



⑦

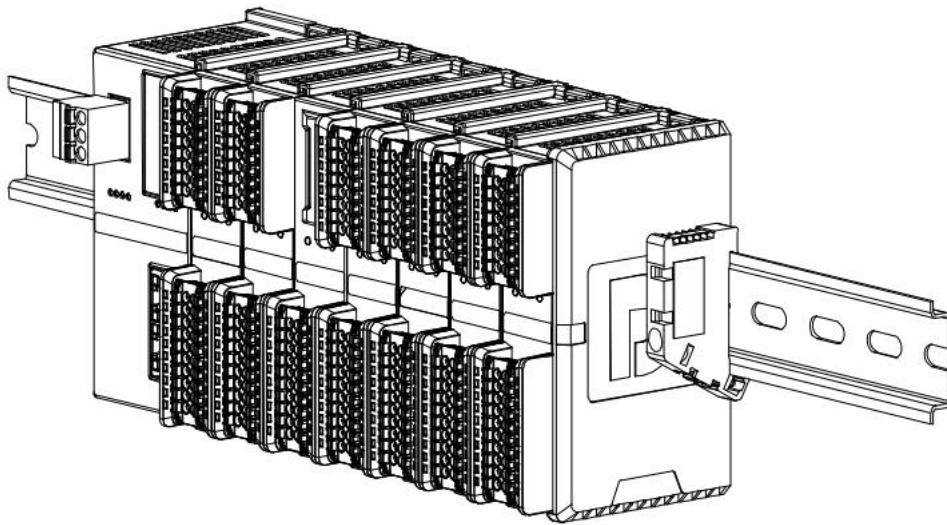
### installation of guide rail fixings

- Install and tighten the rail retainer firmly against the left side face of the coupler as shown in Figure ⑧ below.



⑧

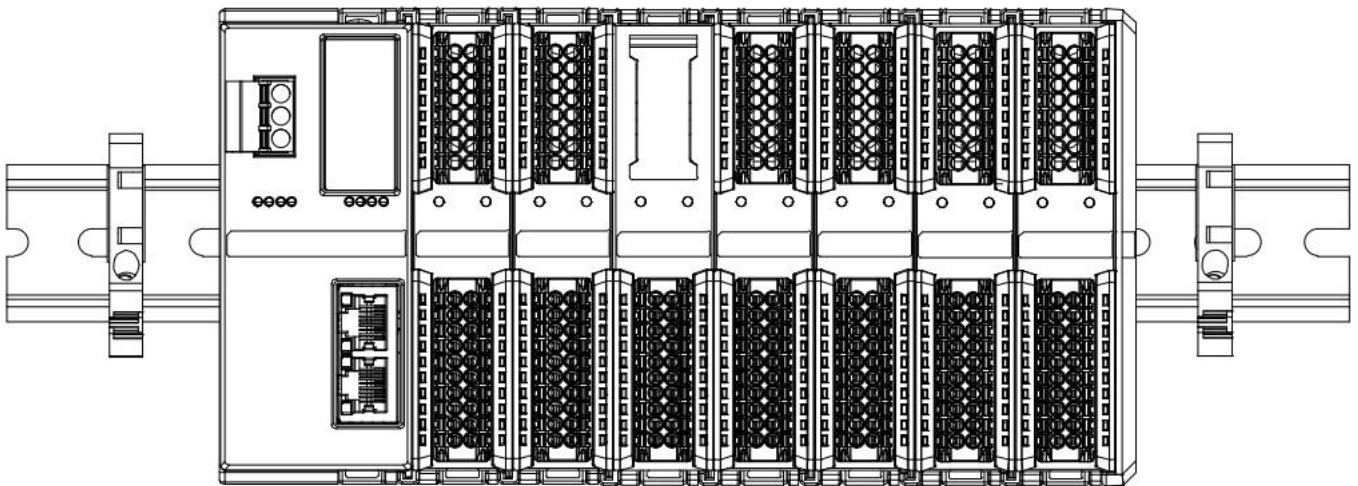
- Install the rail fixing on the right side of the end cap, first push the rail fixing firmly in the direction of the coupler to ensure that the module is installed tightly, and lock the rail fixing with a screwdriver, as shown in Figure ⑨ below.



⑨

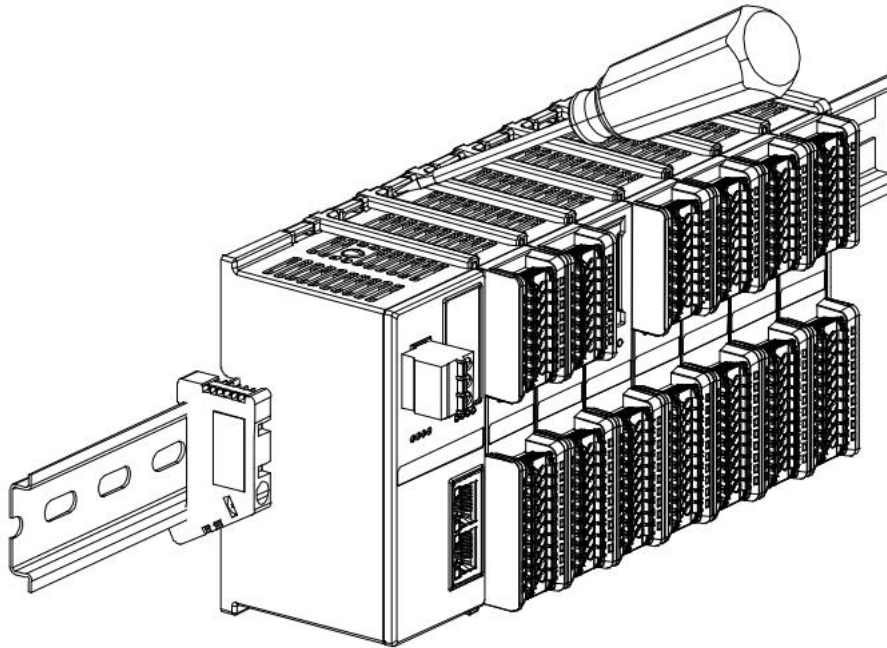
## Uninstall

- Loosen the rail retainer at one end of the module with a screwdriver and move it to one side, making sure there is clearance between the module and the rail retainer, as shown in Figure ⑩ below.

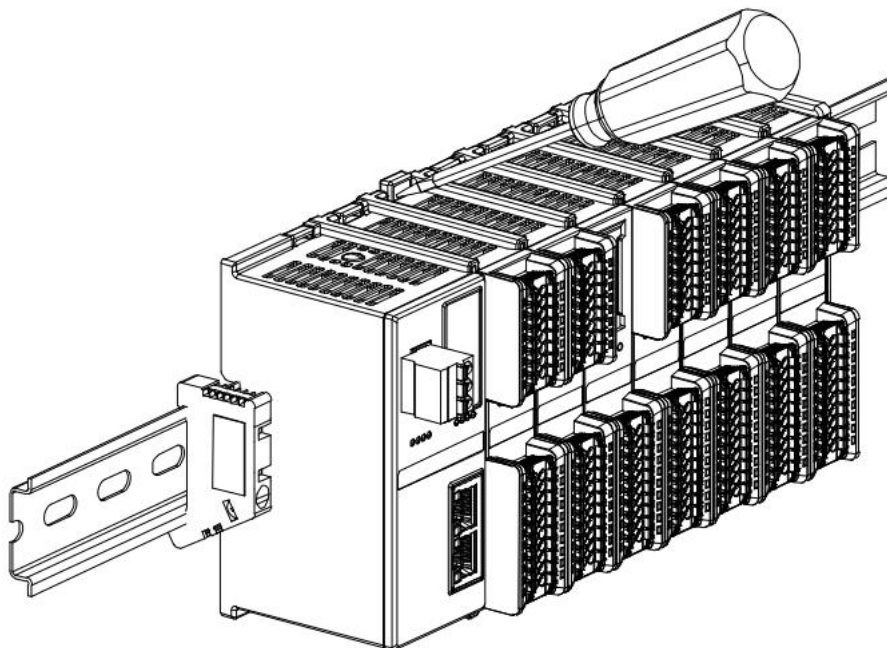


⑩

- Insert a flat-head screwdriver into the snap of the module to be removed, and apply pressure (hear the rattle) in the direction of the module sideways, as shown in the following figures ⑪ and ⑫. **Note:** There is one snap on the top and bottom of each module, and they are all operated in this way.

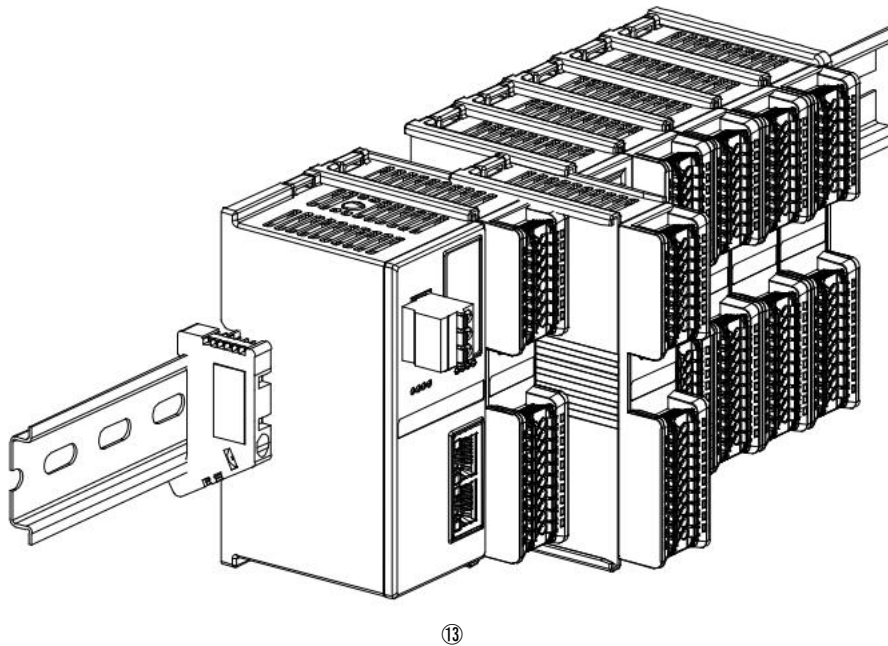


⑪



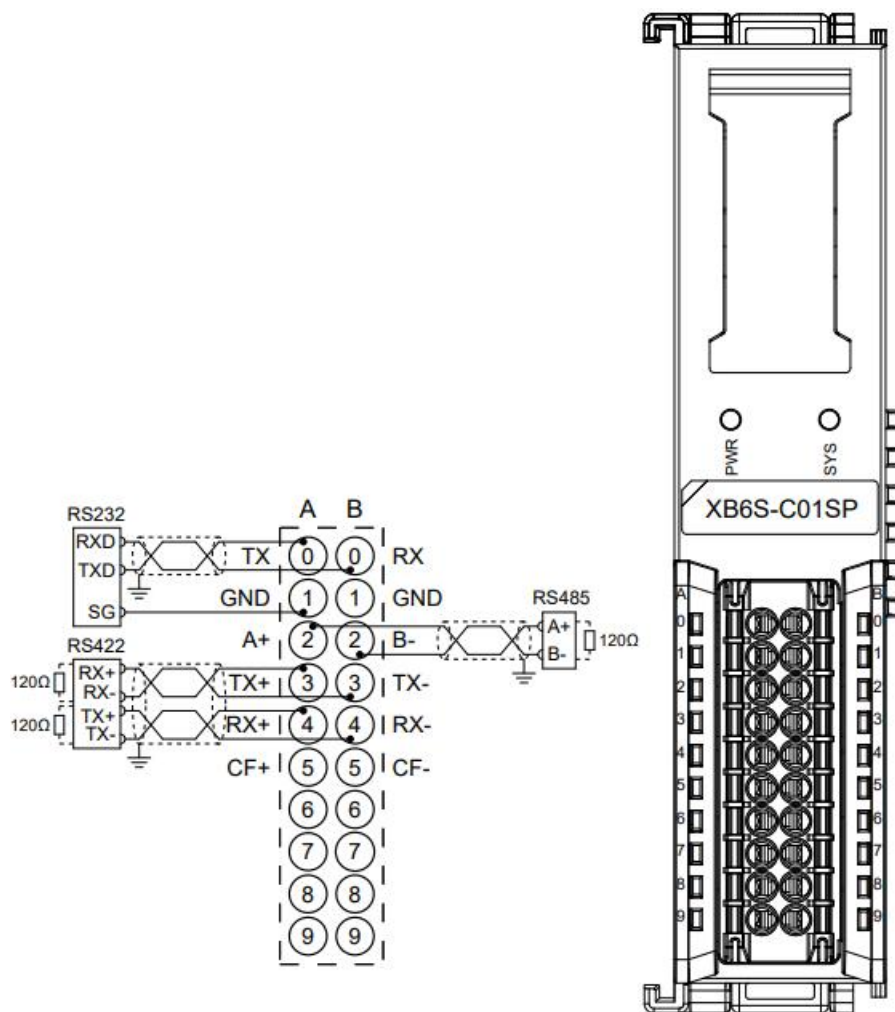
⑫

- Uninstall the module as shown in figure ⑬ below, following the same procedure as for installing the module.



# 5 Wiring

## 5.1 Wiring Diagram



- \*GND is RS232 signal ground, internal conduction
- \*Cables should be shielded twisted pair and reliably grounded
- \*CF+/CF- is reserved for debugging interface of the module, not enabled yet.

## 5.2 Terminal Block Definition

| A                 |                     |                     | B                 |                     |                     |
|-------------------|---------------------|---------------------|-------------------|---------------------|---------------------|
| Terminal Markings | Terminal Definition | Description         | Terminal Markings | Terminal Definition | Description         |
| 0                 | TX                  | RS232 transmitter   | 0                 | RX                  | RS232 receiver      |
| 1                 | GND                 | RS232 signal ground | 1                 | GND                 | RS232 signal ground |
| 2                 | A+                  | RS485 transmit+     | 2                 | B-                  | RS485 transmit-     |
| 3                 | TX+                 | RS422 transmit +    | 3                 | TX-                 | RS422 transmit -    |
| 4                 | RX+                 | RS422 receive+      | 4                 | RX-                 | RS422 receive-      |
| 5                 | CF+                 | Reserve             | 5                 | CF-                 | Reserve             |
| 6                 | NC                  | Empty terminal      | 6                 | NC                  | Empty terminal      |
| 7                 | NC                  | Empty terminal      | 7                 | NC                  | Empty terminal      |
| 8                 | NC                  | Empty terminal      | 8                 | NC                  | Empty terminal      |
| 9                 | NC                  | Empty terminal      | 9                 | NC                  | Empty terminal      |

# 6 Operation

## 6.1 Configuration Parameter Definitions

| Parameter name     | Range of values | Clarification       |
|--------------------|-----------------|---------------------|
| Communication Mode | 0               | NoSelect            |
|                    | 1               | Modbus RTU Master   |
|                    | 2               | Modbus RTU Slave    |
|                    | 3               | Modbus ASCII Master |
|                    | 4               | Modbus ASCII Slave  |
|                    | 5               | FP_Request          |
|                    | 6               | FP_Input_Output     |
|                    | 7               | FP_Input            |
|                    | 8               | FP_Output           |
| Baud               | 0               | 1200 bps            |
|                    | 1               | 2400 bps            |
|                    | 2               | 4800 bps            |
|                    | 3               | 9600 bps            |
|                    | 4               | 19200 bps           |
|                    | 5               | 38400 bps           |
|                    | 6               | 57600 bps           |
|                    | 7               | 115200 bps          |
| Stop Bit           | 0               | 1 Bit               |
|                    | 1               | 2 Bits              |
| Word Format        | 0               | 8 Bits              |
|                    | 1               | 7 Bits              |
| Parity             | 0               | Parity None         |
|                    | 1               | Parity Odd          |
|                    | 2               | Parity Even         |
| Control Mode       | 0               | Disable             |
|                    | 1               | Level               |

|                              |               |   |
|------------------------------|---------------|---|
|                              | 2             | Rising Edge   |
| Node Output Mode             | 0             | Poll  |
|                              | 1             | Change Trigger  |
| Communication Error Behavior | 0             | Clear   |
|                              | 1             | Hold  |
| Respond Timeout              | 5~3000        | Unit ms, default 50   |
| Poll Delay                   | 5~3000        | Unit ms, default 10   |
| Slave ID                     | 1~247         | Valid in slave mode, default 1                                  |
| Slave Respond Delay          | 0~3000        | Unit ms, default 50   |
| Node_1_Param_1               | 0 to FFFFFFFF | Default 0, see <a href="#">Node Code Parameters</a> for details |
| Node_1_Param_2               | 0 to FFFFFFFF | Default 0, see <a href="#">Node Code Parameters</a> for details |
| ...                          | ...           | ...   |
| Node_16_Param_1              | 0 to FFFFFFFF | Default 0, see <a href="#">Node Code Parameters</a> for details |
| Node_16_Param_2              | 0 to FFFFFFFF | Default 0, see <a href="#">Node Code Parameters</a> for details |



## 6.2 Node Code Parameters

### 6.2.1 Modbus Master Function

In Modbus master mode, parameter 1 (Param\_1) is composed of a 2-byte start address parameter and a 2-byte node code parameter, and parameter 2 (Param\_2) consists of a 2-bit byte exchange parameter and a 1-byte slave station number parameter.

| Param_1       |  |   |                      |
|---------------|--|---|----------------------|
| BIT31~BIT16   |  | BIT15~BIT0  |                      |
| Start Address |  | Node Code See <a href="#">Modbus Master Function Node Code</a> for details. |                      |
| Param_2       |  |   |                      |
| BIT31~BIT16   |  | BIT15~BIT10   | BIT9~BIT8            |
| Reserve       |  | See <a href="#">ExByte Flag</a>   | Slave Station Number |

Note: The start address parameter range is 0000~FFFFH and the slave station number parameter range is 1~247.

### 6.2.2 Modbus Master Function Node Code

| SubModule Grouping          | SubModule     | SubModule ID |
|-----------------------------|---------------|--------------|
| Modbus Master Read Coils    | Read 8 bits   | 0x0101       |
|                             | Read 16 bits  | 0x0102       |
|                             | Read 24 bits  | 0x0103       |
|                             | Read 32 bits  | 0x0104       |
|                             | Read 40 bits  | 0x0105       |
|                             | Read 48 bits  | 0x0106       |
|                             | Read 56 bits  | 0x0107       |
|                             | Read 64 bits  | 0x0108       |
|                             | Read 72 bits  | 0x0109       |
|                             | Read 80 bits  | 0x010A       |
|                             | Read 88 bits  | 0x010B       |
|                             | Read 96 bits  | 0x010C       |
|                             | Read 104 bits | 0x010D       |
|                             | Read 112 bits | 0x010E       |
| Read 120 bits               | 0x010F        |              |
| Read 128 bits               | 0x0110        |              |
| Modbus Master Read Discrete | Read 8 bits   | 0x0201       |
|                             | Read 16 bits  | 0x0202       |
|                             | Read 24 bits  | 0x0203       |
|                             | Read 32 bits  | 0x0204       |
|                             | Read 40 bits  | 0x0205       |
|                             | Read 48 bits  | 0x0206       |

|                           |               |        |
|---------------------------|---------------|--------|
|                           | Read 56 bits  | 0x0207 |
|                           | Read 64 bits  | 0x0208 |
|                           | Read 72 bits  | 0x0209 |
|                           | Read 80 bits  | 0x020A |
|                           | Read 88 bits  | 0x020B |
|                           | Read 96 bits  | 0x020C |
|                           | Read 104 bits | 0x020D |
|                           | Read 112 bits | 0x020E |
|                           | Read 120 bits | 0x020F |
|                           | Read 128 bits | 0x0210 |
| Modbus Master Read Hold   | Read 1 Reg    | 0x0301 |
|                           | Read 2 Regs   | 0x0302 |
|                           | Read 3 Regs   | 0x0303 |
|                           | Read 4 Regs   | 0x0304 |
|                           | Read 5 Regs   | 0x0305 |
|                           | Read 6 Regs   | 0x0306 |
|                           | Read 7 Regs   | 0x0307 |
|                           | Read 8 Regs   | 0x0308 |
|                           | Read 9 Regs   | 0x0309 |
|                           | Read 10 Regs  | 0x030A |
|                           | Read 11 Regs  | 0x030B |
|                           | Read 12 Regs  | 0x030C |
|                           | Read 13 Regs  | 0x030D |
|                           | Read 14 Regs  | 0x030E |
|                           | Read 15 Regs  | 0x030F |
|                           | Read 16 Regs  | 0x0310 |
| Modbus Master Read Input  | Read 1 Reg    | 0x0401 |
|                           | Read 2 Regs   | 0x0402 |
|                           | Read 3 Regs   | 0x0403 |
|                           | Read 4 Regs   | 0x0404 |
|                           | Read 5 Regs   | 0x0405 |
|                           | Read 6 Regs   | 0x0406 |
|                           | Read 7 Regs   | 0x0407 |
|                           | Read 8 Regs   | 0x0408 |
|                           | Read 9 Regs   | 0x0409 |
|                           | Read 10 Regs  | 0x040A |
|                           | Read 11 Regs  | 0x040B |
|                           | Read 12 Regs  | 0x040C |
|                           | Read 13 Regs  | 0x040D |
|                           | Read 14 Regs  | 0x040E |
|                           | Read 15 Regs  | 0x040F |
|                           | Read 16 Regs  | 0x0410 |
| Modbus Master Write Coils | Write 8 bits  | 0x0501 |

|                          |                  |        |
|--------------------------|------------------|--------|
|                          | Write 16 bits    | 0x0502 |
|                          | Write 24 bits    | 0x0503 |
|                          | Write 32 bits    | 0x0504 |
|                          | Write 40 bits    | 0x0505 |
|                          | Write 48 bits    | 0x0506 |
|                          | Write 56 bits    | 0x0507 |
|                          | Write 64 bits    | 0x0508 |
|                          | Write 72 bits    | 0x0509 |
|                          | Write 80 bits    | 0x050A |
|                          | Write 88 bits    | 0x050B |
|                          | Write 96 bits    | 0x050C |
|                          | Write 104 bits   | 0x050D |
|                          | Write 112 bits   | 0x050E |
|                          | Write 120 bits   | 0x050F |
|                          | Write 128 bits   | 0x0510 |
|                          | Write Single bit | 0x05FF |
| Modbus Master Write Hold | Write 1 Reg      | 0x0601 |
|                          | Write 2 Regs     | 0x0602 |
|                          | Write 3 Regs     | 0x0603 |
|                          | Write 4 Regs     | 0x0604 |
|                          | Write 5 Regs     | 0x0605 |
|                          | Write 6 Regs     | 0x0606 |
|                          | Write 7 Regs     | 0x0607 |
|                          | Write 8 Regs     | 0x0608 |
|                          | Write 9 Regs     | 0x0609 |
|                          | Write 10 Regs    | 0x060A |
|                          | Write 11 Regs    | 0x060B |
|                          | Write 12 Regs    | 0x060C |
|                          | Write 13 Regs    | 0x060D |
|                          | Write 14 Regs    | 0x060E |
|                          | Write 15 Regs    | 0x060F |
|                          | Write 16 Regs    | 0x0610 |
| Write Single Reg         | 0x06FF           |        |

### 6.2.3 Modbus Slave Function

In Modbus slave mode, parameter 1 (Param\_1) is composed of a 2-byte start address parameter and a 2-byte node code parameter, and parameter 2 (Param\_2) consists of a 2-bit byte swap parameter.

| Param_1       |  |           |
|---------------|--|-----------|
| BIT31~BIT16   | BIT15~BIT0   |           |
| Start Address | Node Code See <a href="#">Modbus Slave Function Node Code</a> for details. |           |
| Param_2       |  |           |
| BIT31~BIT16   | BIT15~BIT2   | BIT1~BIT0 |
| Reserve       | See <a href="#">ExByte Flag</a>  |           |

Note: Starting address parameter range is 0000~FFFFH

### 6.2.4 Modbus Slave Function Node Code

| SubModule Grouping      | SubModule     | SubModule ID |
|-------------------------|---------------|--------------|
| Modbus Slave Read Coils | Read 8 bits   | 0x0701       |
|                         | Read 16 bits  | 0x0702       |
|                         | Read 24 bits  | 0x0703       |
|                         | Read 32 bits  | 0x0704       |
|                         | Read 40 bits  | 0x0705       |
|                         | Read 48 bits  | 0x0706       |
|                         | Read 56 bits  | 0x0707       |
|                         | Read 64 bits  | 0x0708       |
|                         | Read 72 bits  | 0x0709       |
|                         | Read 80 bits  | 0x070A       |
|                         | Read 88 bits  | 0x070B       |
|                         | Read 96 bits  | 0x070C       |
|                         | Read 104 bits | 0x070D       |
|                         | Read 112 bits | 0x070E       |
| Read 120 bits           | 0x070F        |              |
| Read 128 bits           | 0x0710        |              |
| Modbus Slave Read Hold  | Read 1 Reg    | 0x0801       |
|                         | Read 2 Regs   | 0x0802       |
|                         | Read 3 Regs   | 0x0803       |
|                         | Read 4 Regs   | 0x0804       |
|                         | Read 5 Regs   | 0x0805       |
|                         | Read 6 Regs   | 0x0806       |
|                         | Read 7 Regs   | 0x0807       |
|                         | Read 8 Regs   | 0x0808       |
|                         | Read 9 Regs   | 0x0809       |
|                         | Read 10 Regs  | 0x080A       |
|                         | Read 11 Regs  | 0x080B       |

|                             |                |        |
|-----------------------------|----------------|--------|
|                             | Read 12 Regs   | 0x080C |
|                             | Read 13 Regs   | 0x080D |
|                             | Read 14 Regs   | 0x080E |
|                             | Read 15 Regs   | 0x080F |
|                             | Read 16 Regs   | 0x0810 |
| Modbus Slave Write Coils    | Write 8 bits   | 0x0901 |
|                             | Write 16 bits  | 0x0902 |
|                             | Write 24 bits  | 0x0903 |
|                             | Write 32 bits  | 0x0904 |
|                             | Write 40 bits  | 0x0905 |
|                             | Write 48 bits  | 0x0906 |
|                             | Write 56 bits  | 0x0907 |
|                             | Write 64 bits  | 0x0908 |
|                             | Write 72 bits  | 0x0909 |
|                             | Write 80 bits  | 0x090A |
|                             | Write 88 bits  | 0x090B |
|                             | Write 96 bits  | 0x090C |
|                             | Write 104 bits | 0x090D |
|                             | Write 112 bits | 0x090E |
|                             | Write 120 bits | 0x090F |
| Write 128 bits              | 0x0910         |        |
| Modbus Slave Write Discrete | Write 8 bits   | 0x0A01 |
|                             | Write 16 bits  | 0x0A02 |
|                             | Write 24 bits  | 0x0A03 |
|                             | Write 32 bits  | 0x0A04 |
|                             | Write 40 bits  | 0x0A05 |
|                             | Write 48 bits  | 0x0A06 |
|                             | Write 56 bits  | 0x0A07 |
|                             | Write 64 bits  | 0x0A08 |
|                             | Write 72 bits  | 0x0A09 |
|                             | Write 80 bits  | 0x0A0A |
|                             | Write 88 bits  | 0x0A0B |
|                             | Write 96 bits  | 0x0A0C |
|                             | Write 104 bits | 0x0A0D |
|                             | Write 112 bits | 0x0A0E |
|                             | Write 120 bits | 0x0A0F |
| Write 128 bits              | 0x0A10         |        |
| Modbus Slave Write Hold     | Write 1 Reg    | 0x0B01 |
|                             | Write 2 Regs   | 0x0B02 |
|                             | Write 3 Regs   | 0x0B03 |
|                             | Write 4 Regs   | 0x0B04 |
|                             | Write 5 Regs   | 0x0B05 |
|                             | Write 6 Regs   | 0x0B06 |

|                          |               |        |
|--------------------------|---------------|--------|
|                          | Write 7 Regs  | 0x0B07 |
|                          | Write 8 Regs  | 0x0B08 |
|                          | Write 9 Regs  | 0x0B09 |
|                          | Write 10 Regs | 0x0B0A |
|                          | Write 11 Regs | 0x0B0B |
|                          | Write 12 Regs | 0x0B0C |
|                          | Write 13 Regs | 0x0B0D |
|                          | Write 14 Regs | 0x0B0E |
|                          | Write 15 Regs | 0x0B0F |
|                          | Write 16 Regs | 0x0B10 |
| Modbus Slave Write Input | Write 1 Reg   | 0x0C01 |
|                          | Write 2 Regs  | 0x0C02 |
|                          | Write 3 Regs  | 0x0C03 |
|                          | Write 4 Regs  | 0x0C04 |
|                          | Write 5 Regs  | 0x0C05 |
|                          | Write 6 Regs  | 0x0C06 |
|                          | Write 7 Regs  | 0x0C07 |
|                          | Write 8 Regs  | 0x0C08 |
|                          | Write 9 Regs  | 0x0C09 |
|                          | Write 10 Regs | 0x0C0A |
|                          | Write 11 Regs | 0x0C0B |
|                          | Write 12 Regs | 0x0C0C |
|                          | Write 13 Regs | 0x0C0D |
|                          | Write 14 Regs | 0x0C0E |
|                          | Write 15 Regs | 0x0C0F |
|                          | Write 16 Regs | 0x0C10 |

## 6.2.5 ExByte Flag

| Byte Number | function                          | Lengths |
|-------------|-----------------------------------|---------|
| 0           | non-exchange                      | 2 Bits  |
| 1           | 2Bytes Exchange                   | 2 Bits  |
| 2           | 4Bytes Register Exchange          | 2 Bits  |
| 3           | 4Bytes Big-Little-Endian Exchange | 2 Bits  |

## 6.2.6 Freeport function

In Freeport mode, parameter 1 (Param\_1) is a 2-byte node code parameter composition.

| Param_1     |  |
|-------------|--|
| BIT31~BIT16 | BIT15~BIT0   |
| Reserve     | Node Code See <a href="#">Freeport Function Node Code</a> for details. |
| Param_2     |  |
| BIT31~BIT0  |  |
| Reserve     |  |

## 6.2.7 Freeport Function Node Code

| SubModule Grouping | SubModule        | SubModule ID |
|--------------------|------------------|--------------|
| Freeport Input     | Input 1 byte     | 0x0D01       |
|                    | Input 2 bytes    | 0x0D02       |
|                    | Input 4 bytes    | 0x0D03       |
|                    | Input 8 bytes    | 0x0D04       |
|                    | Input 16 bytes   | 0x0D05       |
|                    | Input 32 bytes   | 0x0D06       |
|                    | Input 64 bytes   | 0x0D07       |
|                    | Input 128 bytes  | 0x0D08       |
|                    | Input 160 bytes  | 0x0D09       |
|                    | Input 192 bytes  | 0x0D0A       |
|                    | Input 224 bytes  | 0x0D0B       |
|                    | Input 255 bytes  | 0x0D0C       |
| Freeport Output    | Output 1 byte    | 0x0E01       |
|                    | Output 2 bytes   | 0x0E02       |
|                    | Output 4 bytes   | 0x0E03       |
|                    | Output 8 bytes   | 0x0E04       |
|                    | Output 16 bytes  | 0x0E05       |
|                    | Output 32 bytes  | 0x0E06       |
|                    | Output 64 bytes  | 0x0E07       |
|                    | Output 128 bytes | 0x0E08       |
|                    | Output 160 bytes | 0x0E09       |

|                 |                  |        |
|-----------------|------------------|--------|
|                 | Output 192 bytes | 0x0E0A |
|                 | Output 224 bytes | 0x0E0B |
|                 | Output 255 bytes | 0x0E0C |
| Freeport Parity | CRC8             | 0x0F01 |
|                 | CRC16            | 0x0F02 |
|                 | XOR              | 0x0F03 |
|                 | SUM8             | 0x0F04 |
|                 | LRC              | 0x0F05 |

## 6.2.8 Control and Status Node Codes

When the configuration parameter Control Mode is set to disable, there is no need to configure the control and status node code; when the configuration parameter Control Mode is set to Level or Rising Edge, it is necessary to configure the node code in Node 1 Parameter 1 (Node\_1\_Param\_1) first.

The node code is shown in the following table:

| SubModule Grouping | SubModule             | SubModule ID | Note                           |
|--------------------|-----------------------|--------------|--------------------------------|
| Control&Status     | 1 Node                | 0x0001       | Configuration in Master Mode   |
|                    | 2 Node                | 0x0002       |                                |
|                    | 3 Node                | 0x0003       |                                |
|                    | 4 Node                | 0x0004       |                                |
|                    | 5 Node                | 0x0005       |                                |
|                    | 6 Node                | 0x0006       |                                |
|                    | 7 Node                | 0x0007       |                                |
|                    | 8 Node                | 0x0008       |                                |
|                    | 9 Node                | 0x0009       |                                |
|                    | 10 Node               | 0x000A       |                                |
|                    | 11 Node               | 0x000B       |                                |
|                    | 12 Node               | 0x000C       |                                |
|                    | 13 Node               | 0x000D       |                                |
|                    | 14 Node               | 0x000E       |                                |
|                    | 15 Node               | 0x000F       |                                |
|                    | Freeport Req          | 0x0020       | Configuration in Freeport Mode |
|                    | Freeport Input_Output | 0x0021       |                                |
|                    | Freeport Input        | 0x0022       |                                |
|                    | Freeport Output       | 0x0023       | Configuration in Slave Mode    |
|                    | Modbus Slave          | 0x0030       |                                |



## 6.3 Process data

The module factory defaults to XB6S-C01SP-32Bytes, the length of upstream and downstream data is 32Bytes each, when the amount of PDO data generated by the communication node parameter configuration of the module reaches a certain range (the amount of upstream data or the amount of downstream data either reaches the range limit), the coupler module will alarm 0x5002, then re-power up and scan the module, the module will switch to the corresponding PDO data length automatically.

Module PDO length range and model match table:

| Module PDO Length Range and Model Matching Table |                              |                                |
|--|------------------------------|--------------------------------|
| Module Model                                     | Upstream Data Length (Bytes) | Downstream Data Length (Bytes) |
| XB6S-C01SP-32Bytes                               | 0~32                         | 0~32                           |
| XB6S-C01SP-64Bytes                               | 33~64                        | 33~64                          |
| XB6S-C01SP-128Bytes                              | 65~128                       | 65~128                         |
| XB6S-C01SP-255Bytes                              | 129~255                      | 129~255                        |

The following is an example of the configuration parameters for model XB6S-C01SP-32Bytes.

- **When the configuration parameter Control Mode is set to disable, the upstream and downstream process data is shown in the following table.**

| Upstream Data (32 Bytes)   |             |   |               |
|----------------------------|-------------|---|---------------|
| Byte Number                | Function    | Meaning                                 | Address Range |
| 1                          | Input data  | Serial communication module input data  | 32 bytes      |
| 2                          | Input data  |   |               |
| ...                        | ...         |   |               |
| 32                         | Input data  |   |               |
| Downstream data (32 bytes) |             |   |               |
| Byte Number                | Function    | Meaning                                 | Address Range |
| 1                          | Output data | Serial communication module output data | 32 bytes      |
| 2                          | Output data |   |               |
| ...                        | ...         |   |               |
| 32                         | Output data |   |               |

- When the configuration parameter Control Mode is set to Level or Rising Edge, the upstream and downstream process data is shown in the table below.

### 6.3.1 Modbus Master Process Data

| Upstream Data (32 Bytes)   |              |  |               |
|----------------------------|--------------|--|---------------|
| Byte Number                | Function     | Meaning  | Address Range |
| 1                          | Status word  | Bit0: Idle state   | 1 byte        |
|                            |              | Bit1: Busy state, i.e., in data transmission                                 |               |
|                            |              | Bit2: Completion status, i.e., data transmission completed                   |               |
| 2                          | Alarm code   | See <a href="#">Alarm Code</a>   | 1 byte        |
| 3                          | Input data   | Serial communication module input data                                       | Last 30 bytes |
| 4                          | Input data   |  |               |
| ...                        | ...          |  |               |
| 32                         | Input data   |  |               |
| Downstream Data (32 Bytes) |              |  |               |
| Byte Number                | Function     | Meaning  | Address Range |
| 1                          | Control word | When the configuration parameter Control Mode is Level, set 1 to send data.  | 1 byte        |
|                            |              | When the configuration parameter Control Mode is Rising Edge, 0->1 send data |               |
| 2                          | Output data  | Serial communication module output data                                      | Last 31 bytes |
| 3                          | Output data  |  |               |
| ...                        | ...          |  |               |
| 32                         | Output data  |  |               |

**Note:** The process data in the above table is the configuration of one node in Level or Rising Edge mode. If two nodes are configured, byte numbers 1 and 2 of the upstream data are the status bits and alarm code of node 1, byte numbers 3 and 4 are the status bits and alarm code of node 2, and the remaining bytes are the data bits; byte number 1 of the downstream data is the control bit of node 1, byte number 2 is the control bit of node 2, and the remaining bytes are the data bits; the maximum number of nodes that can be configured is 15.

## 6.3.2 Modbus Slave Process Data

| Upstream Data (32 Bytes)   |              |  |               |
|----------------------------|--------------|--|---------------|
| Byte Number                | Function     | Meaning  | Address Range |
| 1                          | Status word  | Bit0: Idle state   | 1 byte        |
|                            |              | Bit1: Busy state, i.e., in data transmission               |               |
|                            |              | Bit2: Completion status, i.e., data transmission completed |               |
| 2                          | Alarm code   | See <a href="#">Alarm Code</a>                             | 1 byte        |
| 3                          | Input data   | Serial communication module input data                     | Last 30 bytes |
| 4                          | Input data   |  |               |
| ...                        | ...          |  |               |
| 32                         | Input data   |  |               |
| Downstream Data (32 Bytes) |              |  |               |
| Byte Number                | Function     | Meaning  | Address Range |
| 1                          | Control word | Bit0: Alarm code clear                                     | 1 byte        |
| 2                          | Output data  | Serial communication module output data                    | Last 31 bytes |
| 3                          | Output data  |  |               |
| ...                        | ...          |  |               |
| 32                         | Output data  |  |               |

## 6.3.3 Freeport\_Request Process Data

| Upstream Data (32 Bytes)   |                         |   |               |
|----------------------------|-------------------------|---|---------------|
| Byte Number                | Function                | Meaning   | Address Range |
| 1                          | Status word             | Bit0: Idle state  | 1 byte        |
|                            |                         | Bit1: Busy state, i.e., in data transmission              |               |
|                            |                         | Bit2: Completion State, i.e., data transmission completed |               |
| 2                          | Alarm code              | Bit0: Receive Data Length Error                           | 1 byte        |
|                            |                         | Bit1: Checksum Error                                      |               |
|                            |                         | Bit2: Timeout error                                       |               |
| 3                          | Receive data length     | 0~255   | 1 byte        |
| 4~5                        | Receive exception count | 0~65535   | 2 bytes       |
| 6~7                        | Receive data count      | 0~65535   | 2 bytes       |
| 8                          | Input data              | Serial communication module input data                    | Last 25 bytes |
| 9                          | Input data              |   |               |
| ...                        | ...                     |   |               |
| 32                         | Input data              |   |               |
| Downstream Data (32 Bytes) |                         |   |               |
| Byte Number                | Function                | Meaning   | Address Range |
| 1                          | Control word            | Bit0: Enable  | 1 byte        |
|                            |                         | Bit1: Status word clear                                   |               |
|                            |                         | Bit2: Receive data clear                                  |               |
| 2                          | Send data length        | 0~255   | 1 byte        |
| 3                          | Clear count             | Bit0: receive exception count clear                       | 1 byte        |
|                            |                         | Bit1: Receive data count clear                            |               |
|                            |                         | Bit2: Receive data length clear                           |               |
| 4                          | Error code clear        | Bit0: Length exception clear                              | 1 byte        |
|                            |                         | Bit1: Checksum Error Clear                                |               |
|                            |                         | Bit2: Timeout Error Clear                                 |               |
| 5                          | Output data             | Serial communication module output data                   | Last 28 bytes |
| 6                          | Output data             |   |               |
| ...                        | ...                     |   |               |
| 32                         | Output data             |   |               |

## 6.3.4 Freeport\_Input Process Data

| Upstream Data (32 Bytes)   |                         |   |               |
|----------------------------|-------------------------|---|---------------|
| Byte Number                | Function                | Meaning                                 | Address Range |
| 1                          | Alarm code              | Bit0: Receive data length error         | 1 byte        |
|                            |                         | Bit1: Checksum Error                    |               |
| 2                          | Receive data length     | 0~255                                   | 1 byte        |
| 3~4                        | Receive exception count | 0~65535                                 | 2 bytes       |
| 5~6                        | Receive data count      | 0~65535                                 | 2 bytes       |
| 7                          | Input data              | Serial communication module input data  | Last 26 bytes |
| 8                          | Input data              |   |               |
| ...                        | ...                     |   |               |
| 32                         | Input data              |   |               |
| Downstream Data (32 Bytes) |                         |   |               |
| Byte Number                | Function                | Meaning                                 | Address Range |
| 1                          | Control word            | Bit0: Enable                            | 1 byte        |
|                            |                         | Bit1: Status word clear                 |               |
|                            |                         | Bit2: Receive data clear                |               |
| 2                          | Clear count             | Bit0: Receive exception count clear     | 1 byte        |
|                            |                         | Bit1: Receive data count clear          |               |
|                            |                         | Bit2: Receive data length clear         |               |
| 3                          | Error code clear        | Bit0: Receive length error clear        | 1 byte        |
|                            |                         | Bit1: Checksum Error Clear              |               |
| 4                          | Output data             | Serial communication module output data | Last 29 bytes |
| 5                          | Output data             |   |               |
| ...                        | ...                     |   |               |
| 32                         | Output data             |   |               |

### 6.3.5 Freeport\_Output Process Data

| <b>Downstream Data (32 Bytes)</b> |                  |   |                      |
|-----------------------------------|------------------|---|----------------------|
| <b>Byte Number</b>                | <b>Function</b>  | <b>Meaning</b>                          | <b>Address Range</b> |
| 1                                 | Control word     | Bit0: Enable                            | 1 byte               |
| 2                                 | Send data length | 0~255                                   | 1 byte               |
| 3                                 | Output data      | Serial communication module output data | Last 30 bytes        |
| 4                                 | Output data      |   |                      |
| ...                               | ...              |   |                      |
| 32                                | Output data      |   |                      |

## 6.3.6 Freeport\_Input\_Output Process Data

| Upstream Data (32 Bytes)   |                         |   |               |
|----------------------------|-------------------------|---|---------------|
| Byte Number                | Function                | Meaning                                 | Address Range |
| 1                          | Alarm code              | Bit0: Receive data length error         | 1 byte        |
|                            |                         | Bit1: Checksum Error                    |               |
| 2                          | Receive data length     | 0~255                                   | 1 byte        |
| 3~4                        | Receive exception count | 0~65535                                 | 2 bytes       |
| 5~6                        | Receive data count      | 0~65535                                 | 2 bytes       |
| 7                          | Input data              | Serial communication module input data  | Last 26 bytes |
| 8                          | Input data              |   |               |
| ...                        | ...                     |   |               |
| 32                         | Input data              |   |               |
| Downstream Data (32 Bytes) |                         |   |               |
| Byte Number                | Function                | Meaning                                 | Address Range |
| 1                          | Control word            | Bit0: Enable                            | 1 byte        |
|                            |                         | Bit1: Status word clear                 |               |
|                            |                         | Bit2: Receive data clear                |               |
| 2                          | Send data length        | 0~255                                   | 1 byte        |
| 3                          | Clear count             | Bit0: Receive exception count clear     | 1 byte        |
|                            |                         | Bit1: Receive data count clear          |               |
|                            |                         | Bit2: Receive data length clear         |               |
| 4                          | Error code clear        | Bit0: Receive length error clear        | 1 byte        |
|                            |                         | Bit1: Checksum Error Clear              |               |
| 5                          | Output data             | Serial communication module output data | Last 28 bytes |
| 6                          | Output data             |   |               |
| ...                        | ...                     |   |               |
| 32                         | Output data             |   |               |

### 6.3.7 Modbus Alarm Code

| <b>Error Code</b> | <b>Comment</b>              |
|-------------------|-----------------------------|
| 0x00              | No error                    |
| 0x01              | Slave device address error  |
| 0x02              | Function code error         |
| 0x03              | CRC checksum error          |
| 0x04              | Response bytes error        |
| 0x05              | Coil register data error    |
| 0x06              | Holding register data error |
| 0x07              | Illegal data address        |
| 0x08              | Slave device failure        |
| 0x09              | LRC checksum error          |
| 0x0C              | Data length error           |
| 0x0D              | Data conversion error       |
| 0x0E              | ASCII frame error           |
| 0x0F              | Response timeout            |



## 6.4 Module Configuration Description

### 6.4.1 Application in TwinCAT3 software environment

#### 1、Preliminary

- **Hardware Environment**

- **Module Model XB6S-C01SP**
- **EtherCAT Bus Coupler Module, End Cap**  
This description takes the XB6S-EC2002 coupler module as an example
- **A computer with pre-installed TwinCAT3 software**
- **Shielded cables for EtherCAT**
- **One switching power supply**
- **Module installation rails and rail mounts**
- **Device Configuration Files**

Configuration file access: <https://www.solidotech.com/documents/configfile>

- **Hardware configuration and wiring**

Follow "[4 Installation and uninstall](#)" and "[5 Wiring](#)".

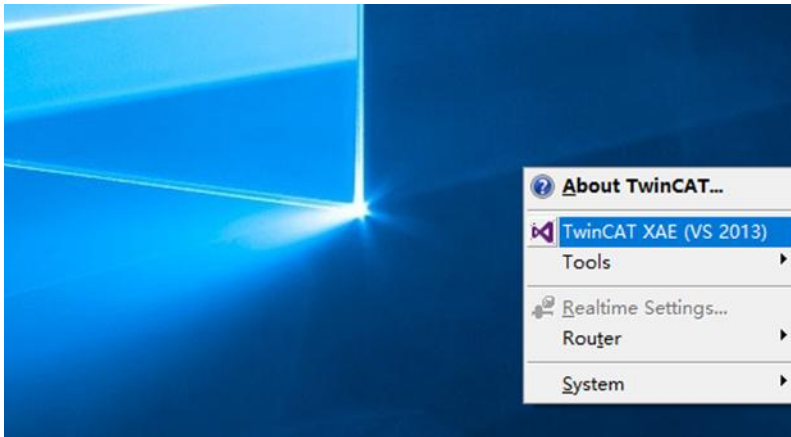
#### 2、Preset Profiles

Place the ESI configuration file (EcatTerminal-XB6S\_V1.19.13\_ENUM.xml) in the TwinCAT installation directory under "C:\TwinCAT\3.1\Config\Io\EtherCAT" as shown below.

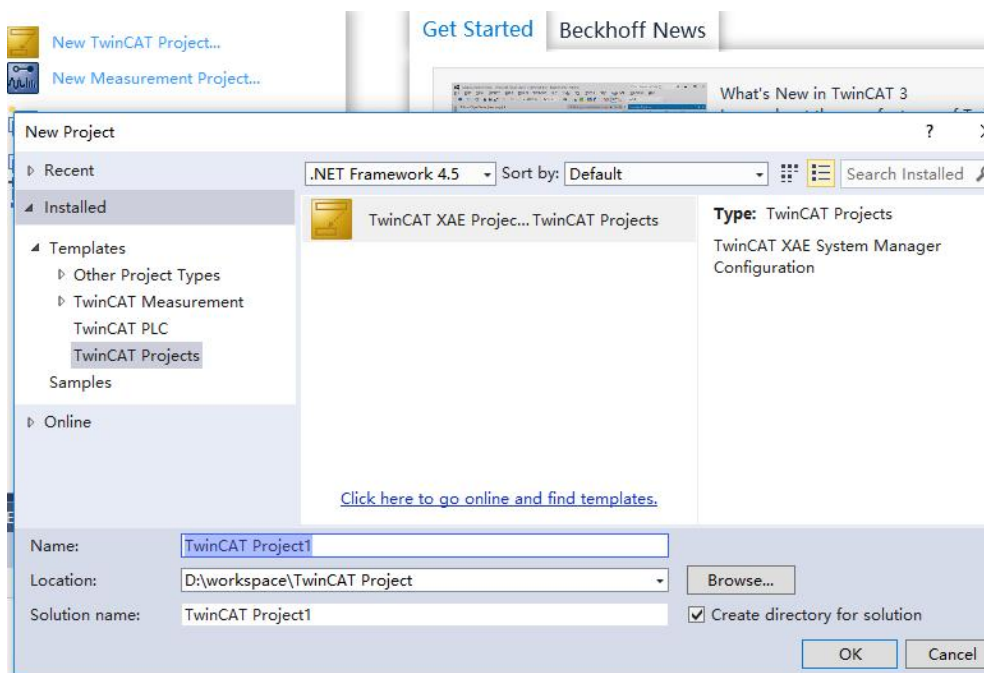
| 名称                                  | 修改日期             | 类型     | 大小        |
|-------------------------------------|------------------|--------|-----------|
| Beckhoff EL32xx.xml                 | 2017/10/25 15:43 | XML 文档 | 5,997 KB  |
| Beckhoff EL66xx.xml                 | 2017/10/27 8:55  | XML 文档 | 1,820 KB  |
| Beckhoff EKx9xx.xml                 | 2017/11/3 9:53   | XML 文档 | 1,223 KB  |
| Beckhoff EP7xx.xml                  | 2017/11/8 9:46   | XML 文档 | 9,290 KB  |
| Beckhoff ATH2xx.xml                 | 2017/11/23 13:22 | XML 文档 | 439 KB    |
| Beckhoff EPP3xx.xml                 | 2017/12/8 8:48   | XML 文档 | 2,099 KB  |
| Beckhoff EPP1xx.xml                 | 2017/12/14 11:34 | XML 文档 | 480 KB    |
| Beckhoff EL34xx.xml                 | 2017/12/15 15:35 | XML 文档 | 5,634 KB  |
| Beckhoff EK13xx.xml                 | 2017/12/19 14:30 | XML 文档 | 16 KB     |
| Beckhoff EPP2xx.xml                 | 2017/12/28 12:22 | XML 文档 | 1,811 KB  |
| Beckhoff EJ1xx.xml                  | 2018/1/4 10:00   | XML 文档 | 67 KB     |
| Beckhoff EJ3xx.xml                  | 2018/1/4 10:07   | XML 文档 | 1,169 KB  |
| Beckhoff EJ7xx.xml                  | 2018/1/4 10:11   | XML 文档 | 2,339 KB  |
| Beckhoff EJ9xx.xml                  | 2018/1/4 10:23   | XML 文档 | 160 KB    |
| Beckhoff EJ6xx.xml                  | 2018/1/4 10:31   | XML 文档 | 313 KB    |
| Beckhoff EL30xx.xml                 | 2018/1/11 13:03  | XML 文档 | 11,508 KB |
| Beckhoff EL37xx.xml                 | 2018/1/23 13:59  | XML 文档 | 11,837 KB |
| Beckhoff EJ2xx.xml                  | 2018/1/23 14:21  | XML 文档 | 239 KB    |
| Beckhoff EL5xx.xml                  | 2018/1/23 15:11  | XML 文档 | 6,307 KB  |
| Beckhoff EJ5xx.xml                  | 2018/1/23 15:12  | XML 文档 | 218 KB    |
| Beckhoff EL2xx.xml                  | 2018/1/24 9:40   | XML 文档 | 2,868 KB  |
| Beckhoff EL33xx.xml                 | 2018/1/26 9:34   | XML 文档 | 6,727 KB  |
| Beckhoff ELM3xx.xml                 | 2018/2/1 10:19   | XML 文档 | 14,238 KB |
| Beckhoff AX5xx.xml                  | 2018/2/8 16:15   | XML 文档 | 930 KB    |
| Beckhoff EL1xx.xml                  | 2018/2/19 17:15  | XML 文档 | 3,387 KB  |
| Beckhoff EL25xx.xml                 | 2018/2/21 10:23  | XML 文档 | 6,543 KB  |
| EcatTerminal-XB6S_V1.19.13_ENUM.xml | 2024/6/25 10:15  | XML 文档 | 1,821 KB  |

### 3. Create Project

- a. Click the TwinCAT icon in the lower right corner of the desktop and select "TwinCAT XAE (VS xxxx)" to open the TwinCAT software, as shown below.

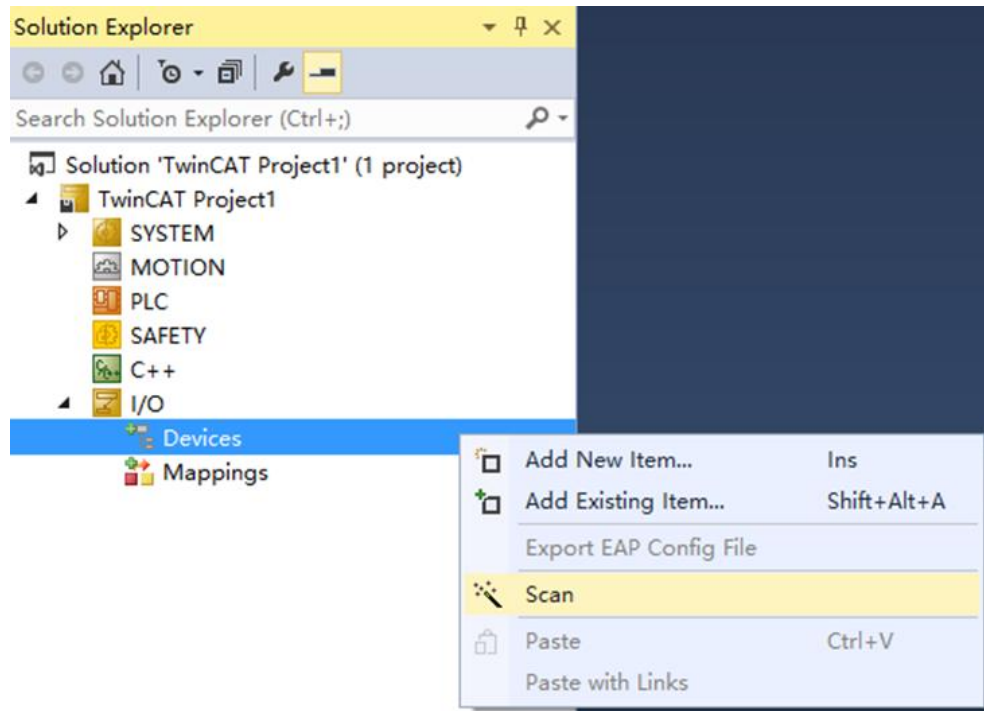


- b. Click "New TwinCAT Project", in the pop-up window, "Name" and "Solution name" correspond to the project name and solution name respectively, and "Location" corresponds to the project path, and these three items can be selected by default, then click "OK", the project is created successfully, as shown in the following figure.



#### 4. Scanning device

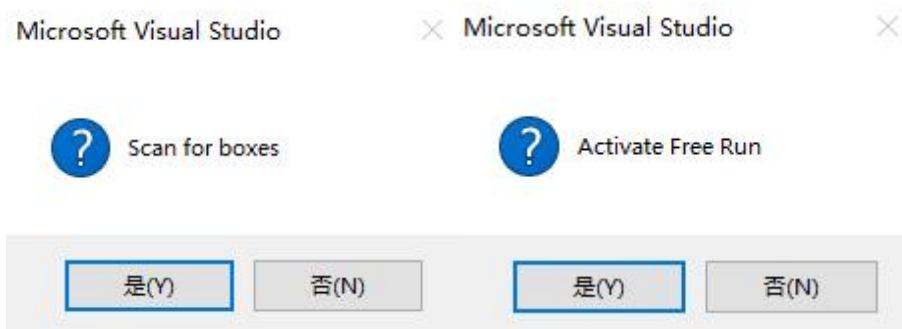
- a. After creating the project, right-click on the "Scan" option under "I/O -> Devices" to perform a slave device scan, as shown in the following figure.



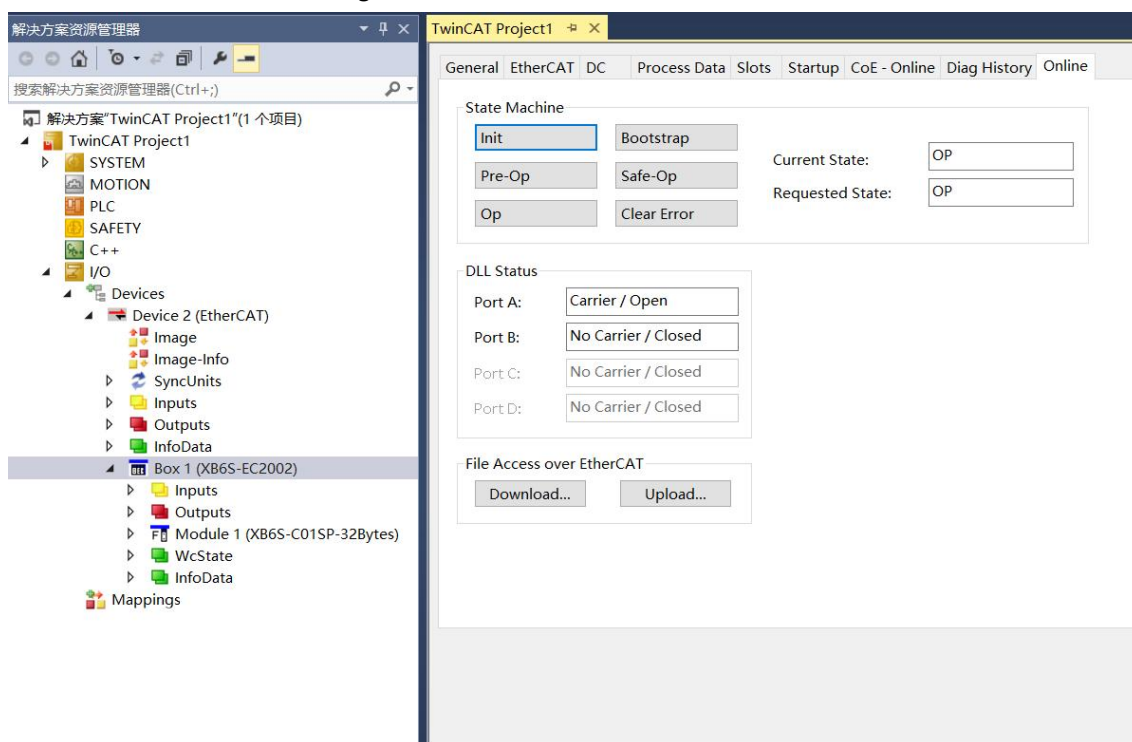
- b. Check the "Local Connection" box, as shown in the following figure.



- c. Click on the pop-up window "Scan for boxes" and select "Yes"; click on the pop-up window "Activate Free Run" and select "Yes", as shown in the figure below.

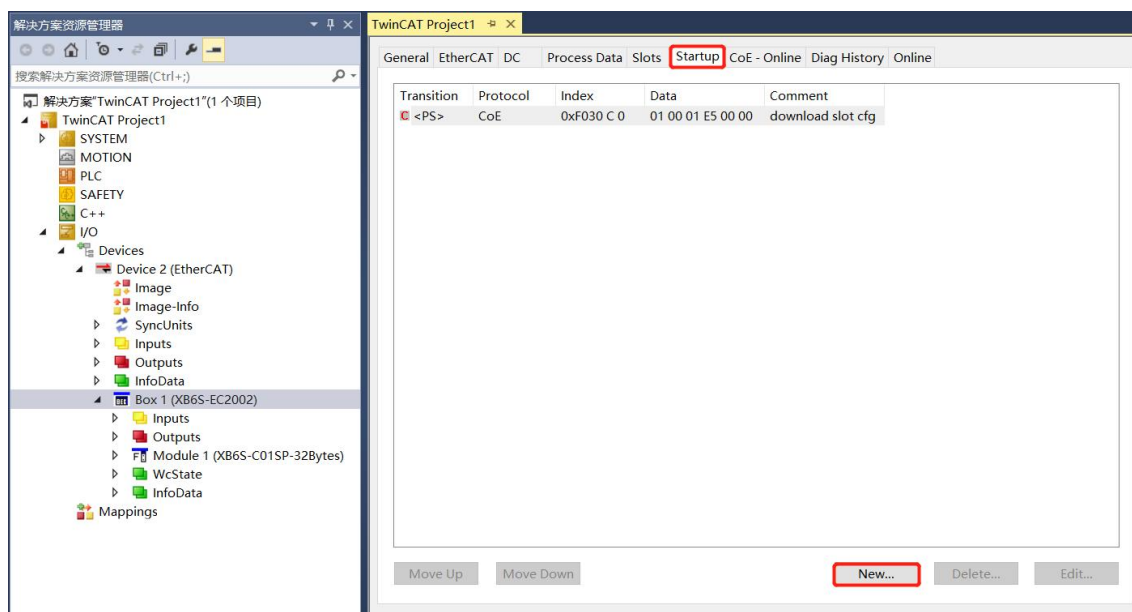


- d. After scanning to the device, you can see Box1 (XB6S-EC2002) and Module1 (XB6S-C01SP-32Bytes) in the left navigation tree, and you can see TwinCAT in the "OP" status at "Online". At "Online", you can see that TwinCAT is in the "OP" state, and you can observe that the RUN indicator of the slave device is on, as shown in the figure below.

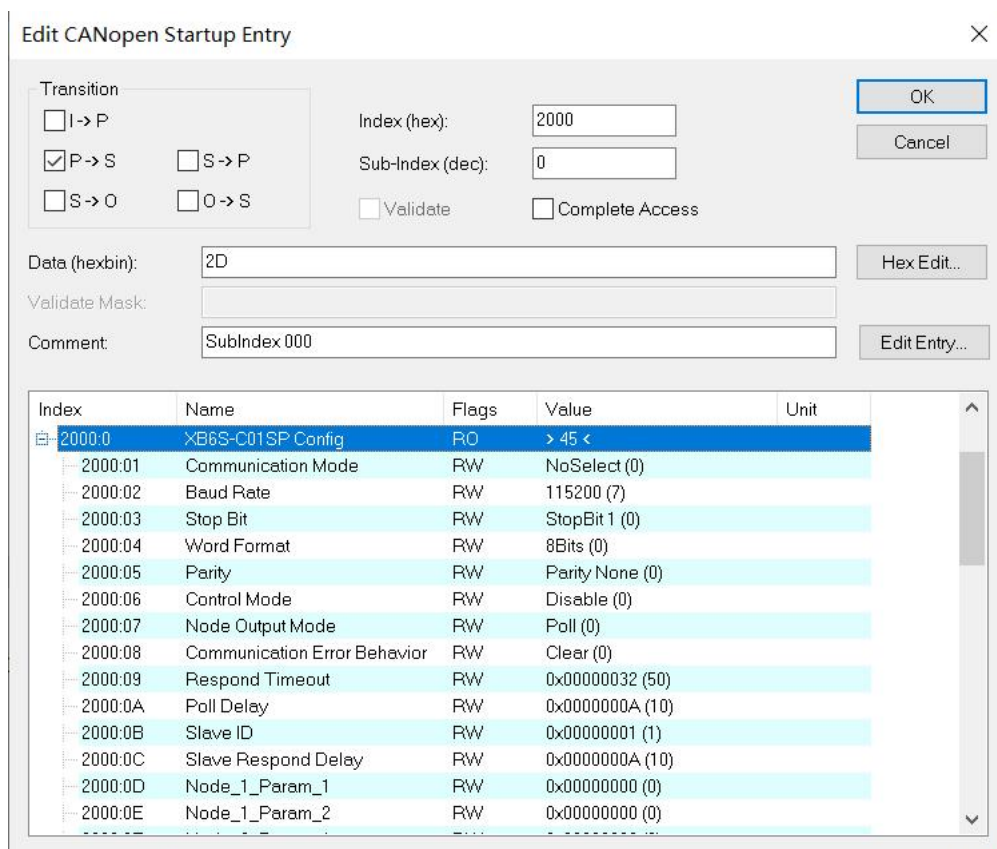


### 5. Validating Basic Functions

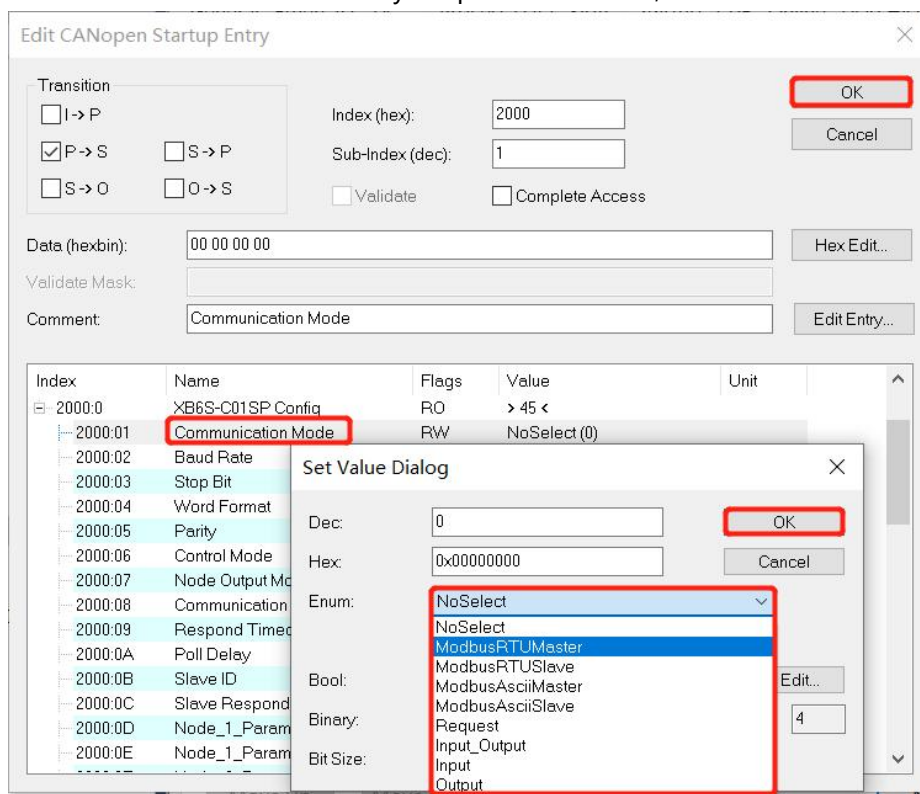
- a. Click "Box1 -> Startup -> New" in the left navigation tree to enter the configuration parameter editing page, as shown in the following figure.



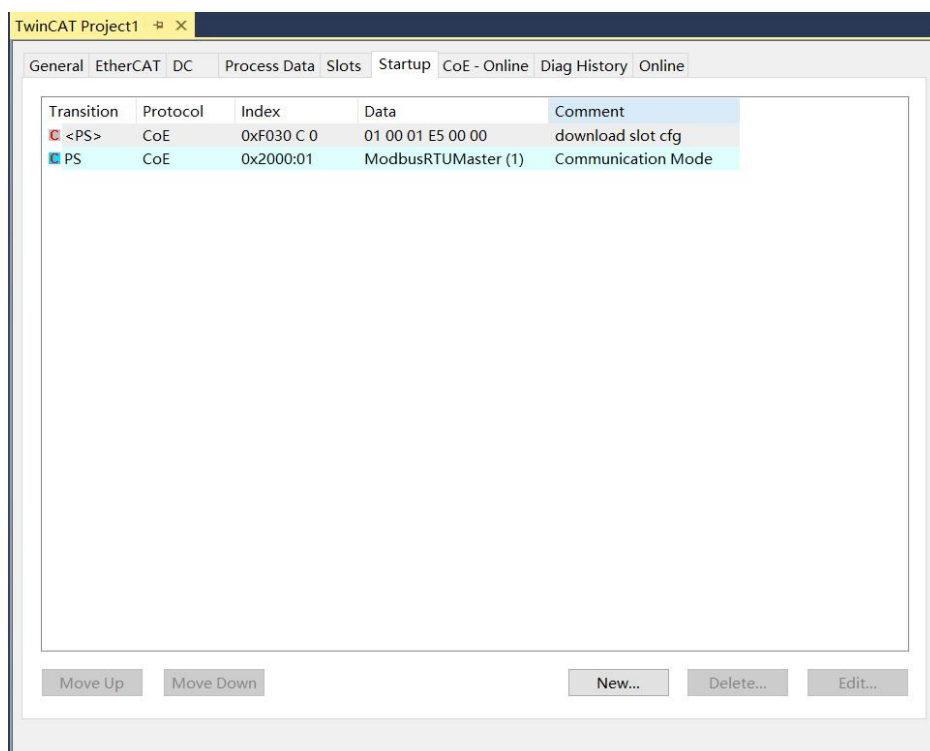
- b. In the Edit CANopen Startup Entry pop-up window, click the "+" in front of Index 2000:0 to expand the Configuration Parameters menu, you can see 44 configuration parameters, click on any one of the parameters, you can set the relevant configuration, as shown in the following figure.



- c. For example, to modify the communication mode parameters, you can double-click "Communication Mode" to modify the parameter values, as shown in the following figure.



- d. After the parameter modification is completed, you can see the modified parameter items and parameter values under Startup, as shown in the following figure. After the parameter setting is completed, it is necessary to carry out Reload operation and re-power up the module to realize that the master station automatically sends down the parameter setting.



- e. The left navigation tree "Module 1 -> Inputs" displays the upstream data of the communication module, which is used to monitor the status of the module, as shown in the figure below.

| Name  | Online | Type  | Size | >Address | In/Out | User ID | Linked to |
|-------|--------|-------|------|----------|--------|---------|-----------|
| RX 1  | 0      | USINT | 1.0  | 41.0     | Input  | 0       |           |
| RX 2  | 0      | USINT | 1.0  | 42.0     | Input  | 0       |           |
| RX 3  | 0      | USINT | 1.0  | 43.0     | Input  | 0       |           |
| RX 4  | 0      | USINT | 1.0  | 44.0     | Input  | 0       |           |
| RX 5  | 0      | USINT | 1.0  | 45.0     | Input  | 0       |           |
| RX 6  | 0      | USINT | 1.0  | 46.0     | Input  | 0       |           |
| RX 7  | 0      | USINT | 1.0  | 47.0     | Input  | 0       |           |
| RX 8  | 0      | USINT | 1.0  | 48.0     | Input  | 0       |           |
| RX 9  | 0      | USINT | 1.0  | 49.0     | Input  | 0       |           |
| RX 10 | 0      | USINT | 1.0  | 50.0     | Input  | 0       |           |
| RX 11 | 0      | USINT | 1.0  | 51.0     | Input  | 0       |           |
| RX 12 | 0      | USINT | 1.0  | 52.0     | Input  | 0       |           |
| RX 13 | 0      | USINT | 1.0  | 53.0     | Input  | 0       |           |
| RX 14 | 0      | USINT | 1.0  | 54.0     | Input  | 0       |           |
| RX 15 | 0      | USINT | 1.0  | 55.0     | Input  | 0       |           |
| RX 16 | 0      | USINT | 1.0  | 56.0     | Input  | 0       |           |
| RX 17 | 0      | USINT | 1.0  | 57.0     | Input  | 0       |           |
| RX 18 | 0      | USINT | 1.0  | 58.0     | Input  | 0       |           |
| RX 19 | 0      | USINT | 1.0  | 59.0     | Input  | 0       |           |
| RX 20 | 0      | USINT | 1.0  | 60.0     | Input  | 0       |           |
| RX 21 | 0      | USINT | 1.0  | 61.0     | Input  | 0       |           |
| RX 22 | 0      | USINT | 1.0  | 62.0     | Input  | 0       |           |
| RX 23 | 0      | USINT | 1.0  | 63.0     | Input  | 0       |           |
| RX 24 | 0      | USINT | 1.0  | 64.0     | Input  | 0       |           |
| RX 25 | 0      | USINT | 1.0  | 65.0     | Input  | 0       |           |
| RX 26 | 0      | USINT | 1.0  | 66.0     | Input  | 0       |           |
| RX 27 | 0      | USINT | 1.0  | 67.0     | Input  | 0       |           |
| RX 28 | 0      | USINT | 1.0  | 68.0     | Input  | 0       |           |
| RX 29 | 0      | USINT | 1.0  | 69.0     | Input  | 0       |           |
| RX 30 | 0      | USINT | 1.0  | 70.0     | Input  | 0       |           |
| RX 31 | 0      | USINT | 1.0  | 71.0     | Input  | 0       |           |
| RX 32 | 0      | USINT | 1.0  | 72.0     | Input  | 0       |           |

- f. The left navigation tree "Module 1 -> Outputs" displays the downstream data of the communication module. is used for to monitor the output status of the module, as shown in the figure below.

| Name  | Online | Type  | Size | >Address | In/Out | User ID | Linked to |
|-------|--------|-------|------|----------|--------|---------|-----------|
| TX 1  | 0      | USINT | 1.0  | 41.0     | Output | 0       |           |
| TX 2  | 0      | USINT | 1.0  | 42.0     | Output | 0       |           |
| TX 3  | 0      | USINT | 1.0  | 43.0     | Output | 0       |           |
| TX 4  | 0      | USINT | 1.0  | 44.0     | Output | 0       |           |
| TX 5  | 0      | USINT | 1.0  | 45.0     | Output | 0       |           |
| TX 6  | 0      | USINT | 1.0  | 46.0     | Output | 0       |           |
| TX 7  | 0      | USINT | 1.0  | 47.0     | Output | 0       |           |
| TX 8  | 0      | USINT | 1.0  | 48.0     | Output | 0       |           |
| TX 9  | 0      | USINT | 1.0  | 49.0     | Output | 0       |           |
| TX 10 | 0      | USINT | 1.0  | 50.0     | Output | 0       |           |
| TX 11 | 0      | USINT | 1.0  | 51.0     | Output | 0       |           |
| TX 12 | 0      | USINT | 1.0  | 52.0     | Output | 0       |           |
| TX 13 | 0      | USINT | 1.0  | 53.0     | Output | 0       |           |
| TX 14 | 0      | USINT | 1.0  | 54.0     | Output | 0       |           |
| TX 15 | 0      | USINT | 1.0  | 55.0     | Output | 0       |           |
| TX 16 | 0      | USINT | 1.0  | 56.0     | Output | 0       |           |
| TX 17 | 0      | USINT | 1.0  | 57.0     | Output | 0       |           |
| TX 18 | 0      | USINT | 1.0  | 58.0     | Output | 0       |           |
| TX 19 | 0      | USINT | 1.0  | 59.0     | Output | 0       |           |
| TX 20 | 0      | USINT | 1.0  | 60.0     | Output | 0       |           |
| TX 21 | 0      | USINT | 1.0  | 61.0     | Output | 0       |           |
| TX 22 | 0      | USINT | 1.0  | 62.0     | Output | 0       |           |
| TX 23 | 0      | USINT | 1.0  | 63.0     | Output | 0       |           |
| TX 24 | 0      | USINT | 1.0  | 64.0     | Output | 0       |           |
| TX 25 | 0      | USINT | 1.0  | 65.0     | Output | 0       |           |
| TX 26 | 0      | USINT | 1.0  | 66.0     | Output | 0       |           |
| TX 27 | 0      | USINT | 1.0  | 67.0     | Output | 0       |           |
| TX 28 | 0      | USINT | 1.0  | 68.0     | Output | 0       |           |
| TX 29 | 0      | USINT | 1.0  | 69.0     | Output | 0       |           |
| TX 30 | 0      | USINT | 1.0  | 70.0     | Output | 0       |           |
| TX 31 | 0      | USINT | 1.0  | 71.0     | Output | 0       |           |
| TX 32 | 0      | USINT | 1.0  | 72.0     | Output | 0       |           |

## 6. RTU Master Mode Function Example

**Example 1: Verify that the module RTU master writes the values of the 2 holding registers in disable mode with a tool or device such as Modbus Slave debugging software.**

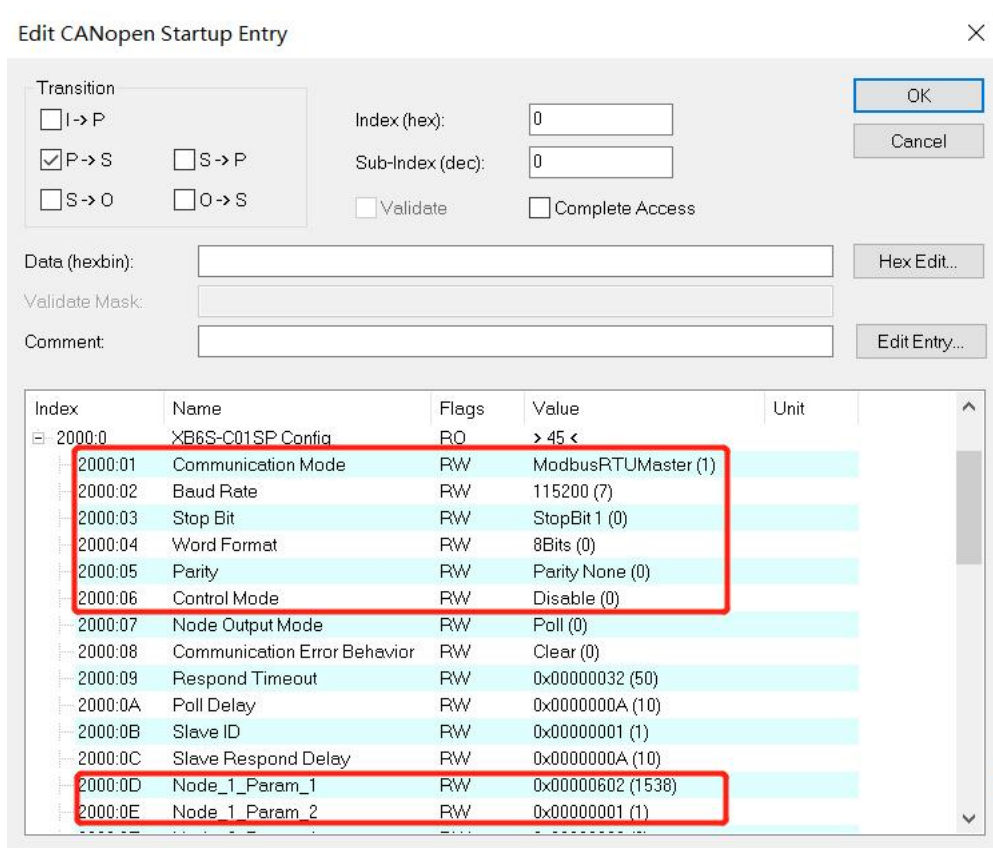
- a. Configure the configuration parameters and select 1 for communication mode i.e. RTU master mode as shown below.

Communication Mode: Select ModbusRTUMaster;

Control Mode: Select Disable;

Node\_1\_Param\_1: Configure 0x00000602, see [6.2.1 Modbus Master Functionality](#) for configuration parameters;

Node\_1\_Param\_2: Configure 0x00000001, see [6.2.1 Modbus Master Functionality](#) for configuration parameters.





b. In the downstream data, the values of the 2 holding registers are written as shown below.

| Name  | Online | Type  | Size | >Address | In/Out | User ID | Linked to |
|-------|--------|-------|------|----------|--------|---------|-----------|
| TX 1  | 1      | USINT | 1.0  | 41.0     | Output | 0       |           |
| TX 2  | 1      | USINT | 1.0  | 42.0     | Output | 0       |           |
| TX 3  | 1      | USINT | 1.0  | 43.0     | Output | 0       |           |
| TX 4  | 1      | USINT | 1.0  | 44.0     | Output | 0       |           |
| TX 5  | 0      | USINT | 1.0  | 45.0     | Output | 0       |           |
| TX 6  | 0      | USINT | 1.0  | 46.0     | Output | 0       |           |
| TX 7  | 0      | USINT | 1.0  | 47.0     | Output | 0       |           |
| TX 8  | 0      | USINT | 1.0  | 48.0     | Output | 0       |           |
| TX 9  | 0      | USINT | 1.0  | 49.0     | Output | 0       |           |
| TX 10 | 0      | USINT | 1.0  | 50.0     | Output | 0       |           |
| TX 11 | 0      | USINT | 1.0  | 51.0     | Output | 0       |           |
| TX 12 | 0      | USINT | 1.0  | 52.0     | Output | 0       |           |
| TX 13 | 0      | USINT | 1.0  | 53.0     | Output | 0       |           |
| TX 14 | 0      | USINT | 1.0  | 54.0     | Output | 0       |           |
| TX 15 | 0      | USINT | 1.0  | 55.0     | Output | 0       |           |
| TX 16 | 0      | USINT | 1.0  | 56.0     | Output | 0       |           |
| TX 17 | 0      | USINT | 1.0  | 57.0     | Output | 0       |           |
| TX 18 | 0      | USINT | 1.0  | 58.0     | Output | 0       |           |
| TX 19 | 0      | USINT | 1.0  | 59.0     | Output | 0       |           |
| TX 20 | 0      | USINT | 1.0  | 60.0     | Output | 0       |           |
| TX 21 | 0      | USINT | 1.0  | 61.0     | Output | 0       |           |
| TX 22 | 0      | USINT | 1.0  | 62.0     | Output | 0       |           |
| TX 23 | 0      | USINT | 1.0  | 63.0     | Output | 0       |           |
| TX 24 | 0      | USINT | 1.0  | 64.0     | Output | 0       |           |
| TX 25 | 0      | USINT | 1.0  | 65.0     | Output | 0       |           |
| TX 26 | 0      | USINT | 1.0  | 66.0     | Output | 0       |           |
| TX 27 | 0      | USINT | 1.0  | 67.0     | Output | 0       |           |
| TX 28 | 0      | USINT | 1.0  | 68.0     | Output | 0       |           |
| TX 29 | 0      | USINT | 1.0  | 69.0     | Output | 0       |           |
| TX 30 | 0      | USINT | 1.0  | 70.0     | Output | 0       |           |
| TX 31 | 0      | USINT | 1.0  | 71.0     | Output | 0       |           |
| TX 32 | 0      | USINT | 1.0  | 72.0     | Output | 0       |           |

c. The debugging software allows to see the values of the 2 registers received as shown below.

Modbus Slave - Mbslav1

File Edit Connection Setup Display View Window Help

Mbslav1

ID = 1: F = 03

|   | Alias | 00000 |
|---|-------|-------|
| 0 |       | 257   |
| 1 |       | 257   |
| 2 |       |       |
| 3 |       |       |
| 4 |       |       |
| 5 |       |       |
| 6 |       |       |
| 7 |       |       |
| 8 |       |       |
| 9 |       |       |

For Help, press F1.

**Example 2: Verify that the module RTU master reads the 10 holding registers in Level mode with a tool or device such as Modbus Slave debugging software.**

- a. Configure the configuration parameters and select 1 for communication mode i.e. RTU master mode as shown below.

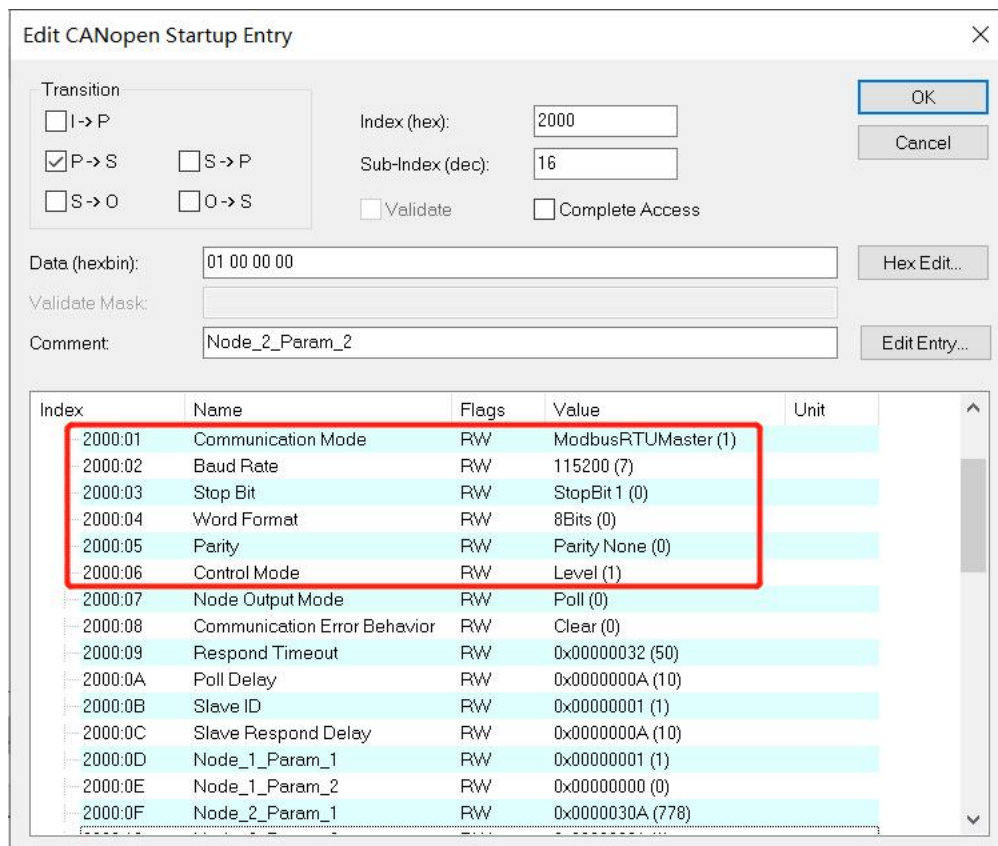
Communication Mode: Select ModbusRTUMaster;

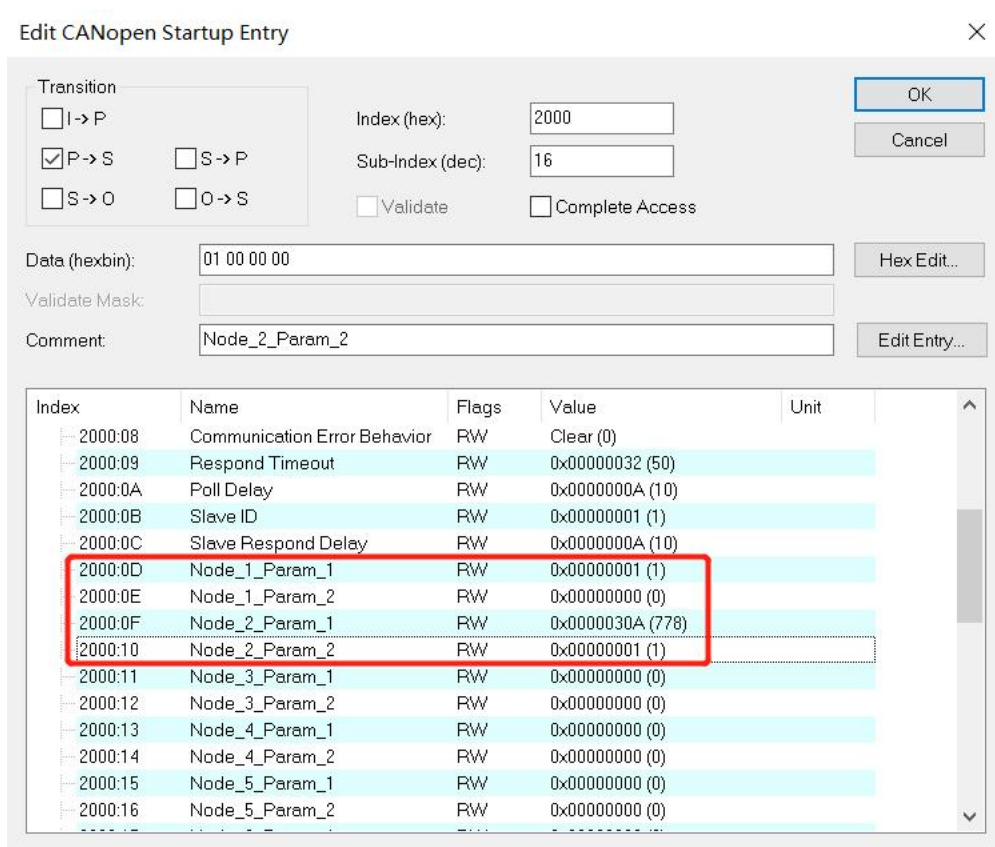
Control Mode: Select Level;

Node\_1\_Param\_1: Configure 0x00000001, see [6.2.8 Control and Status Node Codes](#) for details on configuration parameters;

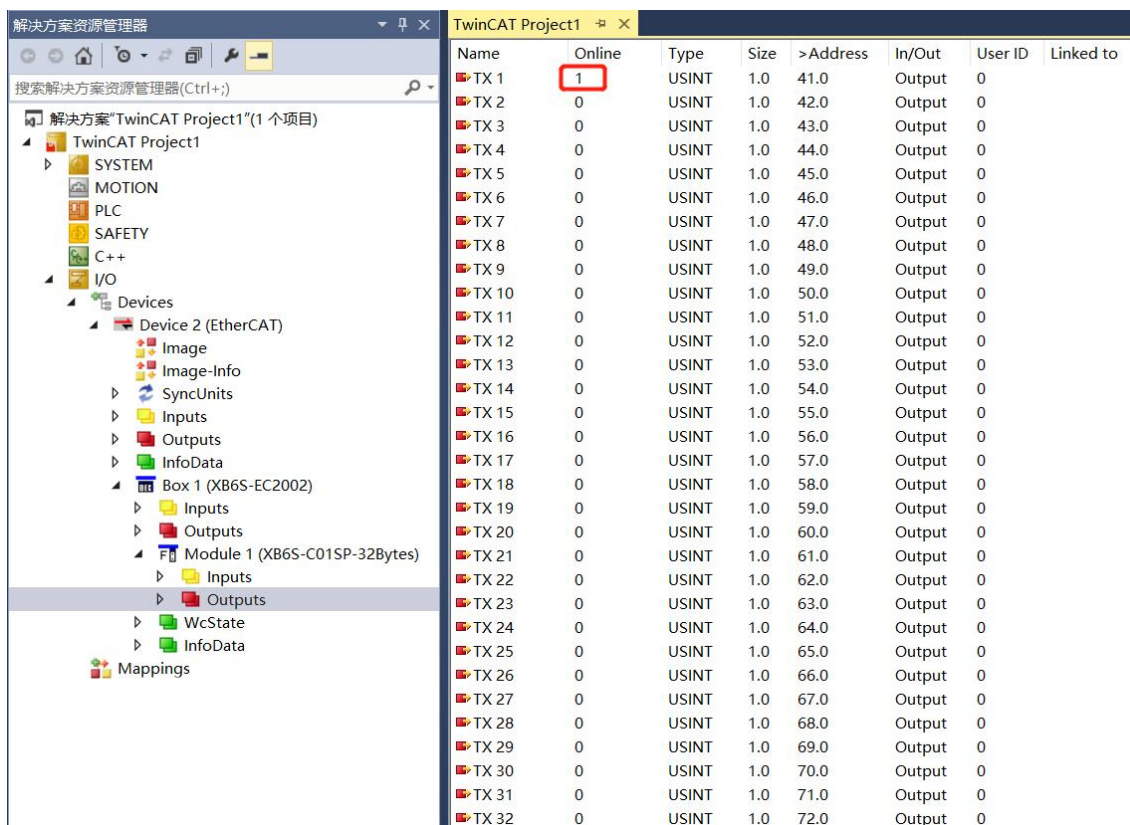
Node\_2\_Param\_1: Configure 0x0000030A, see [6.2.1 Modbus Master Functionality](#) for configuration parameters;

Node\_2\_Param\_2: Configure 0x00000001, see [6.2.1 Modbus Master Functionality](#) for configuration parameters.





- b. Set the control word to 1 in the Downstream data and open the debugging software to send the data as shown below.



c. You can see the received data in the Upstream data as shown below.

The screenshot shows the TwinCAT Project1 interface. On the left is a tree view of the project structure, including 'I/O' and 'Devices'. On the right is a table listing RX data points. A red box highlights the rows for RX 4 through RX 22, where the 'Online' column contains the value 255, indicating that data has been received for these points. Other RX points have an 'Online' status of 0.

| Name  | Online | Type  | Size | >Address | In/Out | User ID | Linked to |
|-------|--------|-------|------|----------|--------|---------|-----------|
| RX 1  | 5      | USINT | 1.0  | 41.0     | Input  | 0       |           |
| RX 2  | 0      | USINT | 1.0  | 42.0     | Input  | 0       |           |
| RX 3  | 0      | USINT | 1.0  | 43.0     | Input  | 0       |           |
| RX 4  | 255    | USINT | 1.0  | 44.0     | Input  | 0       |           |
| RX 5  | 0      | USINT | 1.0  | 45.0     | Input  | 0       |           |
| RX 6  | 255    | USINT | 1.0  | 46.0     | Input  | 0       |           |
| RX 7  | 0      | USINT | 1.0  | 47.0     | Input  | 0       |           |
| RX 8  | 255    | USINT | 1.0  | 48.0     | Input  | 0       |           |
| RX 9  | 0      | USINT | 1.0  | 49.0     | Input  | 0       |           |
| RX 10 | 255    | USINT | 1.0  | 50.0     | Input  | 0       |           |
| RX 11 | 0      | USINT | 1.0  | 51.0     | Input  | 0       |           |
| RX 12 | 255    | USINT | 1.0  | 52.0     | Input  | 0       |           |
| RX 13 | 0      | USINT | 1.0  | 53.0     | Input  | 0       |           |
| RX 14 | 255    | USINT | 1.0  | 54.0     | Input  | 0       |           |
| RX 15 | 0      | USINT | 1.0  | 55.0     | Input  | 0       |           |
| RX 16 | 255    | USINT | 1.0  | 56.0     | Input  | 0       |           |
| RX 17 | 0      | USINT | 1.0  | 57.0     | Input  | 0       |           |
| RX 18 | 255    | USINT | 1.0  | 58.0     | Input  | 0       |           |
| RX 19 | 0      | USINT | 1.0  | 59.0     | Input  | 0       |           |
| RX 20 | 255    | USINT | 1.0  | 60.0     | Input  | 0       |           |
| RX 21 | 0      | USINT | 1.0  | 61.0     | Input  | 0       |           |
| RX 22 | 255    | USINT | 1.0  | 62.0     | Input  | 0       |           |
| RX 23 | 0      | USINT | 1.0  | 63.0     | Input  | 0       |           |
| RX 24 | 0      | USINT | 1.0  | 64.0     | Input  | 0       |           |
| RX 25 | 0      | USINT | 1.0  | 65.0     | Input  | 0       |           |
| RX 26 | 0      | USINT | 1.0  | 66.0     | Input  | 0       |           |
| RX 27 | 0      | USINT | 1.0  | 67.0     | Input  | 0       |           |
| RX 28 | 0      | USINT | 1.0  | 68.0     | Input  | 0       |           |
| RX 29 | 0      | USINT | 1.0  | 69.0     | Input  | 0       |           |
| RX 30 | 0      | USINT | 1.0  | 70.0     | Input  | 0       |           |
| RX 31 | 0      | USINT | 1.0  | 71.0     | Input  | 0       |           |
| RX 32 | 0      | USINT | 1.0  | 72.0     | Input  | 0       |           |

## 7. Example of Freepport\_Input function

**Example: Receive an 8-byte data in Level mode using Freepport\_Input through a tool or device such as a serial port debugging assistant.**

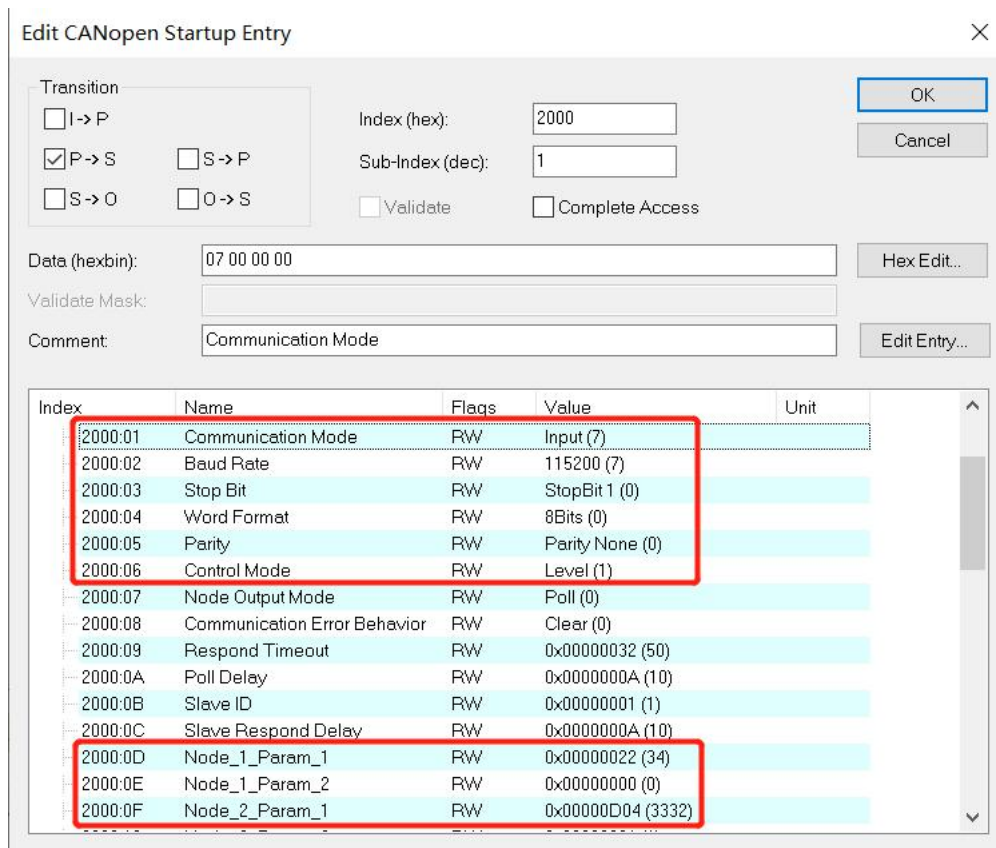
- a. Configure the configuration parameters with communication mode selection 7 i.e. Input mode as shown below.

Communication Mode: Select Input;

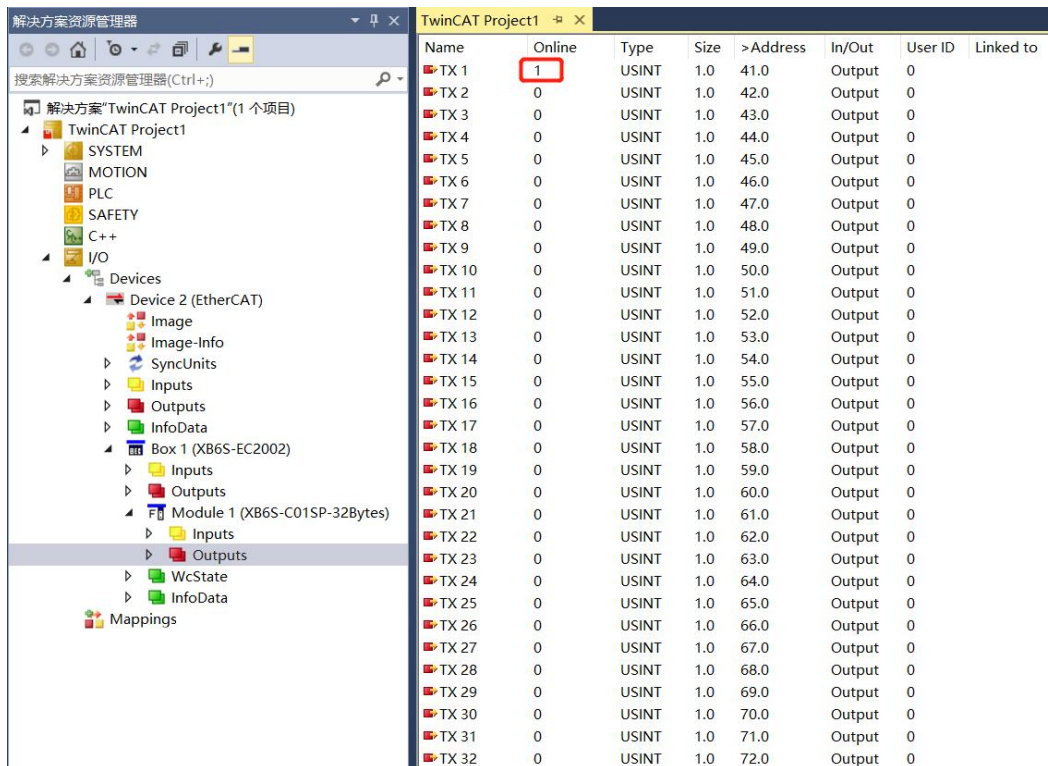
Control Mode: Select Level;

Node\_1\_Param\_1: Configure 0x00000022, see [6.2.8 Control and Status Node Codes](#) for details on configuration parameters;

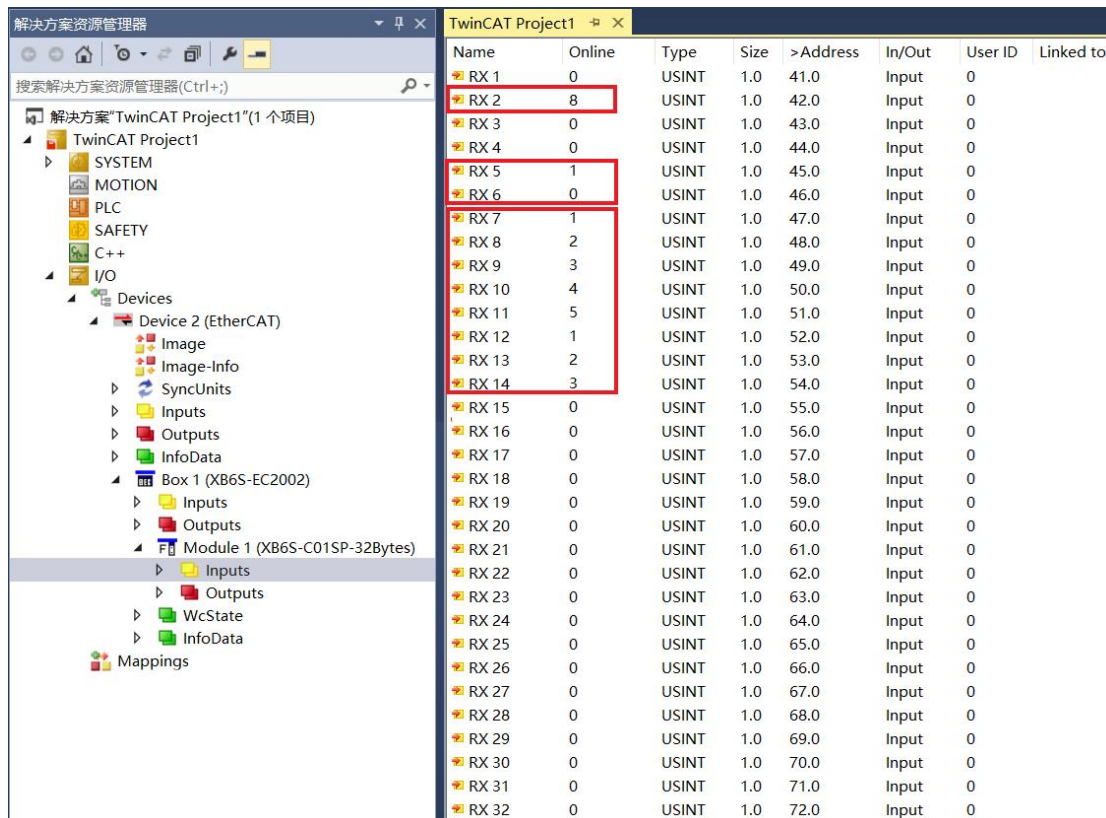
Node\_2\_Param\_1: Configure 0x00000D04, see [6.2.6 Freepport Freepport Functions](#) for details on configuration parameters.



- b. Set the control word to 1 in the downstream data and open the serial port debugging assistant to send data as shown below.



- c. The Upstream data can be seen in the received data length of 8, the received data count of 1, and the received 8 bytes of data (RX7~RX14) as shown below.



## 6.4.2 Application in Sysmac Studio software environment

### 1、 Preliminary

- **Hardware Environment**

- **Module Model XB6S-C01SP**
- **EtherCAT Coupler, End Cap**  
This description uses the XB6S-EC2002 coupler as an example
- **One computer with Sysmac Studio software pre-installed**
- **One Omron PLC, this description takes model NX1P2-9024DT as an example.**
- **Shielded cables for EtherCAT**
- **One switching power supply**
- **Device Configuration Files**

Configuration file access: <https://www.solidotech.com/documents/configfile>

- **Hardware configuration and wiring**

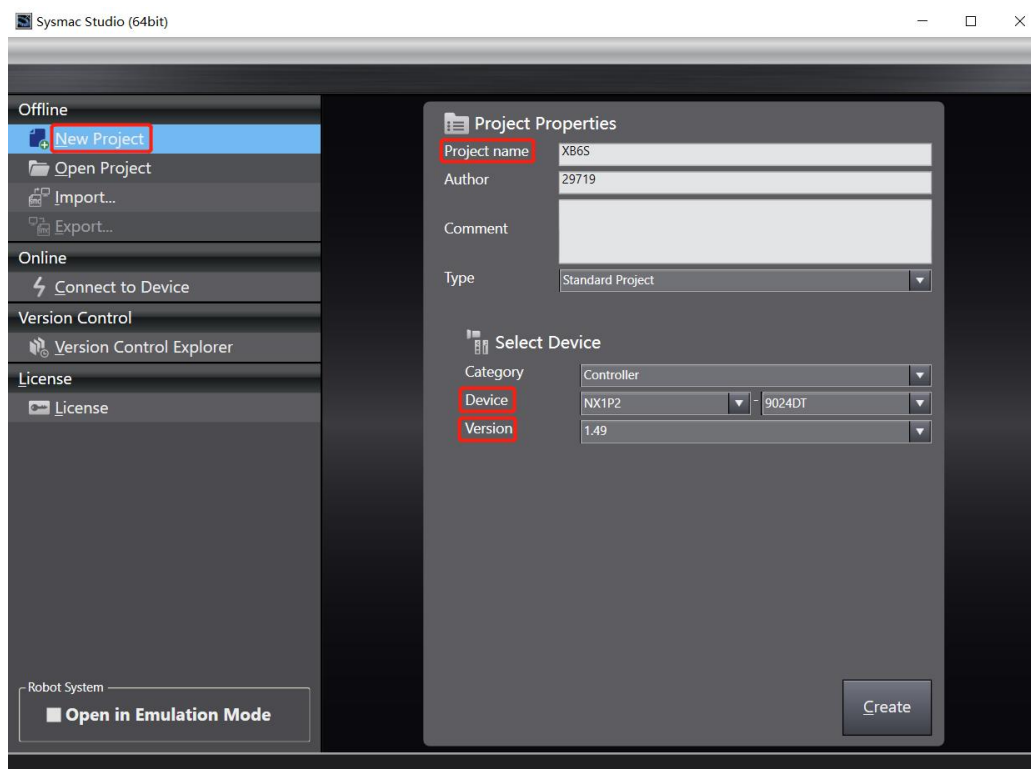
Follow "[4 Installation and uninstall](#)" and "[5 Wiring](#)".

- **Computer IP requirements**

Set the IP address of the computer and the IP address of the PLC to make sure they are on the same network segment.

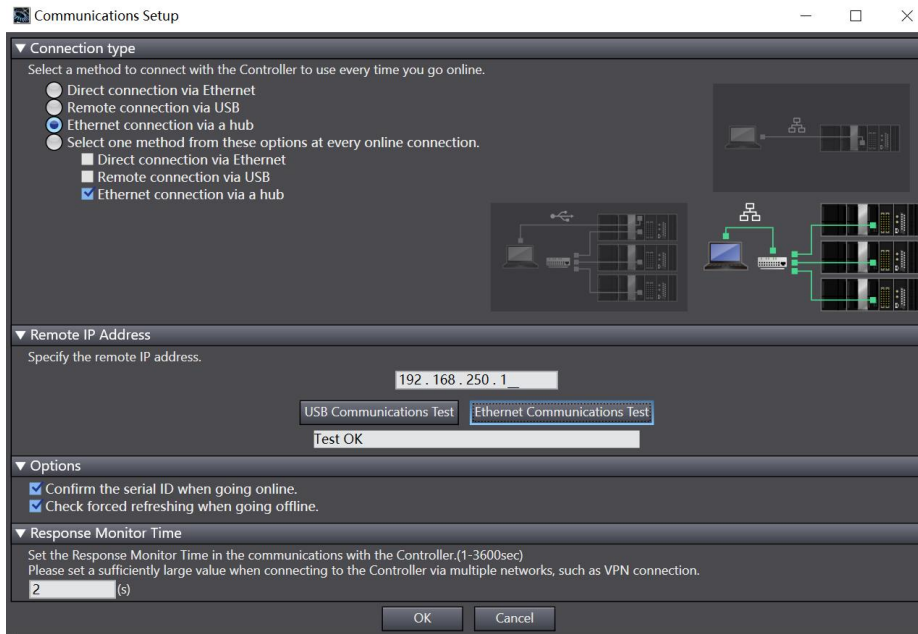
### 2、 New project

- a. Open Sysmac Studio software, click "New Project".



- Project name: Customizable .

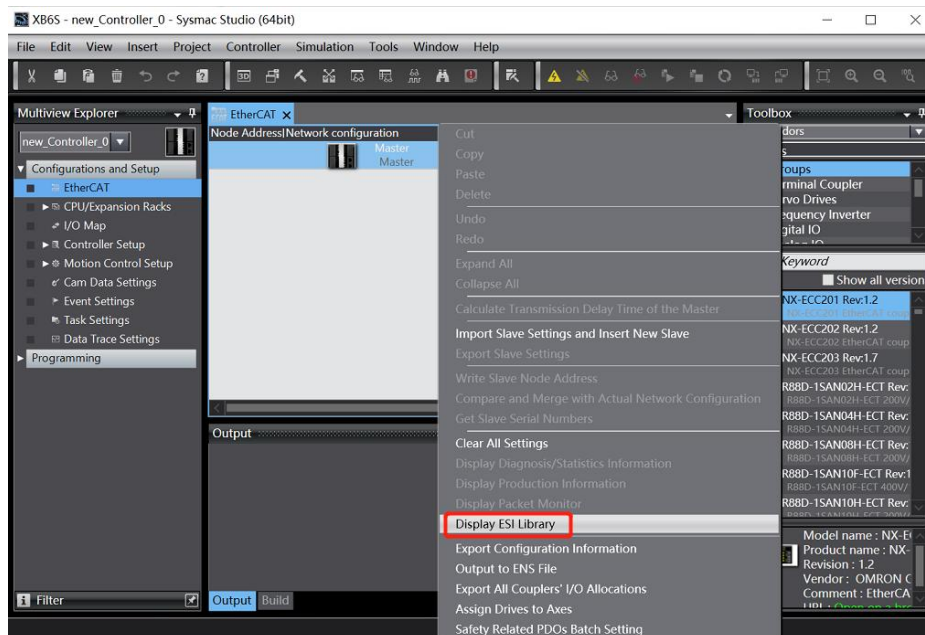
- Select device: "Device" selects the corresponding PLC model and "Version" selects the corresponding PLC version number.
- b. When you have finished entering the project properties, click "Create".
- c. Click "Controller -> Communications Setup" in the menu bar, select the method to be used every time you connect to the controller when you are online, and enter "Remote IP Address". Enter "Remote IP Address" as shown in the following figure.



- d. Click "Ethernet communication test", the system shows that the test is successful.

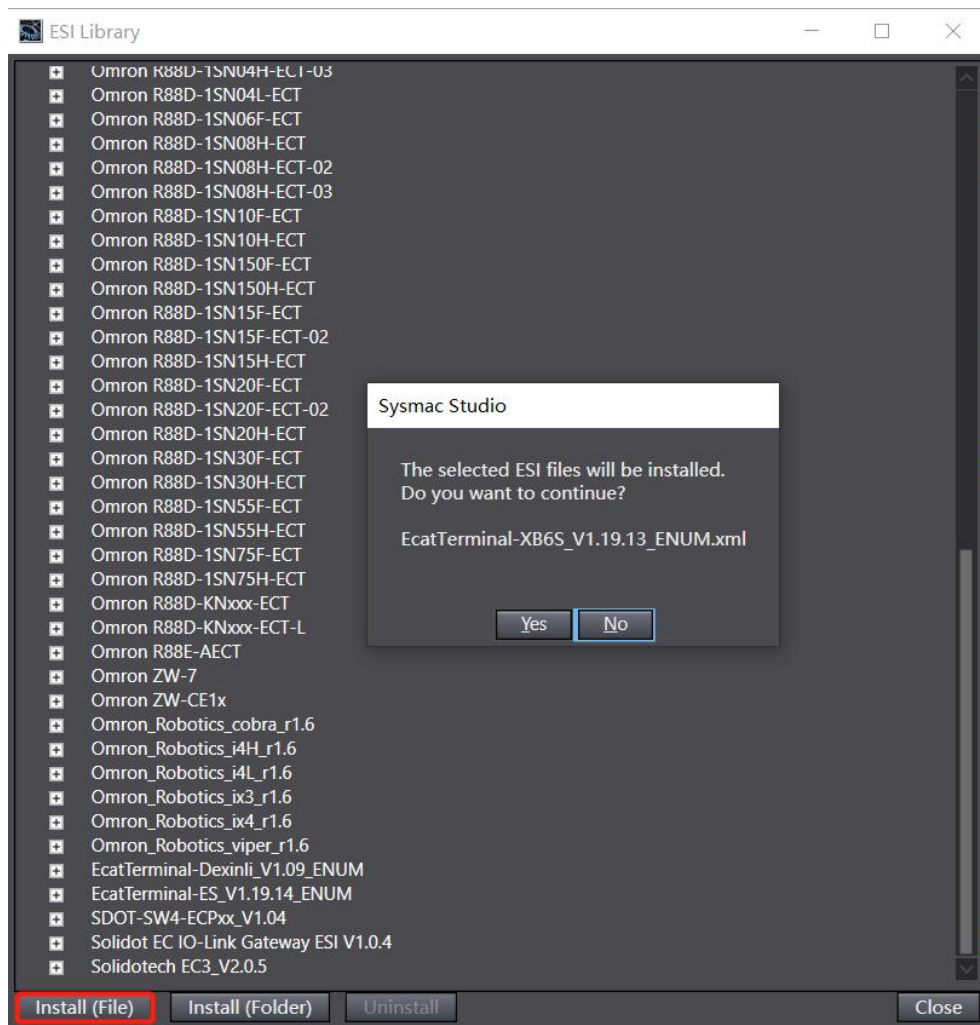
### 3. Installation of XML files

- a. Expand "Configurations and Setup" in the left navigation tree and double click on "EtherCAT".
- b. Right-click "Master" and select "Display ESI Library" as shown in the following figure.





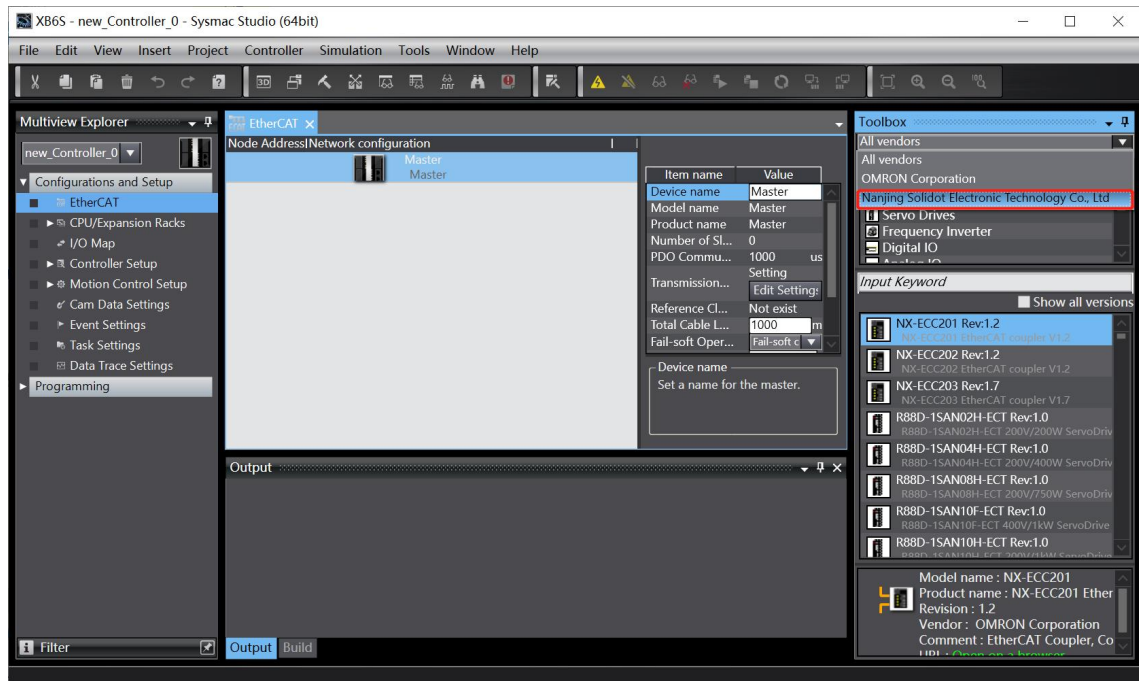
- c. In the pop-up "ESI Library" window, click "Install (File)", select the module's XML file path, click "Yes" to complete the installation, as shown in the following figure.



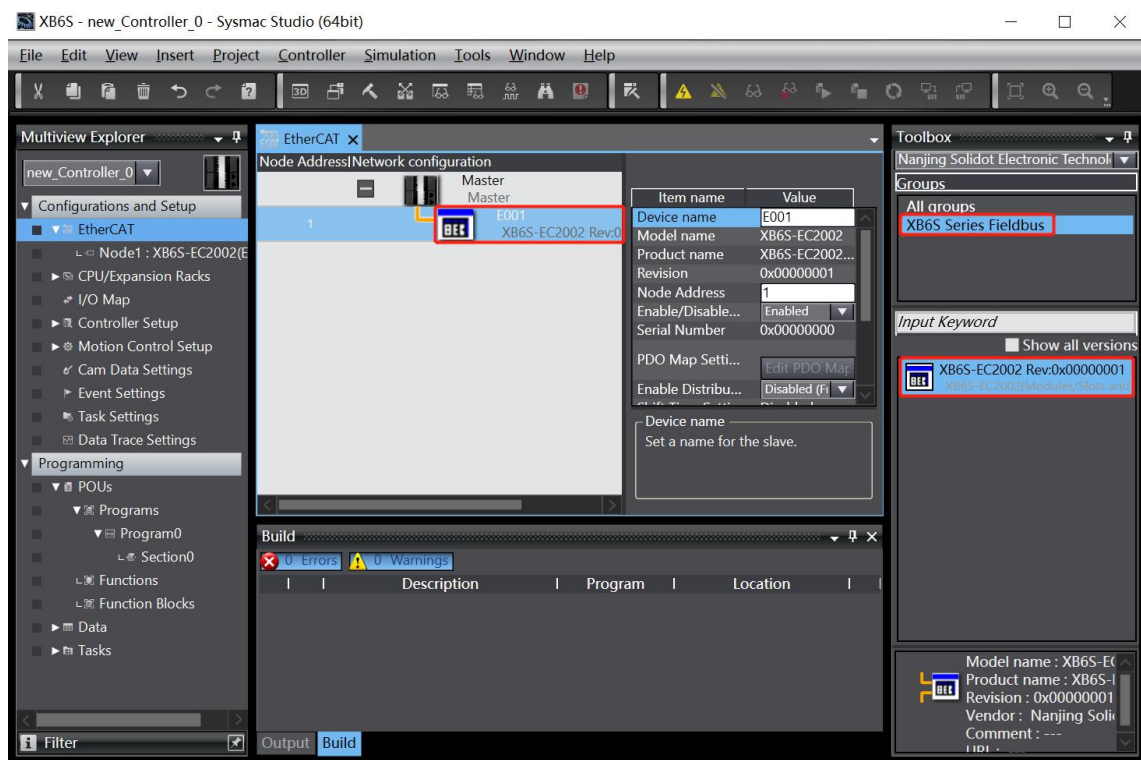
### 4. Add Device

There are two ways to add devices: online scanning and offline adding, and this note introduces offline adding as an example.

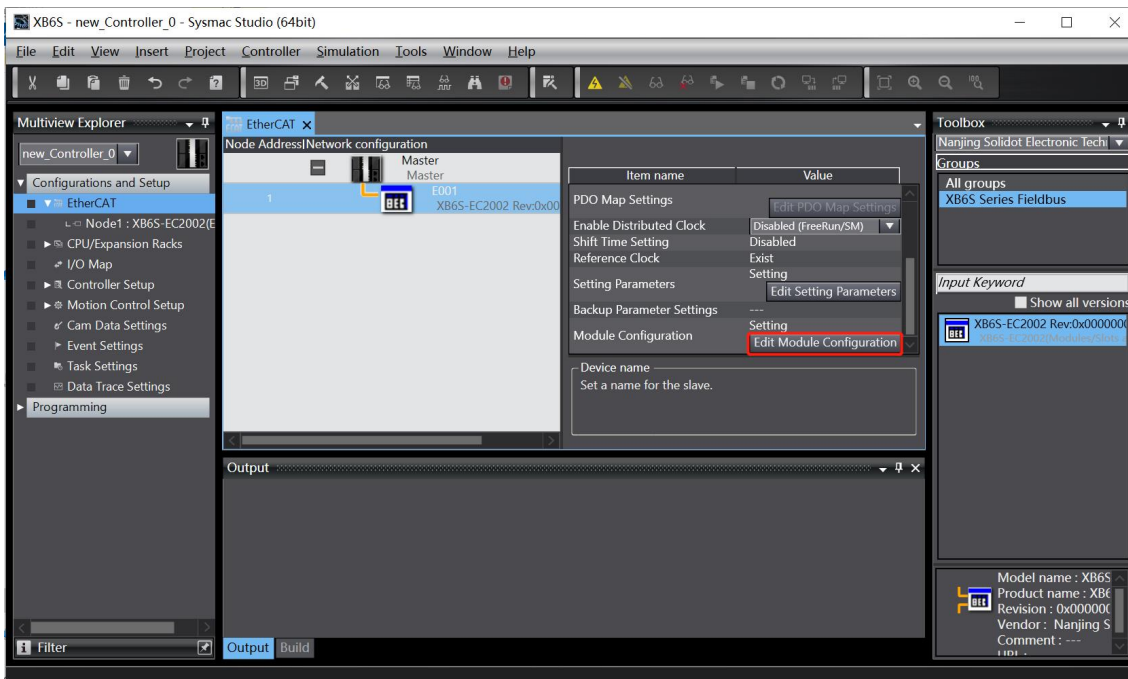
- a. In the right side of the "Toolbox" column, click to expand all suppliers, select "Nanjing Solidot Electronic Technology Co., Ltd."



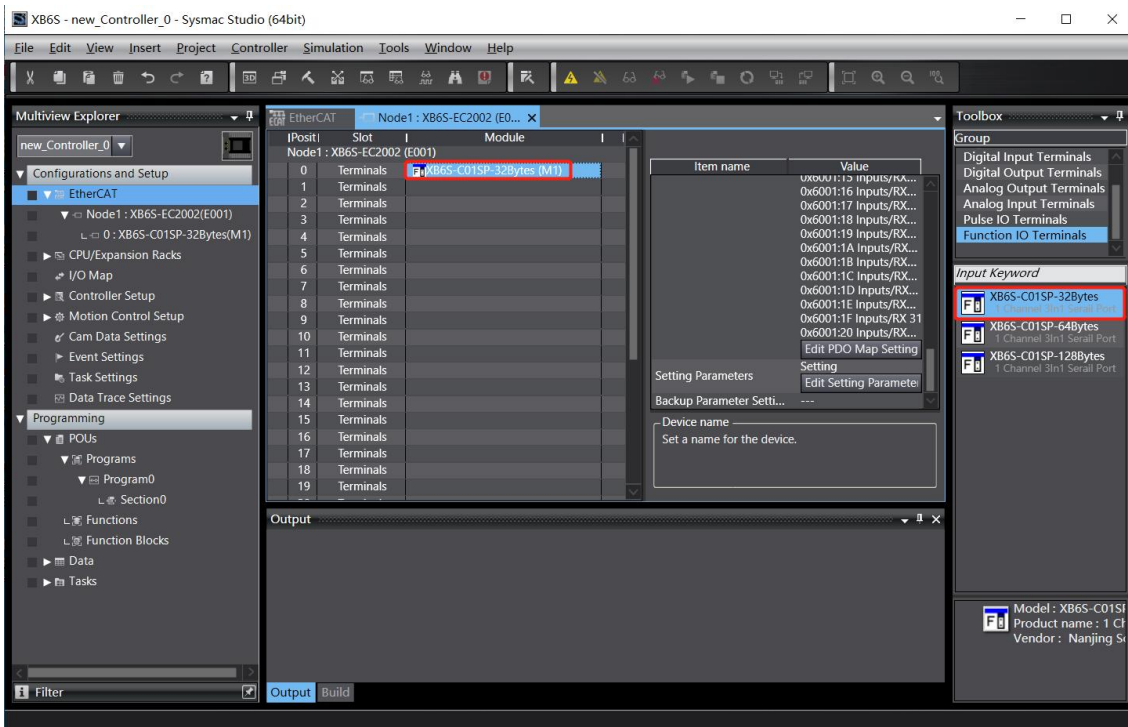
- b. Click to select XB6S Series Fieldbus and double-click the XB6S-EC2002 Coupler Module to add a slave device, as shown below.



- c. In the EtherCAT main page, select the XB6S-EC2002 coupler module you just added, and choose "Edit Module Configuration", as shown in the following figure.

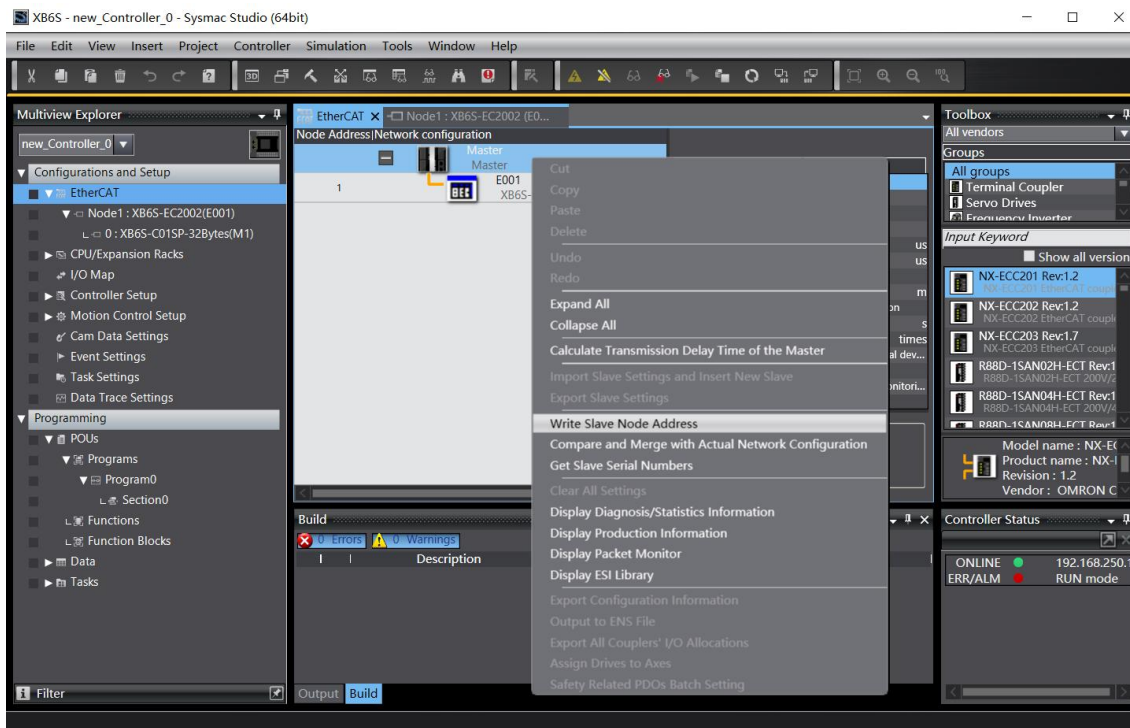


- d. Position the cursor in the "Module Module", click on the module in the right toolbox module list, and add I/O modules one by one in the order of I/O module configuration. Note: The order and model number must be consistent with the physical topology!

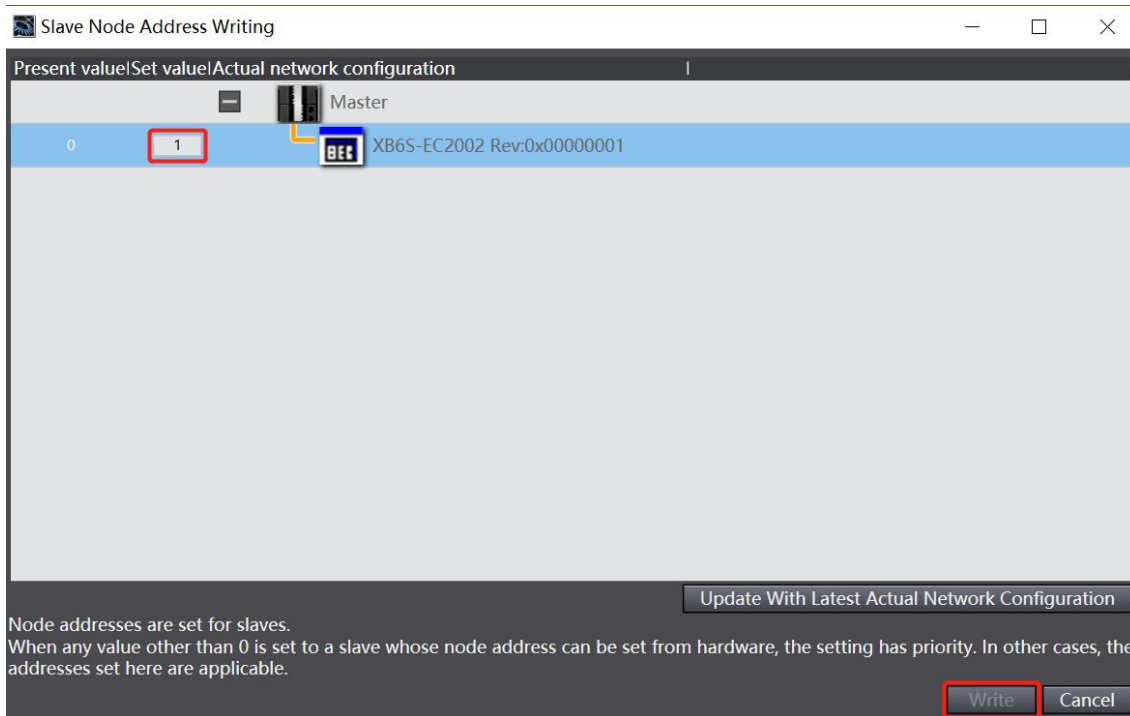


### 5. Setting the node address

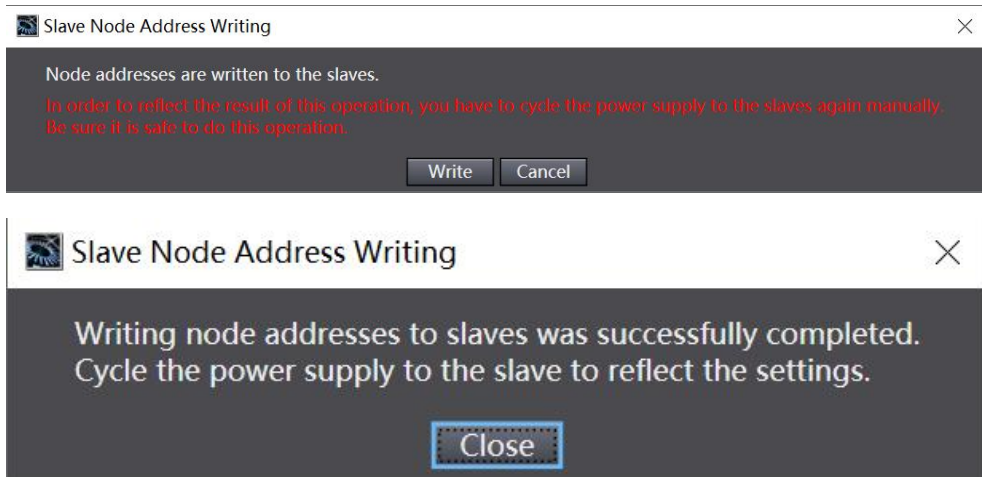
- a. Click "Controller -> Online" in the menu bar to turn the controller to online status. Right-click the master device, click and select "Write Slave Node Address", as shown in the following figure.



- b. In the Set Node Address window, click the value under Set Value, enter the node address, and click "Write" to change the slave device node address, as shown in the following figure.

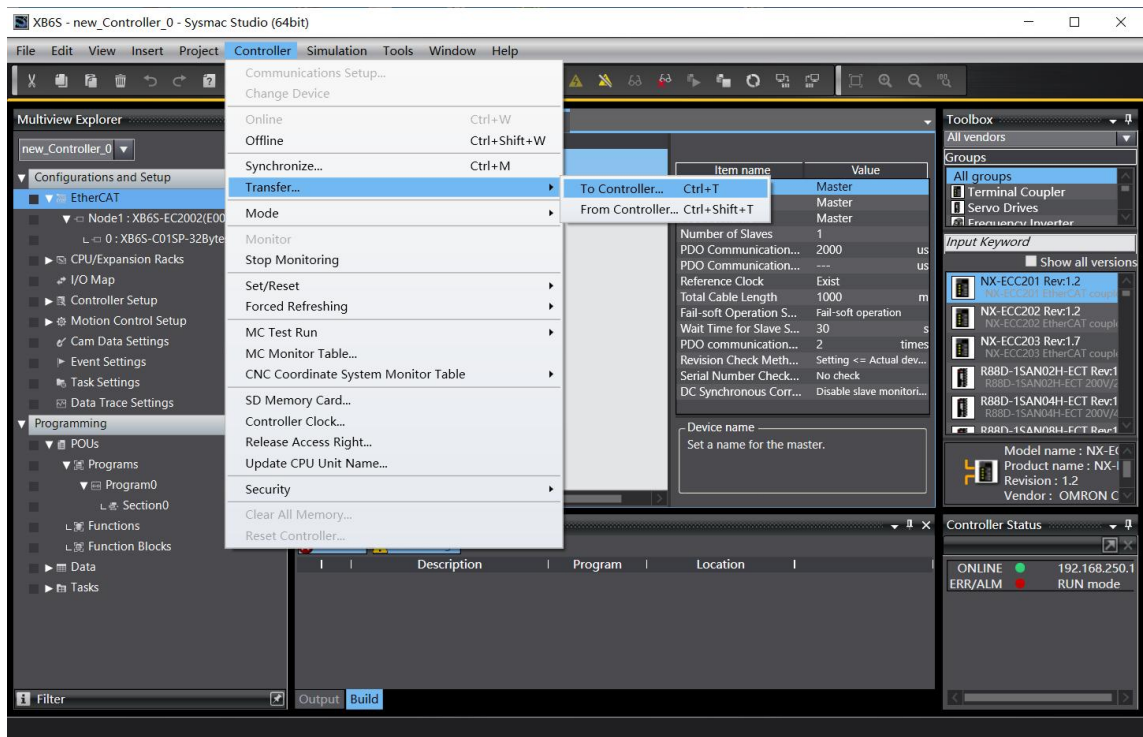


- c. After writing, a re-power prompt will pop up, as shown in the following figure, click "Write", and then follow the prompts to reboot the power from the device.

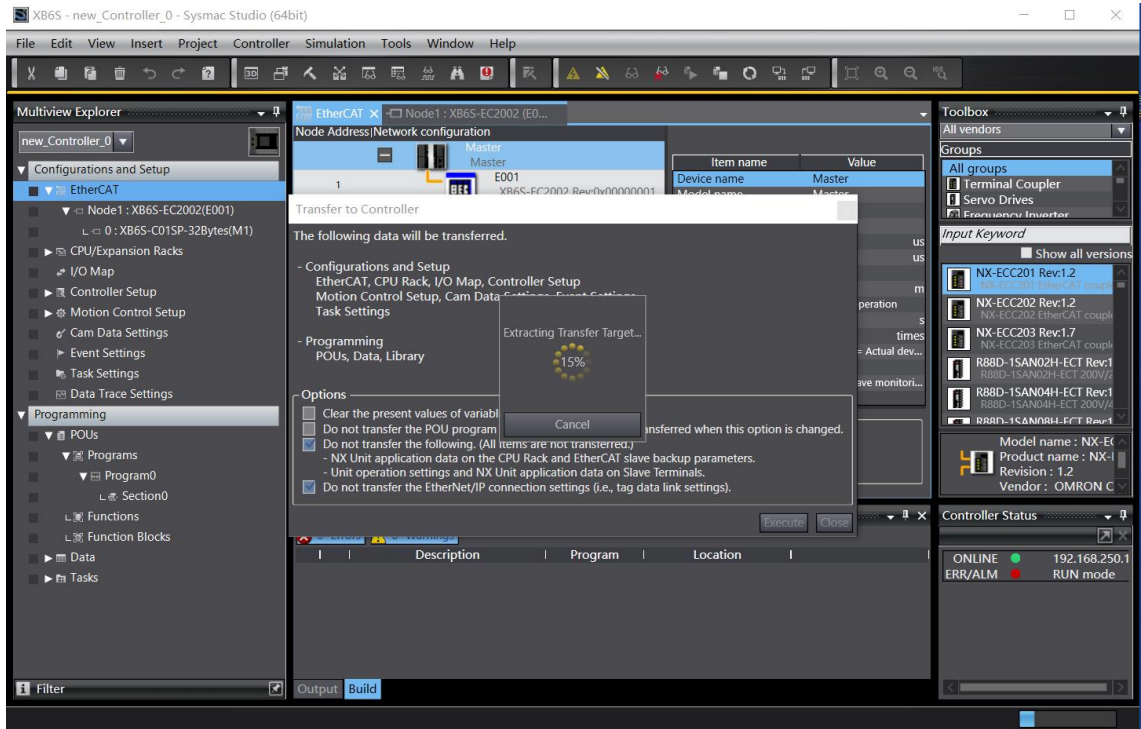


### 6. Download configuration to PLC

- a. Click the button "Controller -> Transfer...-> To Controller..." in the menu bar, as shown in the following figure.

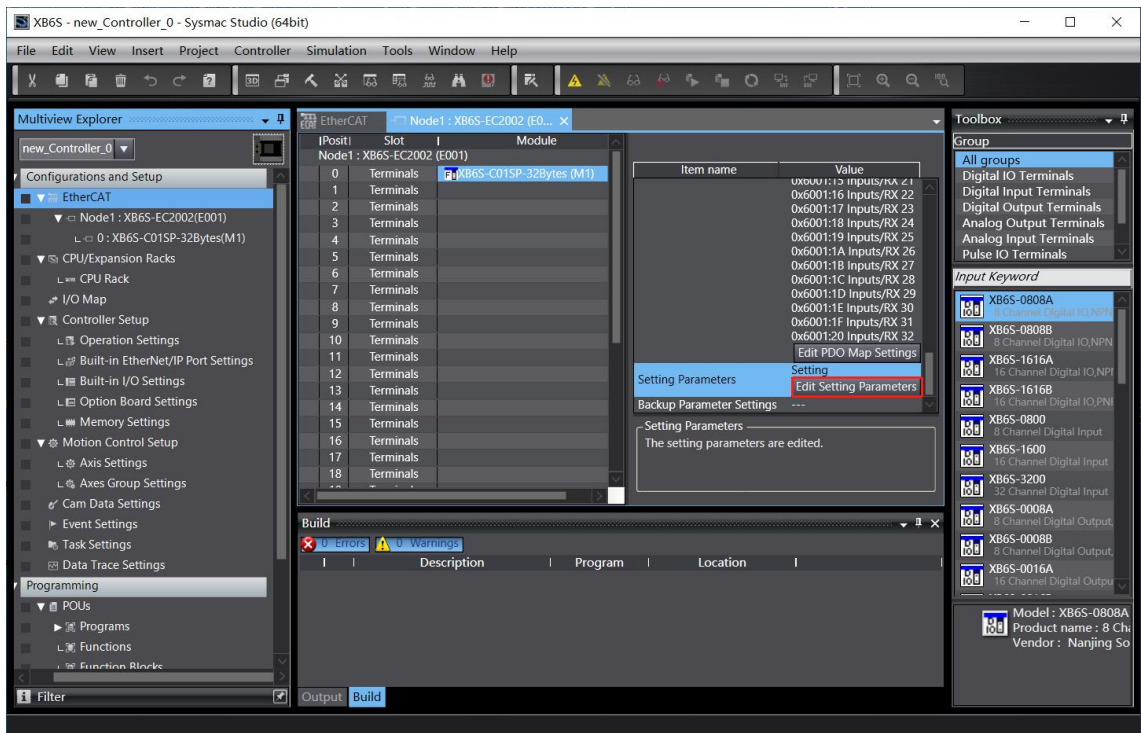


- b. The configuration will be downloaded to the PLC, the transmission confirmation pop-up window will appear, click "Execute", and then click "Yes/OK" in the subsequent pop-up window, as shown in the figure below, after the download is completed, you need to re-power up.



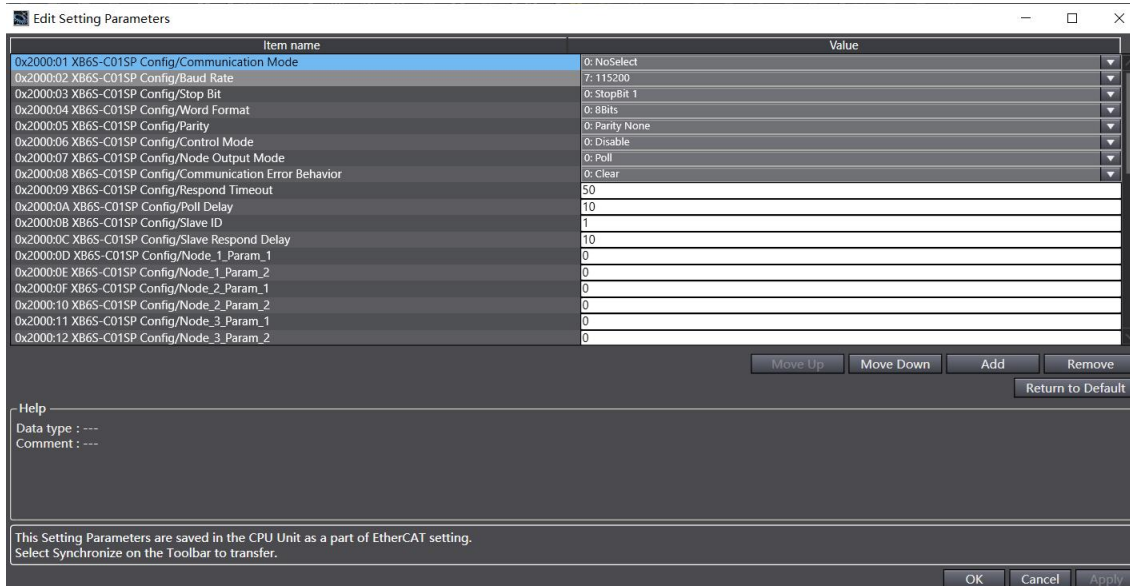
## 7. Setting Parameters

- a. Switch the configuration to offline state, edit module configuration page in node 1, select XB6S-C01SP-32Bytes module, click "Edit Setting Parameters", as shown in the following figure.

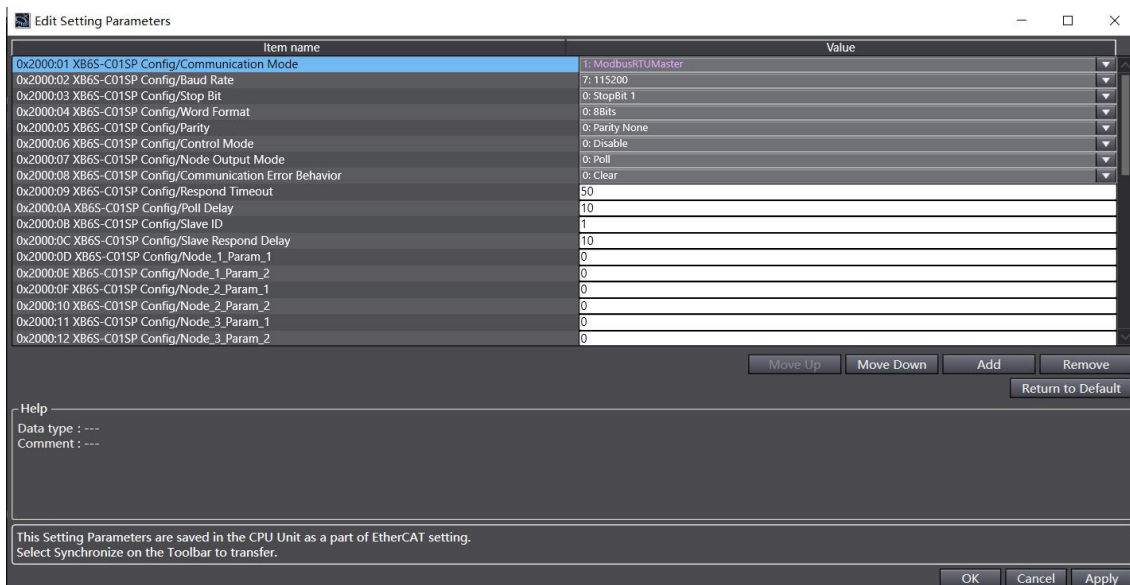


Note: If the PLC firmware version is too low, you need to use EC\_CoESDOWrite, EC\_CoESDORead instruction to write and read the SDO address.

- b. In the XB6S-C01SP Parameter Setting page, you can see 44 configuration parameters, click on any parameter to set the related configuration, as shown in the following figure.

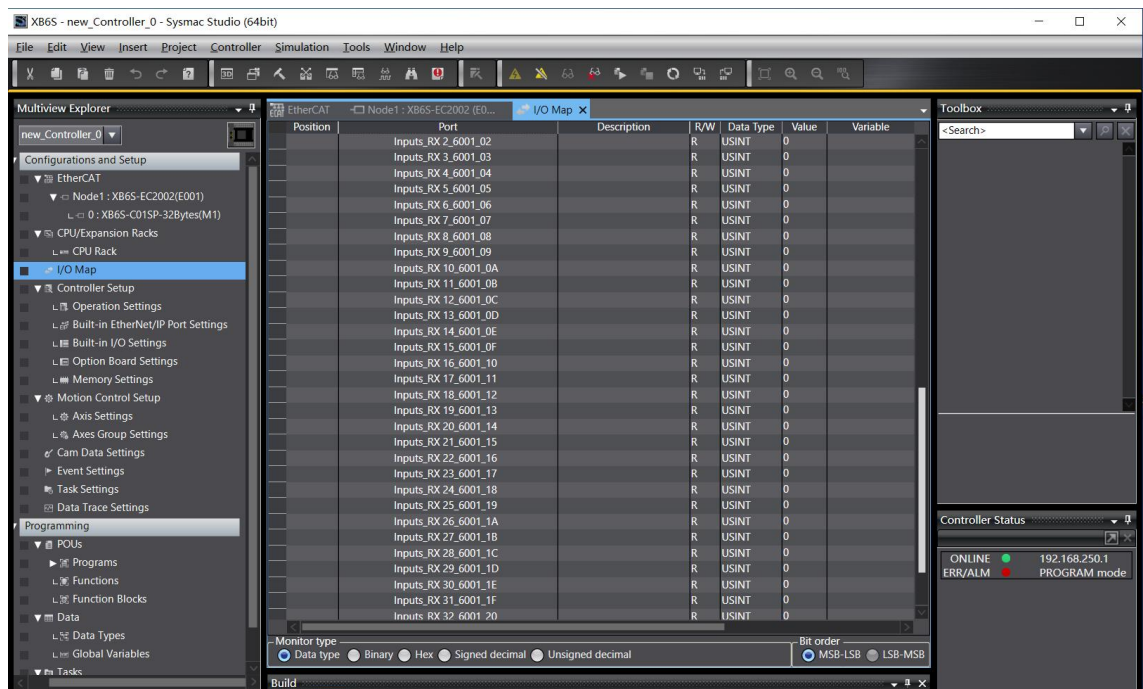
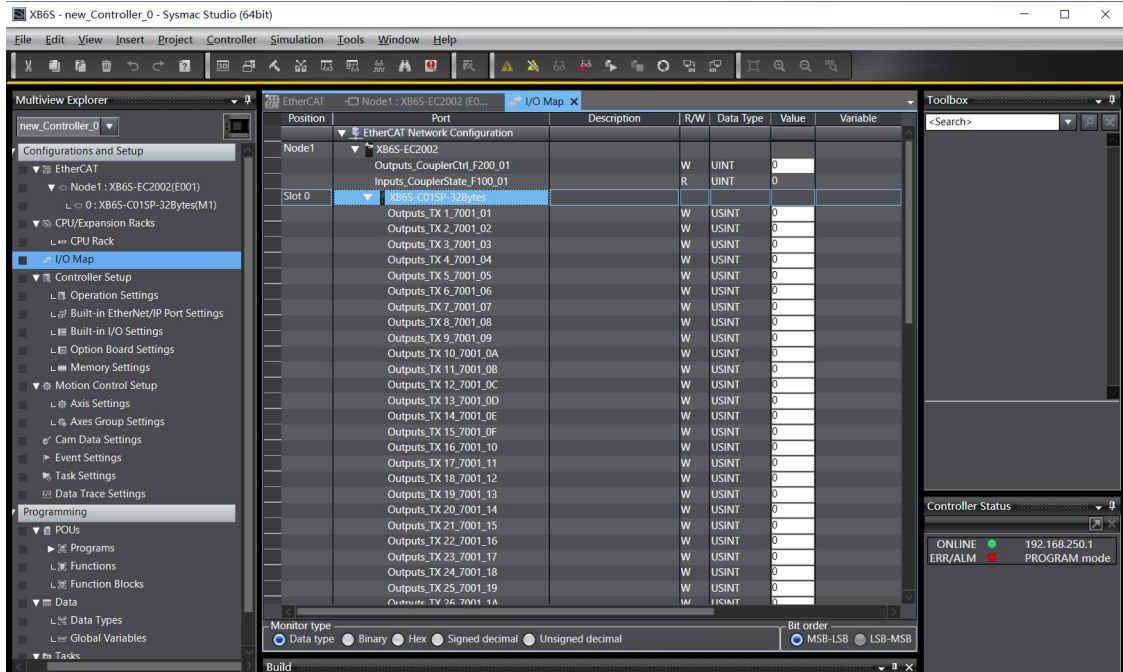


- c. For example, to modify the communication mode parameters, you can click "Communication Mode" to modify the parameter values, as shown in the following figure. After all the parameters are configured, you need to re-download the program to the PLC, and the PLC and module need to be re-powered.



## 8. I/O Functions

- a. Double-click "I/O Map" in the left navigation tree to see the mapping table of the modules in the topology, so that you can monitor the input and output values of the communication modules, as shown in the figure below.





### 9、RTU Master Mode Function Example

**Example 1: Verify that the module RTU master writes the values of the 2 holding registers in disable mode with a tool or device such as Modbus Slave debugging software.**

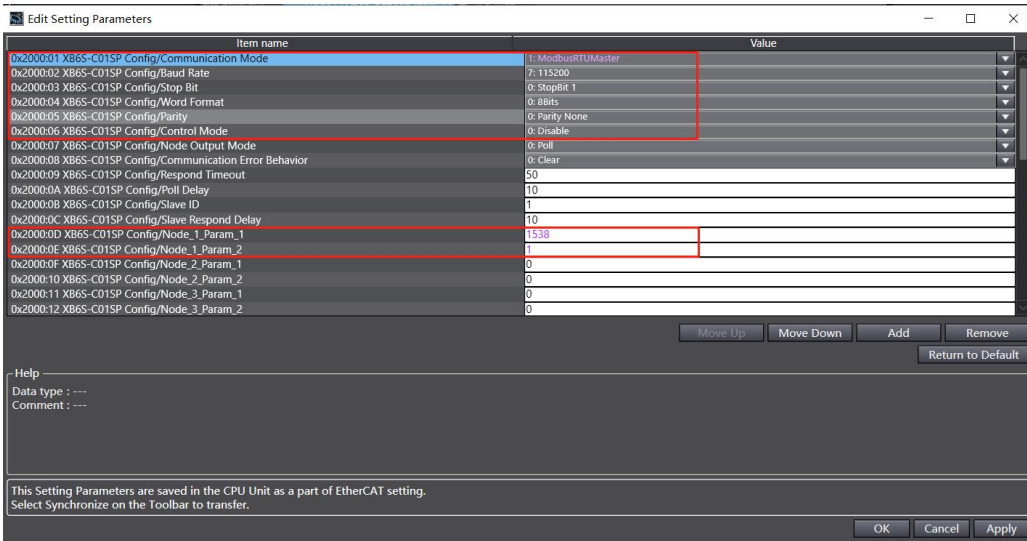
- a. Configure the configuration parameters and select 1 for communication mode i.e. RTU master mode as shown below.

Communication Mode: Select ModbusRTUMaster;

Control Mode: Select Disable;

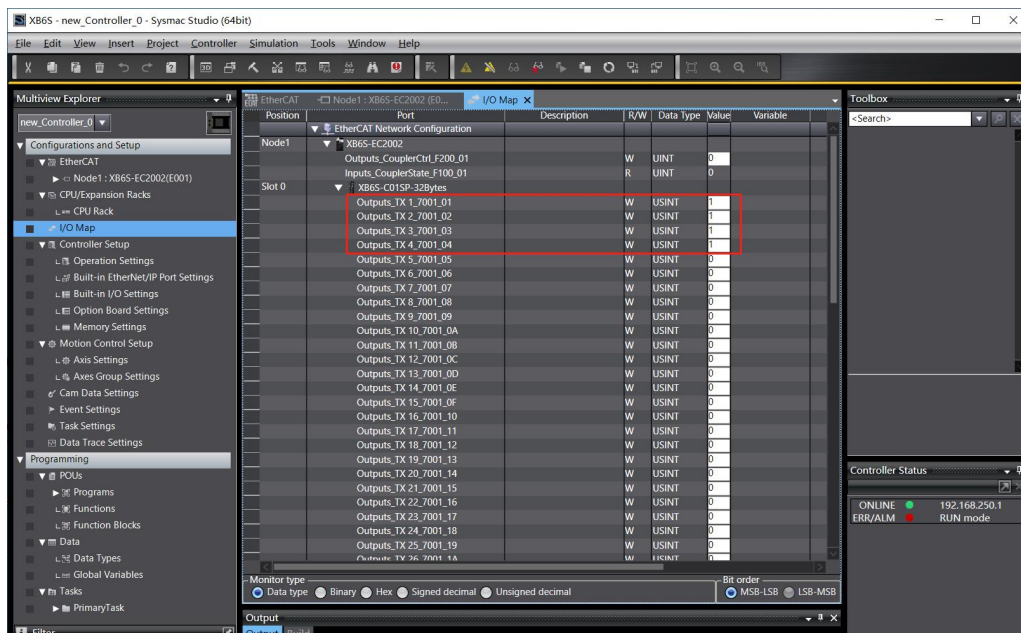
Node\_1\_Param\_1: Configure 1538 (0x00000602), see [6.2.1 Modbus Master Function](#) for configuration parameters;

Node\_1\_Param\_2: Configuration 1 (0x00000001), see [6.2.1 Modbus Master Function](#) for configuration parameters.

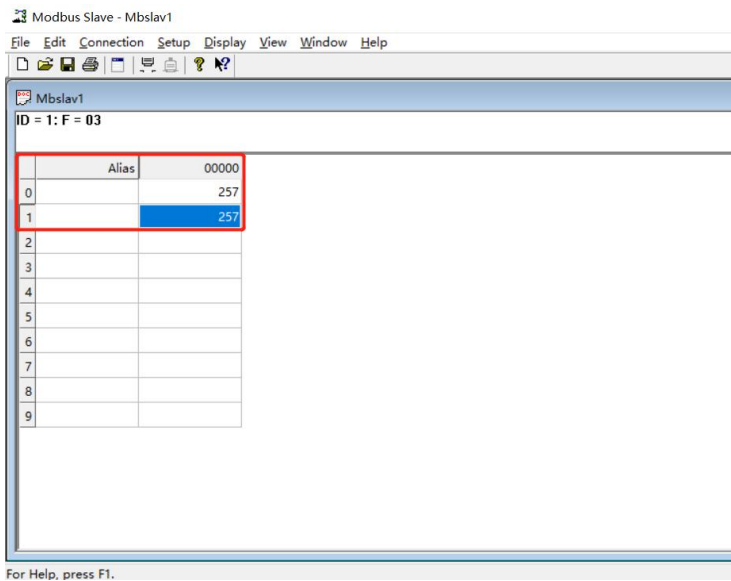


After all parameters are configured, the program needs to be re-downloaded to the PLC, and the PLC and module need to be re-powered.

- b. In the downstream data, the values of the 2 holding registers are written as shown below.



- c. The debugging software allows to see the values of the 2 registers received as shown below.



**Example 2: Verify that the module RTU master reads the 10 holding registers in Level mode with a tool or device such as Modbus Slave debugging software.**

- a. Configure the configuration parameters and select 1 for communication mode i.e. RTU master mode as shown below.

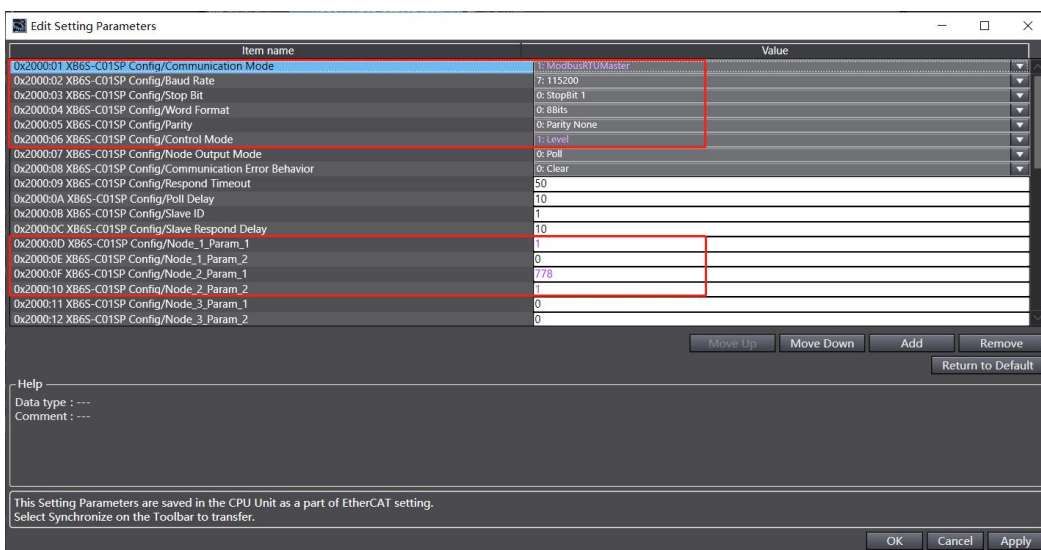
Communication Mode: Select ModbusRTUMaster;

Control Mode: Select Level;

Node\_1\_Param\_1: Configuration 1 (0x00000001), see [6.2.8 Control and Status Node Codes](#) for details on configuration parameters;

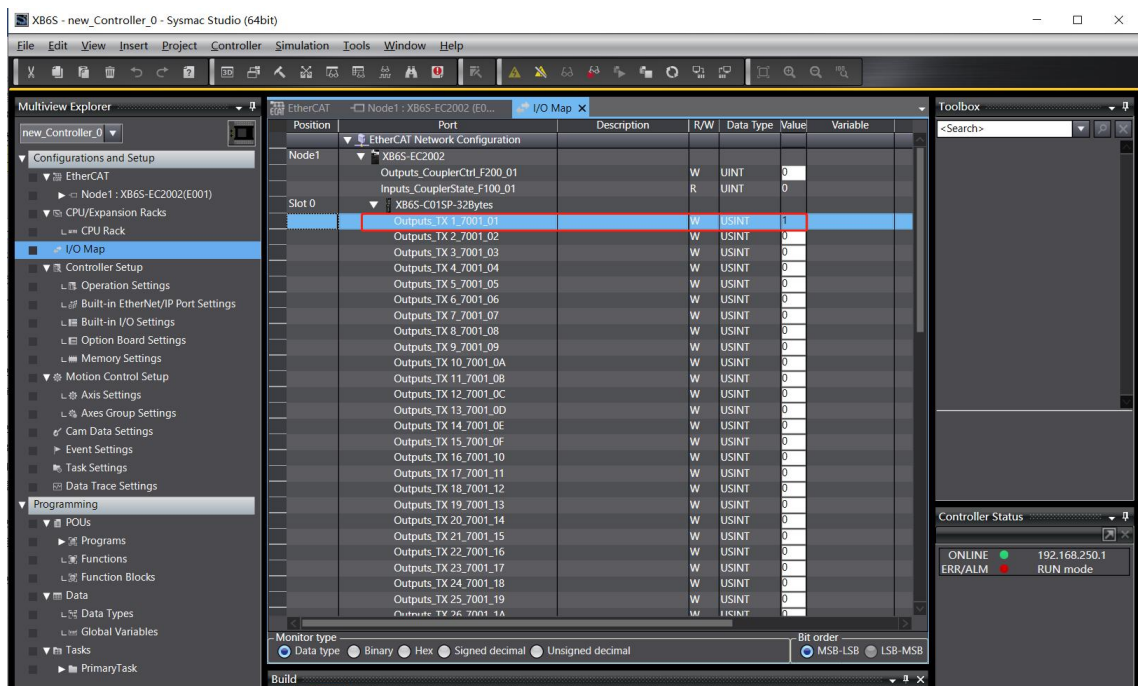
Node\_2\_Param\_1: Configuration 778 (0x0000030A), see [6.2.1 Modbus Master Function](#) for configuration parameters;

Node\_2\_Param\_2: Configuration 1 (0x00000001), see [6.2.1 Modbus Master Function](#) for configuration parameters.

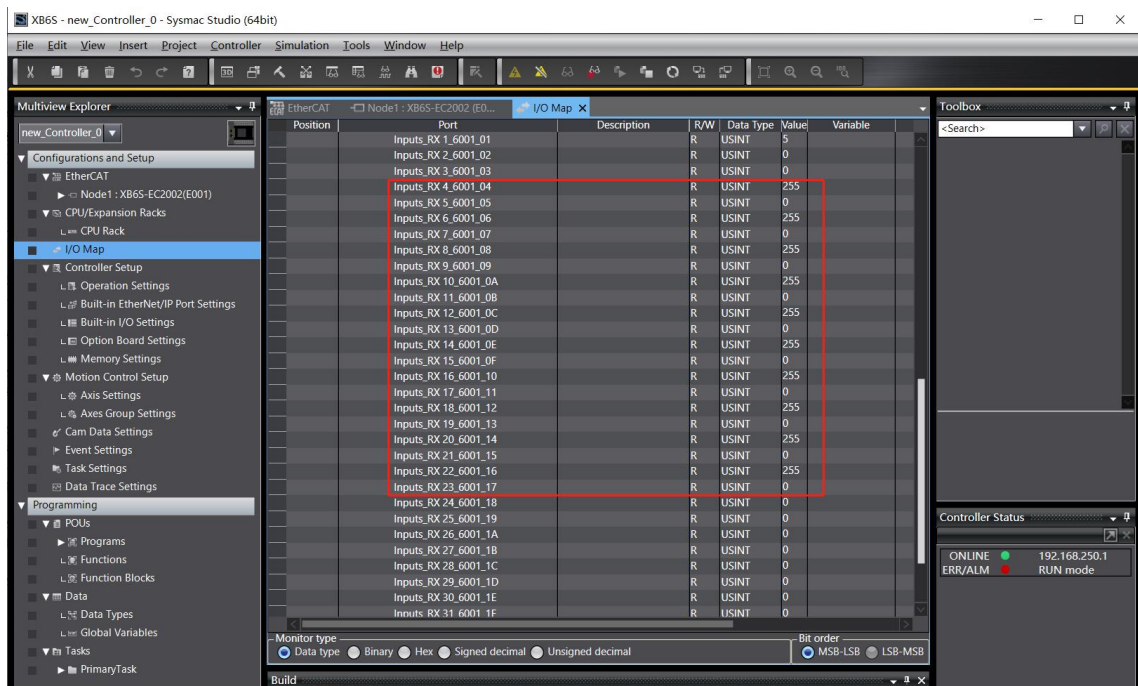


**After all parameters are configured, the program needs to be re-downloaded to the PLC, and the PLC and module need to be re-powered.**

- b. Set the control word to 1 in the Downstream data and open the debugging software to send the data as shown below.



- c. You can see the received data in the Upstream data as shown below.



## 10. Example of Freeport\_Input function

**Example: Receive an 8-byte data in Level mode using Freeport\_Input through a tool or device such as a serial port debugging assistant.**

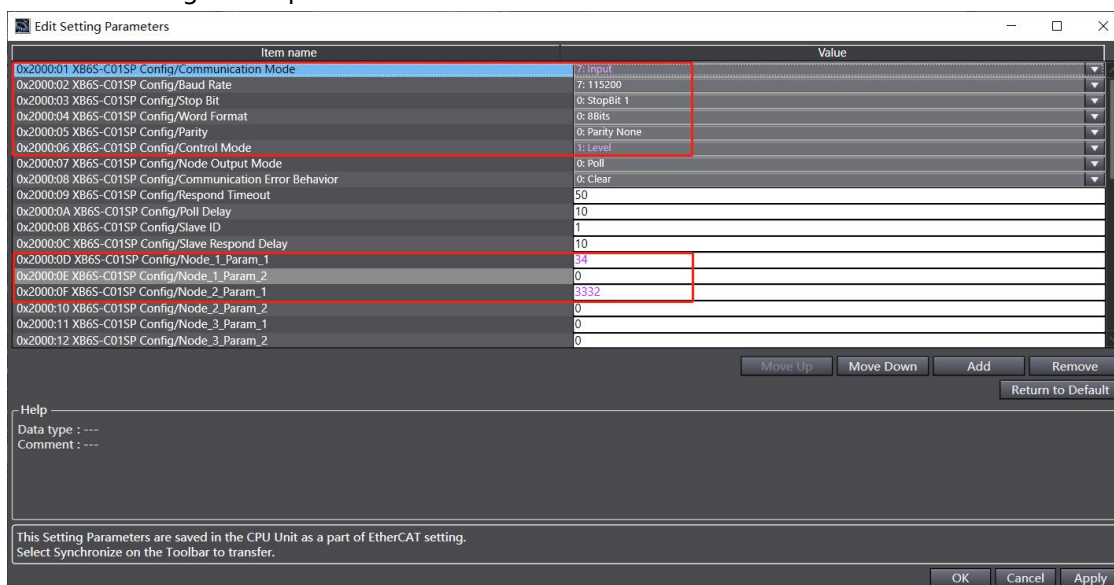
- a. Configure the configuration parameters and select 7 for communication mode i.e. Input mode as shown below.

Communication Mode: Select Input;

Control Mode: Select Level;

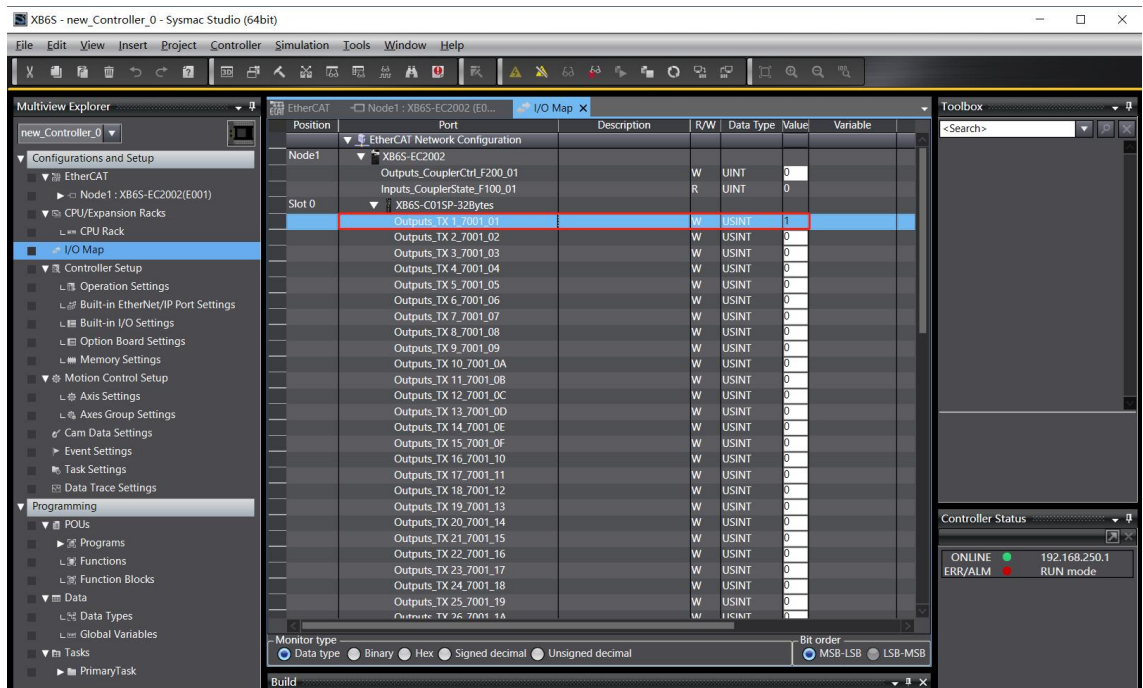
Node\_1\_Param\_1: Configuration 34 (0x00000022), see [6.2.8 Control and Status Node Codes](#) for details on configuration parameters;

Node\_2\_Param\_1: Configuration 3332 (0x00000D04), see [6.2.6 Freeport Freeport Functions](#) for details on configuration parameters.

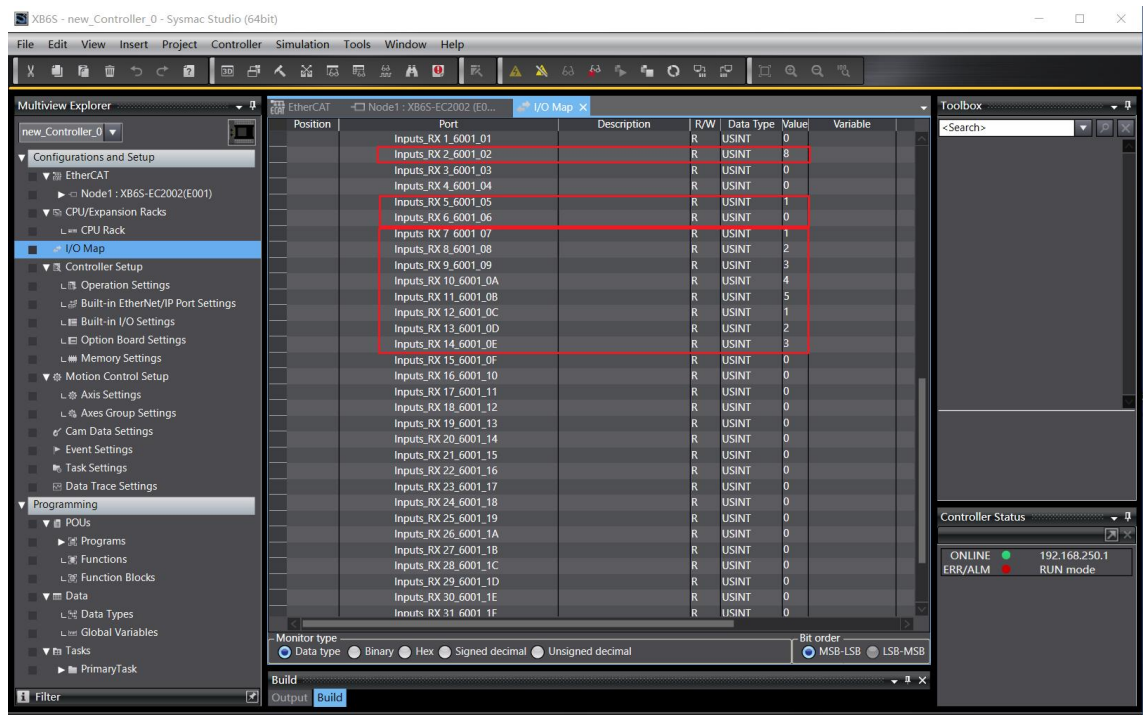


**After all parameters are configured, the program needs to be re-downloaded to the PLC, and the PLC and module need to be re-powered.**

- b. Set the control word to 1 in the downstream data and open the serial port debugging assistant to send data as shown below.



- c. The Upstream data can be seen in the received data length of 8, the received data count of 1, and the received 8 bytes of data (RX7~RX14) as shown below.



## 6.4.3 Application in TIA Portal V17 software environment

### 1、 Preliminary

#### ● Hardware environment

- **Module model XB6S-C01SP**
- **PROFINET Bus Coupler Module, End Cap**  
This description uses the XB6S-PN2002 coupler module as an example
- **One computer with pre-installed TIA Portal V17 software**
- **Shielded cables for PROFINET**
- **One Siemens PLC, this description takes Siemens S7-1500 CPU 1511-1 PN as an example.**
- **One switching power supply**
- **Module installation rails and rail fixings**
- **Device Configuration Files**

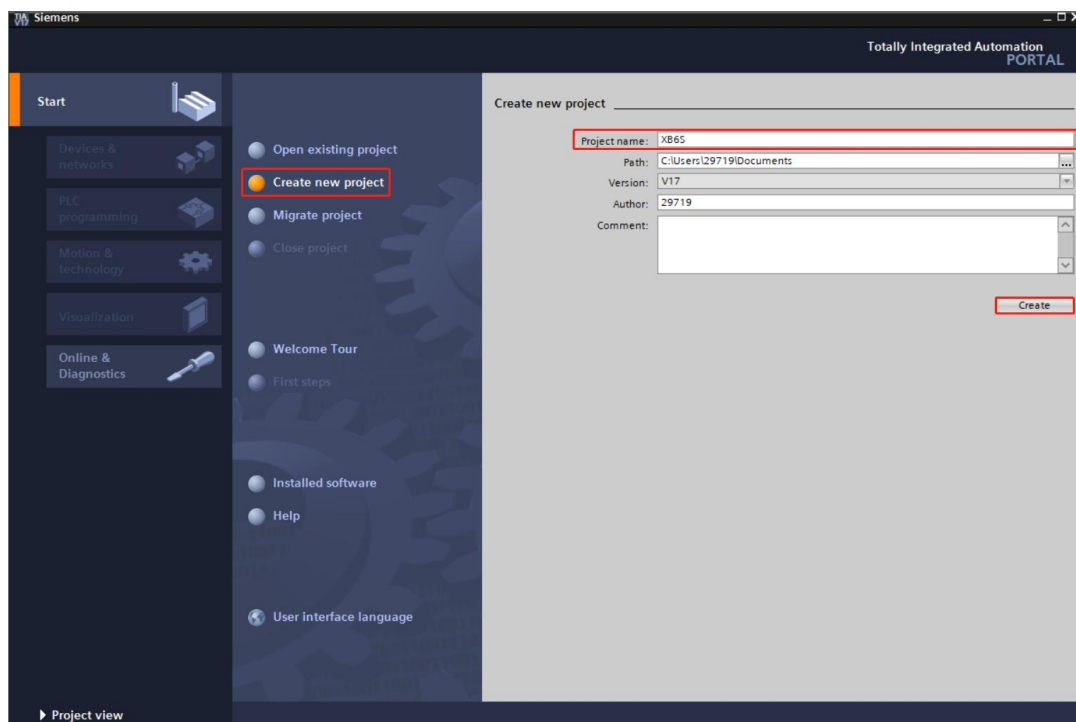
Configuration file access: <https://www.solidotech.com/documents/configfile>

#### ● Hardware configuration and wiring

Follow "[4 Installation and uninstall](#)" and "[5 Wiring](#)".

### 2、 New project

- a. Open TIA Portal V17 software, click "Create New Project", enter the information and click "Create" button, as shown below.

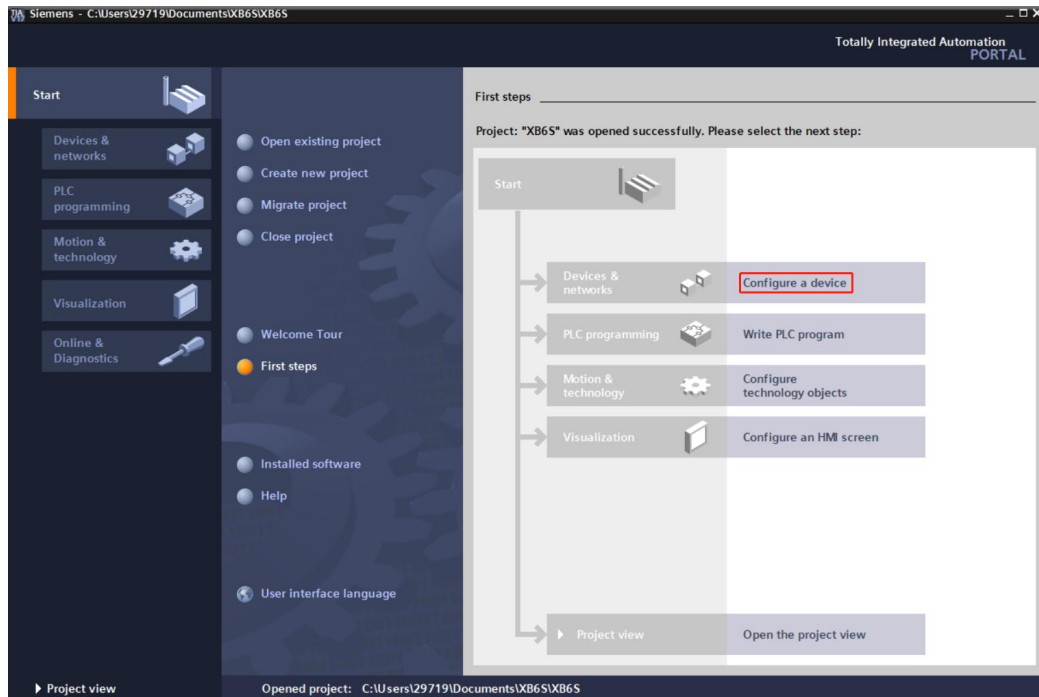


- ◆ Item name: customizable, can be left as default.
- ◆ Path: the project keeps the path, which can be left as default.
- ◆ Version: can be left as default.

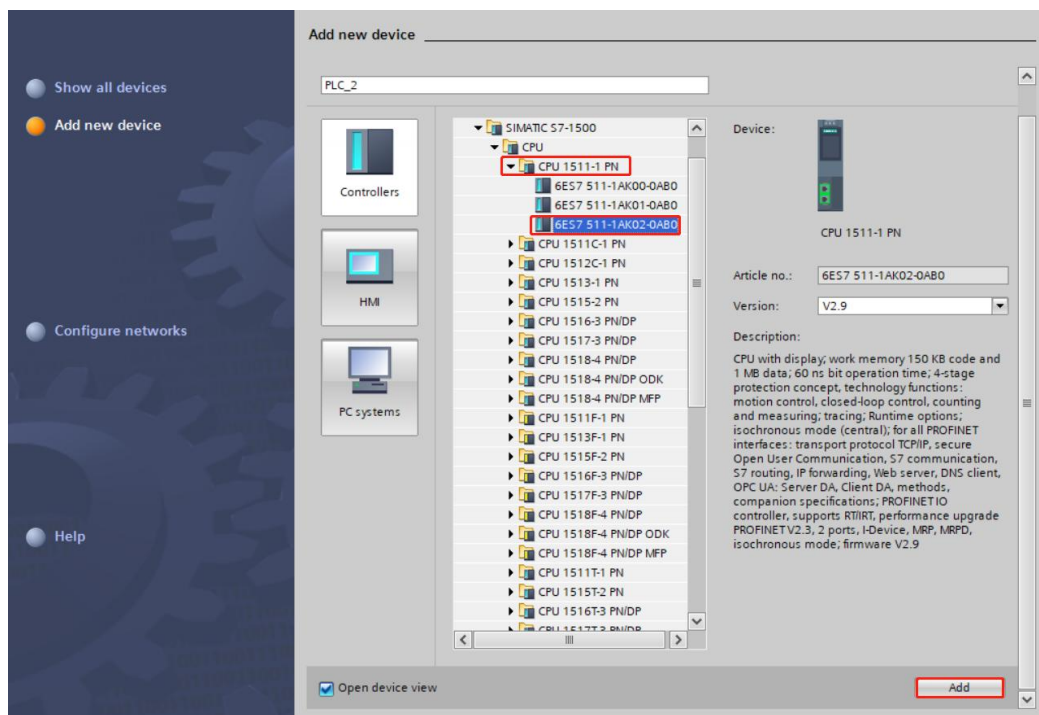
- ◆ AUTHOR: The default can be maintained.
- ◆ Note: Customizable, may not be filled in.

### 3. Adding a PLC controller

- a. Click "Configure A Device", as shown in the following figure.

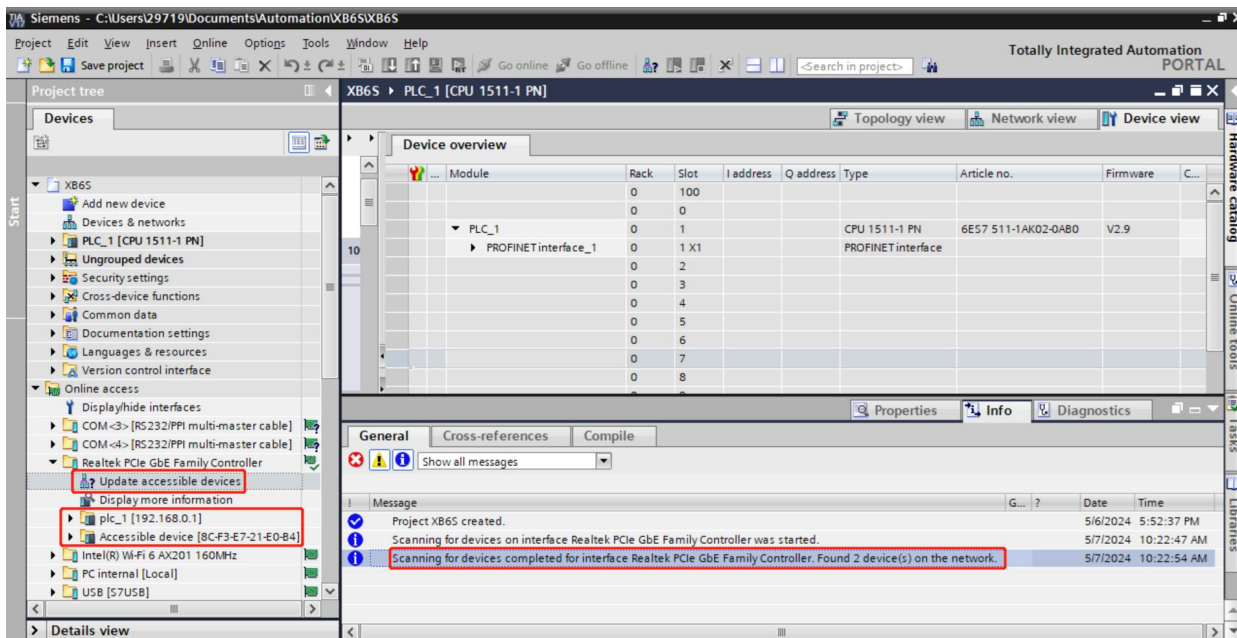


- b. Click "Add New Device", select the PLC model you are currently using, and click "Add", as shown in the following figure. After adding, you can see that the PLC has been added to the device navigation tree.



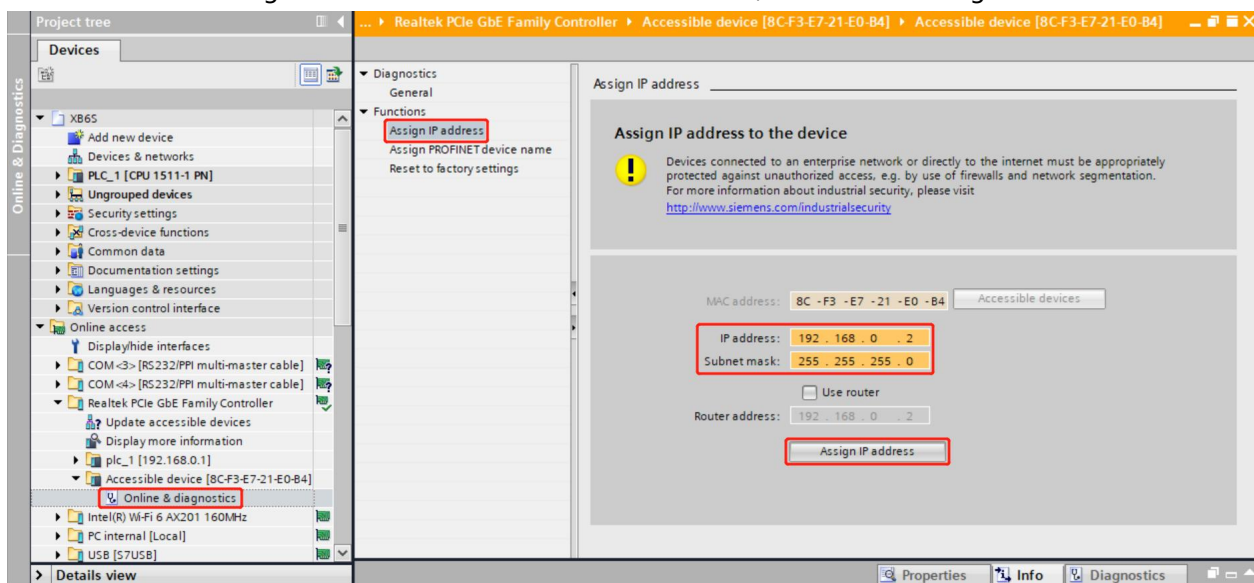
### 4. Scanning connected devices

- a. Click "Online Access -> Update Accessible Devices" in the left navigation tree, as shown in the following figure. After the update is completed, the connected slave devices are displayed as shown in the following figure.



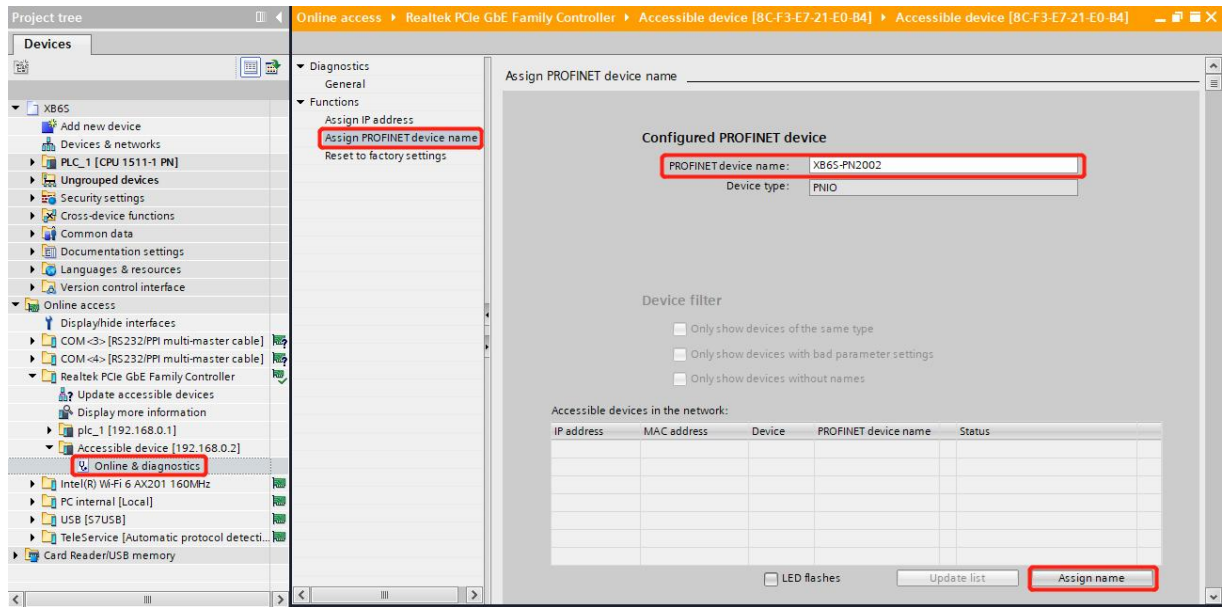
The IP address of the computer must be in the same network segment as the PLC, if not, modify the IP address of the computer and repeat the above steps.

- b. Double-click "Online and Diagnostic" under Slave Devices in the left navigation tree, and you can assign the IP address and configuration name of the current slave under the "Function" menu. Click "Assign IP Address", fill in "Subnet Mask", then "IP Address", and click "Assign IP Address" at the bottom. Click "Assign IP Address" at the bottom of the screen, as shown in the figure below.



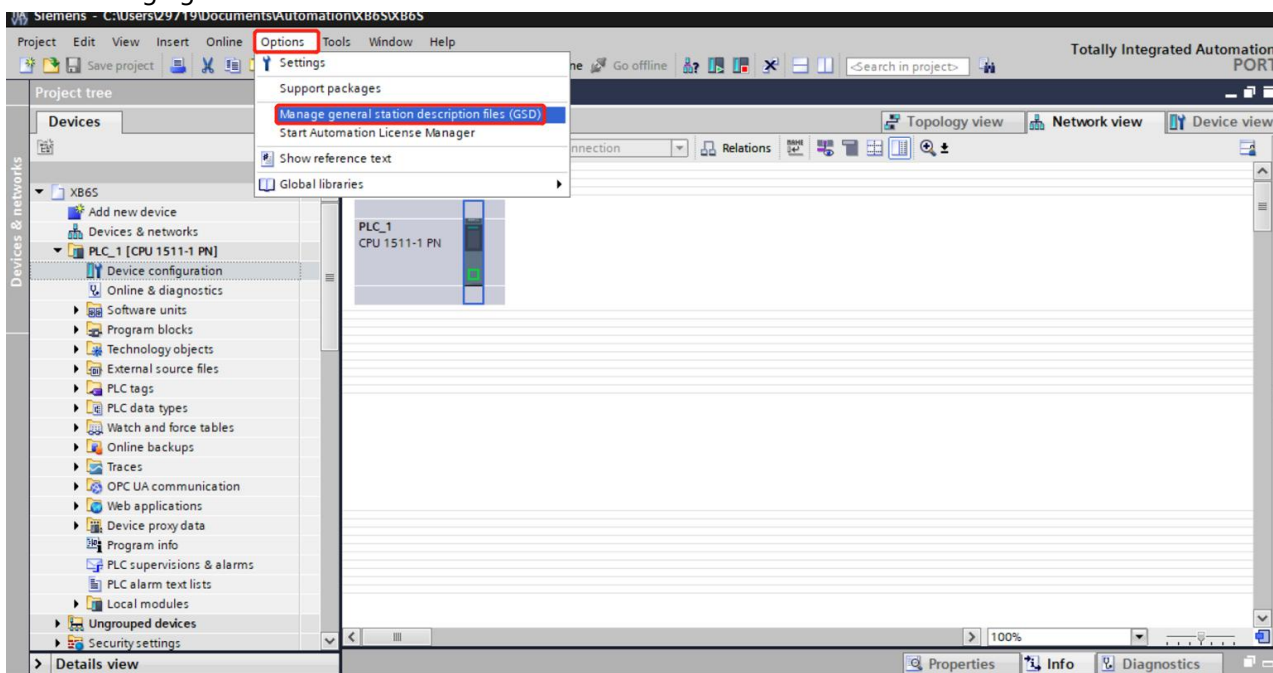


- c. Click "Assign PROFINET Device Name", fill in "PROFINET Device Name", and click "Assign Name", as shown in the following figure.

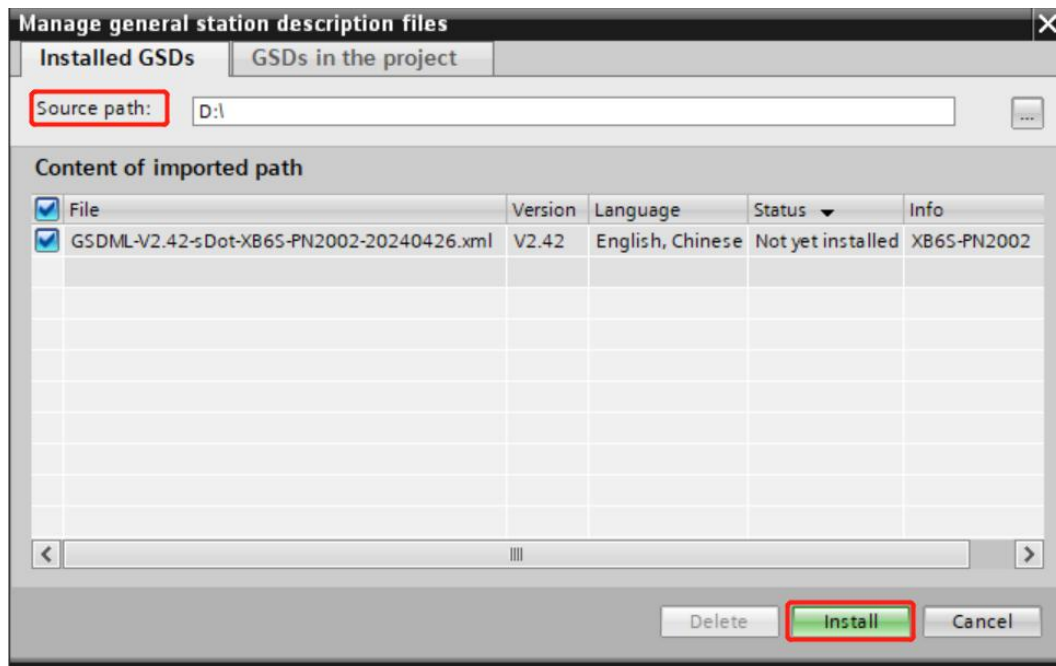


### 5. Adding a GSD Configuration File

- a. In the menu bar, select "Options -> Manage General Station Description File (GSD)" as shown in the following figure.

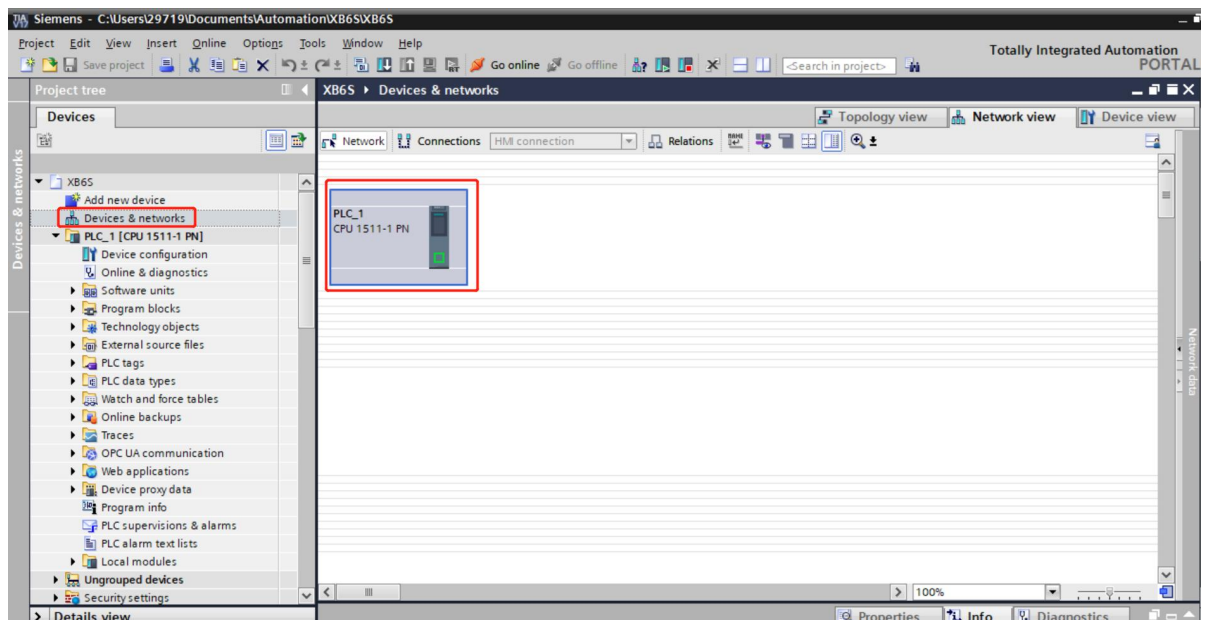


- b. Click "Source Path" to select the folder, check whether the status of the GSD file you want to add is "Not yet installed", click "Install" button if it is not installed. If it is already installed, click "Cancel" to skip the installation step, as shown in the following figure.

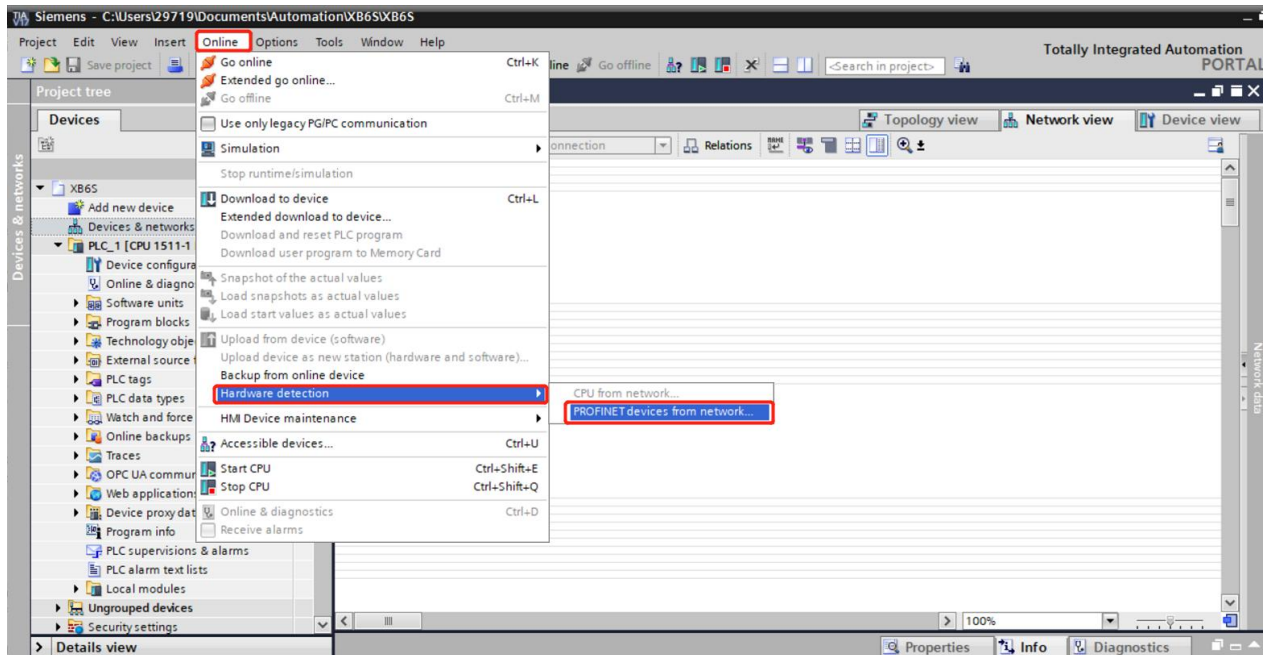


## 6. Hardware Detection Add Device

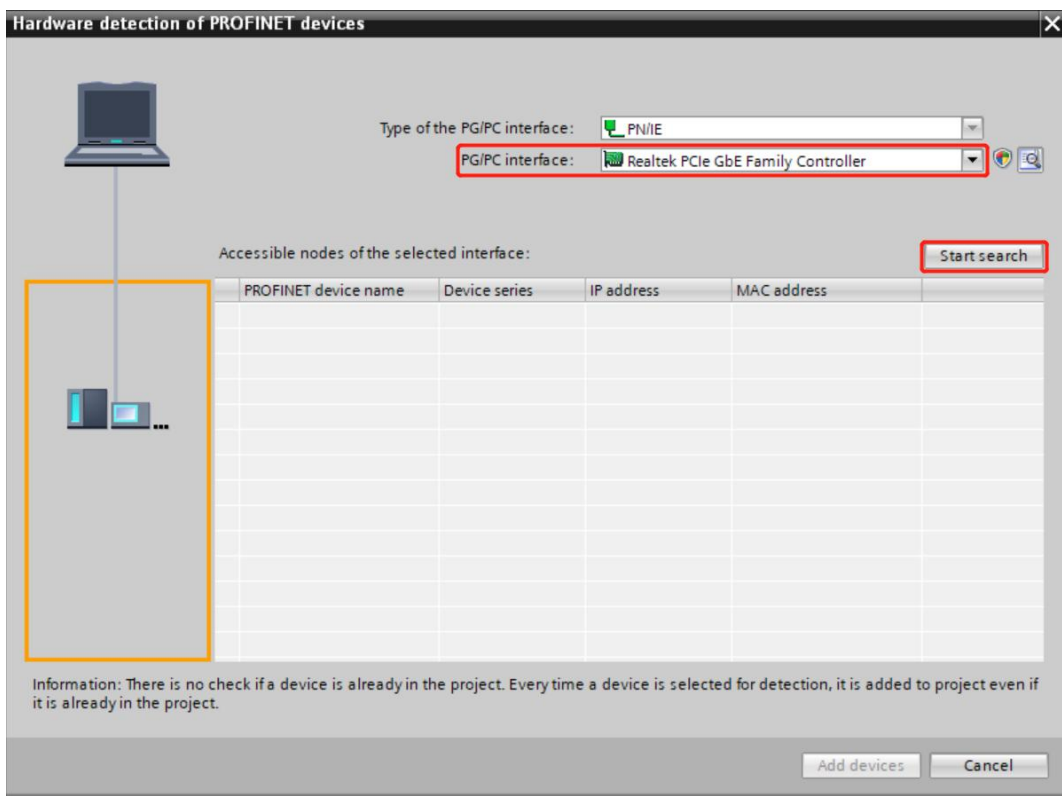
- a. Double click on "Devices and Networks" in the left navigation tree and select PLC in the Network view as shown below.



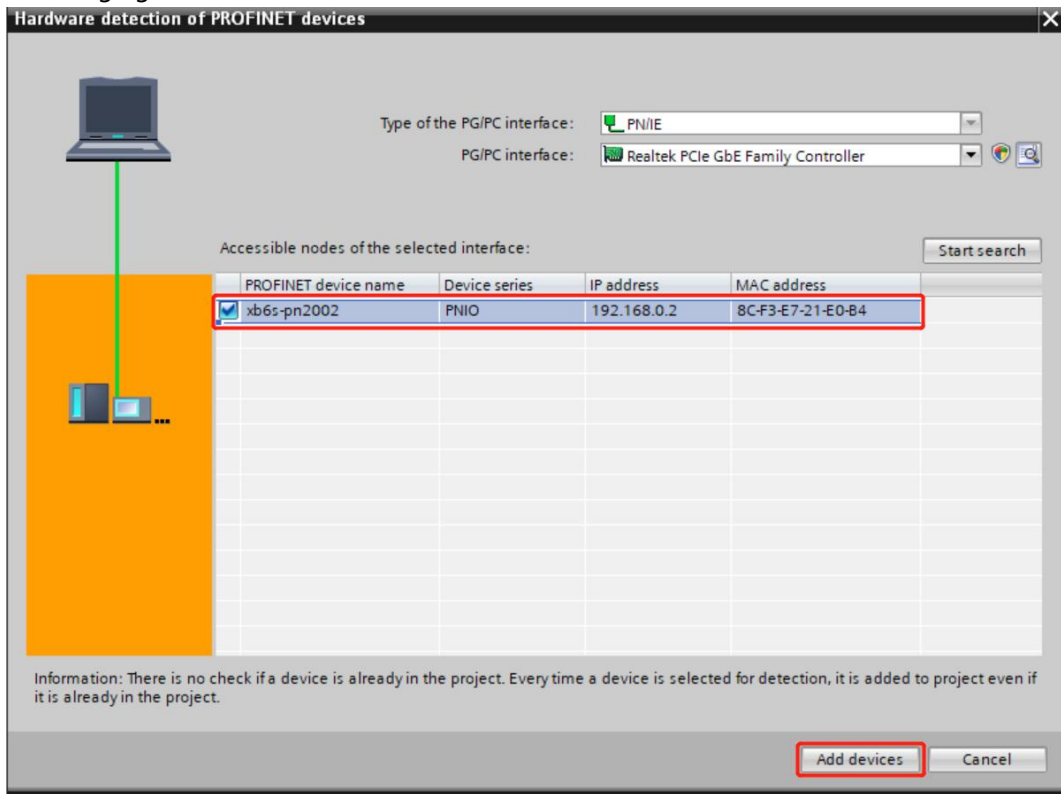
- b. In the menu bar, select "Online -> Hardware Detection -> PROFINET Devices from Network", as shown in the following figure.



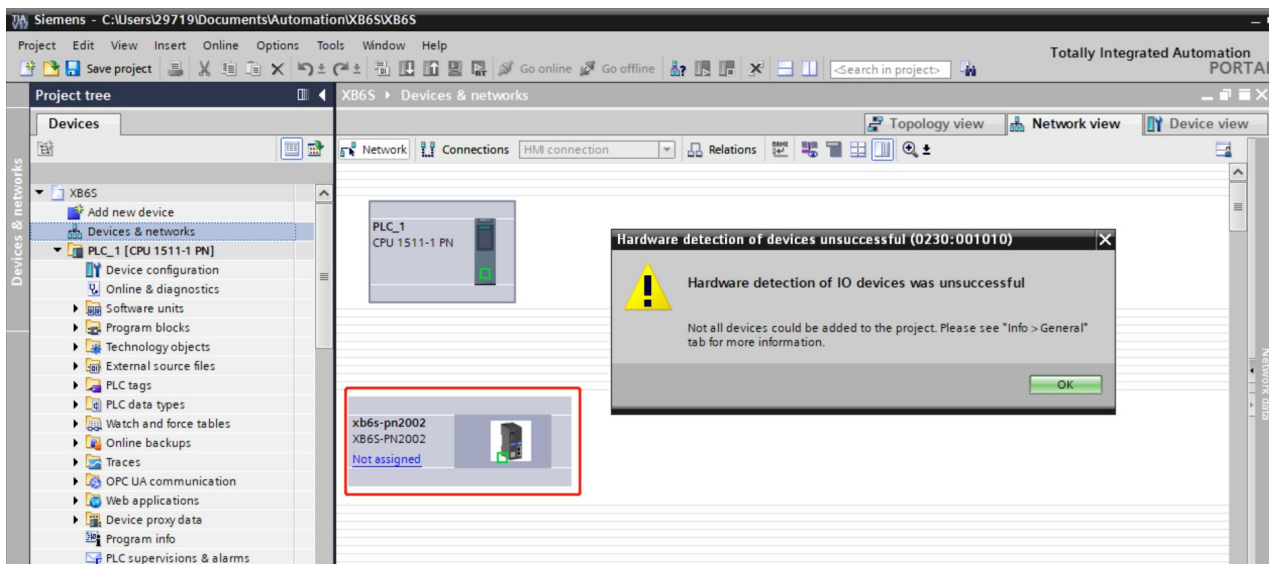
- c. The hardware detection window pops up, select the correct PG/PC interface, click "Start Search", as shown in the following figure.



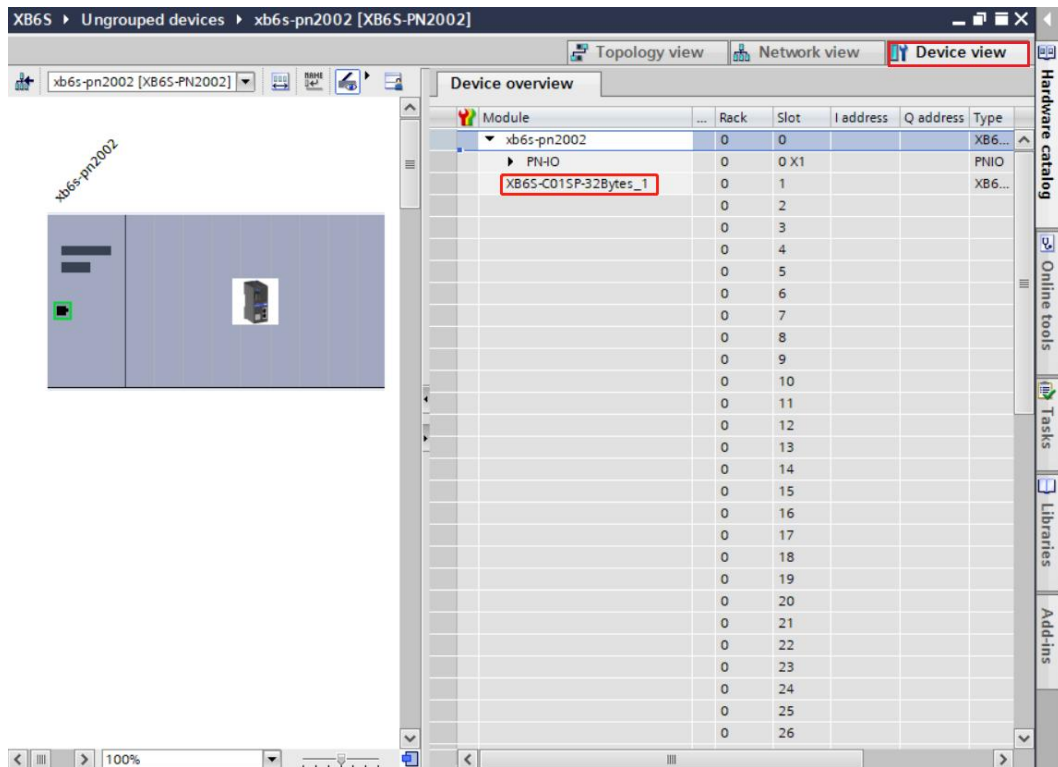
- d. Detect the coupler device in the network, check the coupler, click "Add Device", as shown in the following figure.



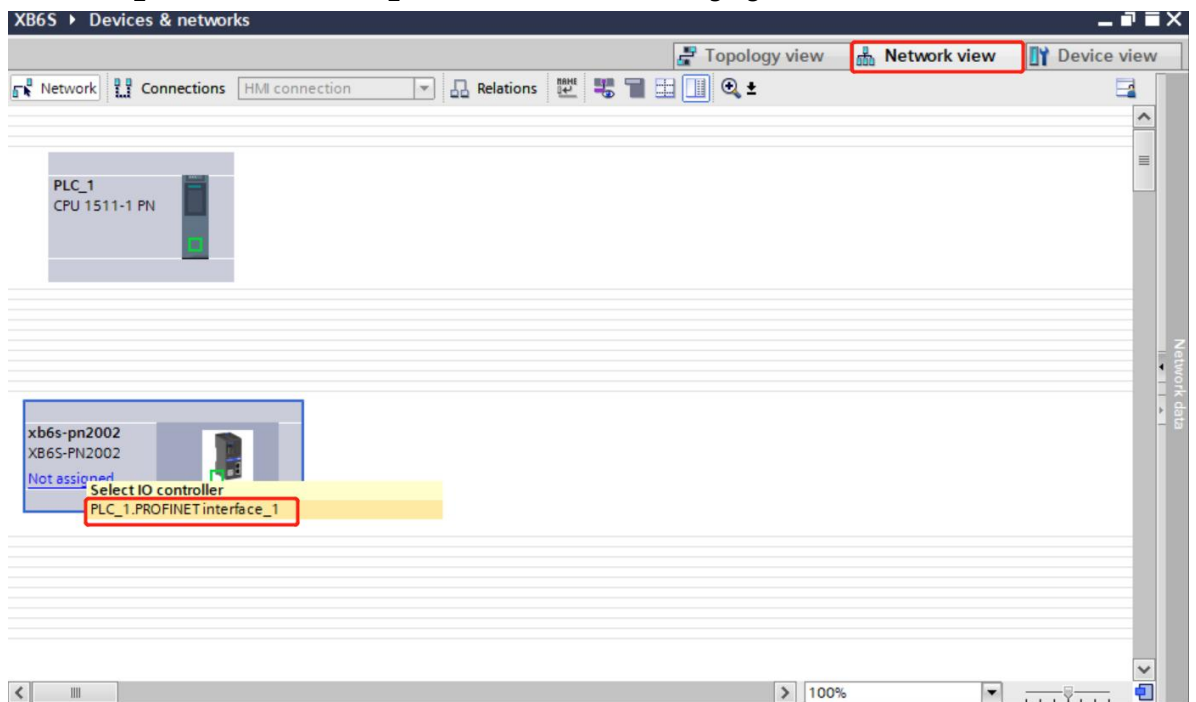
- e. After the coupler is successfully added, you can see the coupler icon in the network view as shown below.



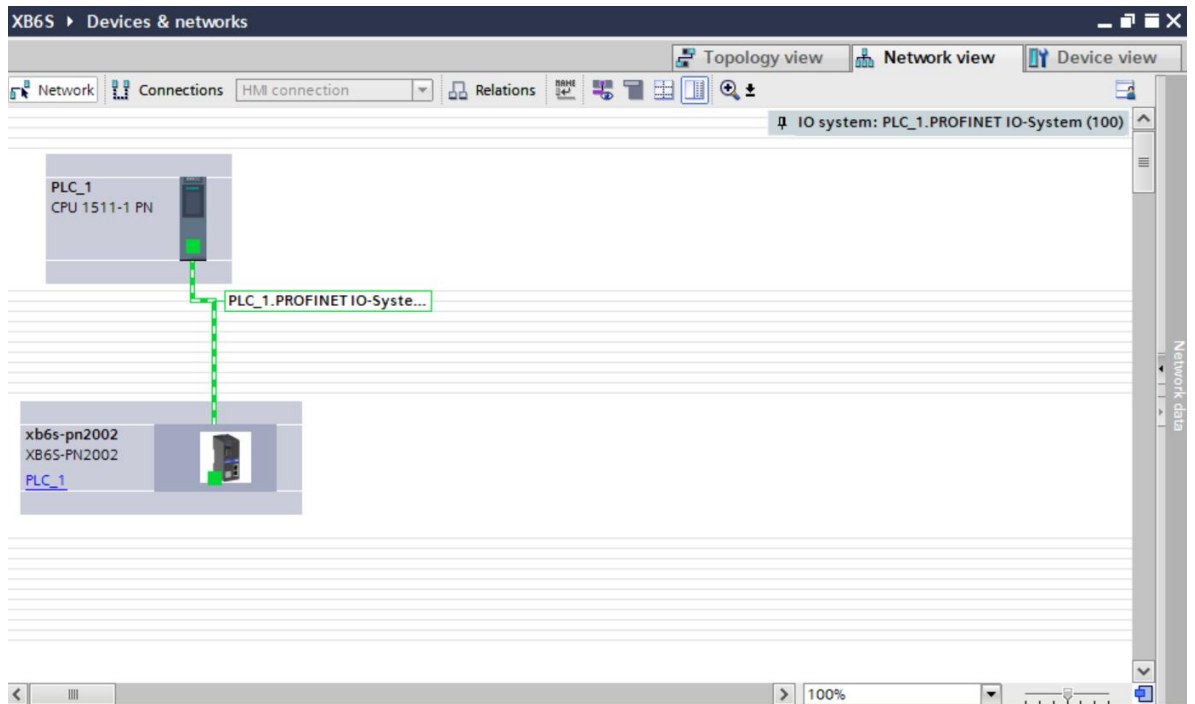
- f. Select the coupler icon and switch to the device view, you can see that the IO devices in the topology are detected to be added, as shown in the following figure.



- g. Switch to the network view, click "Unassigned (blue font)" on the coupler, i.e. slave device, and select "PLC\_1.PROFINET interface\_1" as shown in the following figure.

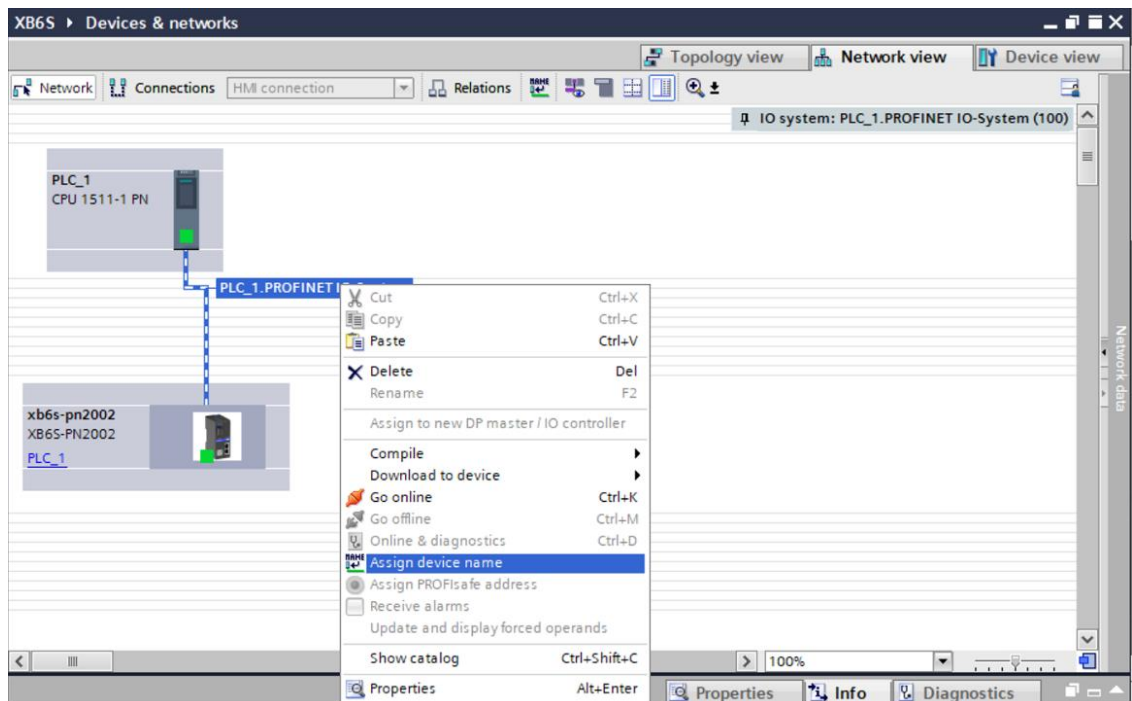


- h. When the connection is complete, it is shown in the following figure.

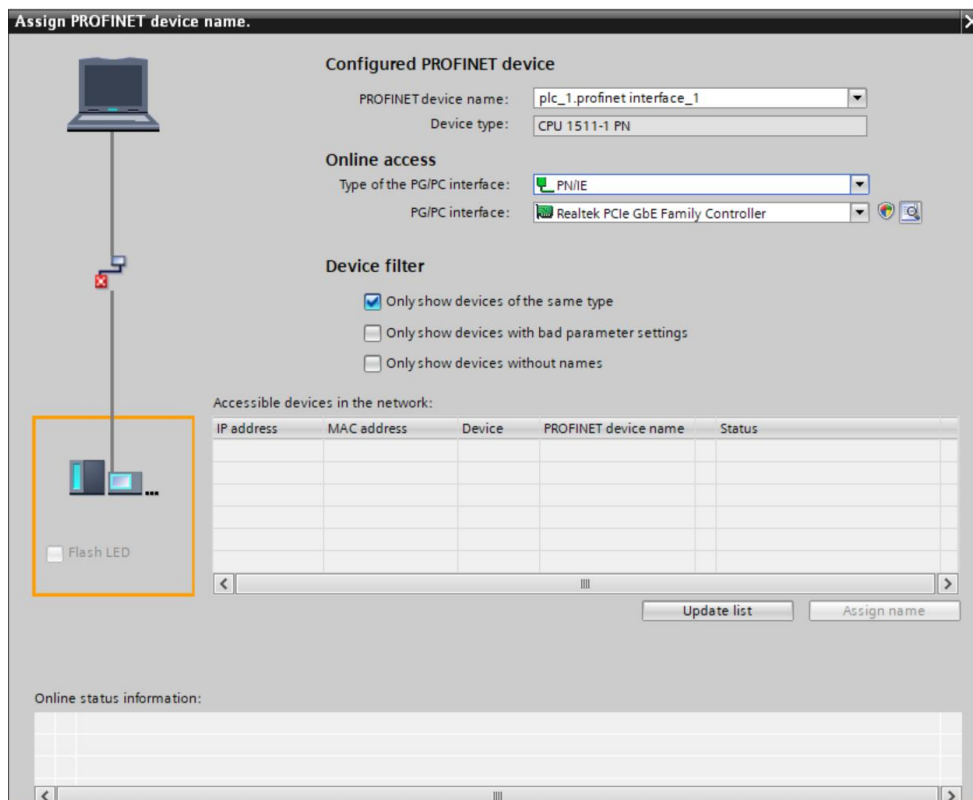


## 7. Assign device name

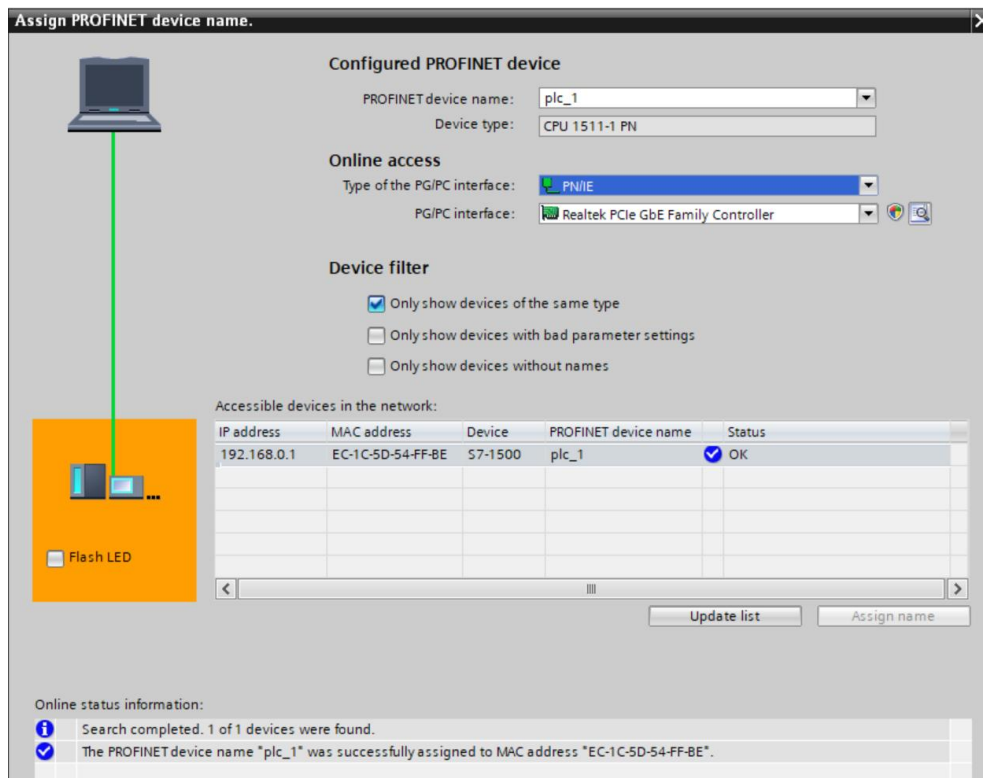
- a. In the network view, right-click on the connection line between the PLC and the coupler, and select "Assign Device Name" as shown in the following figure.



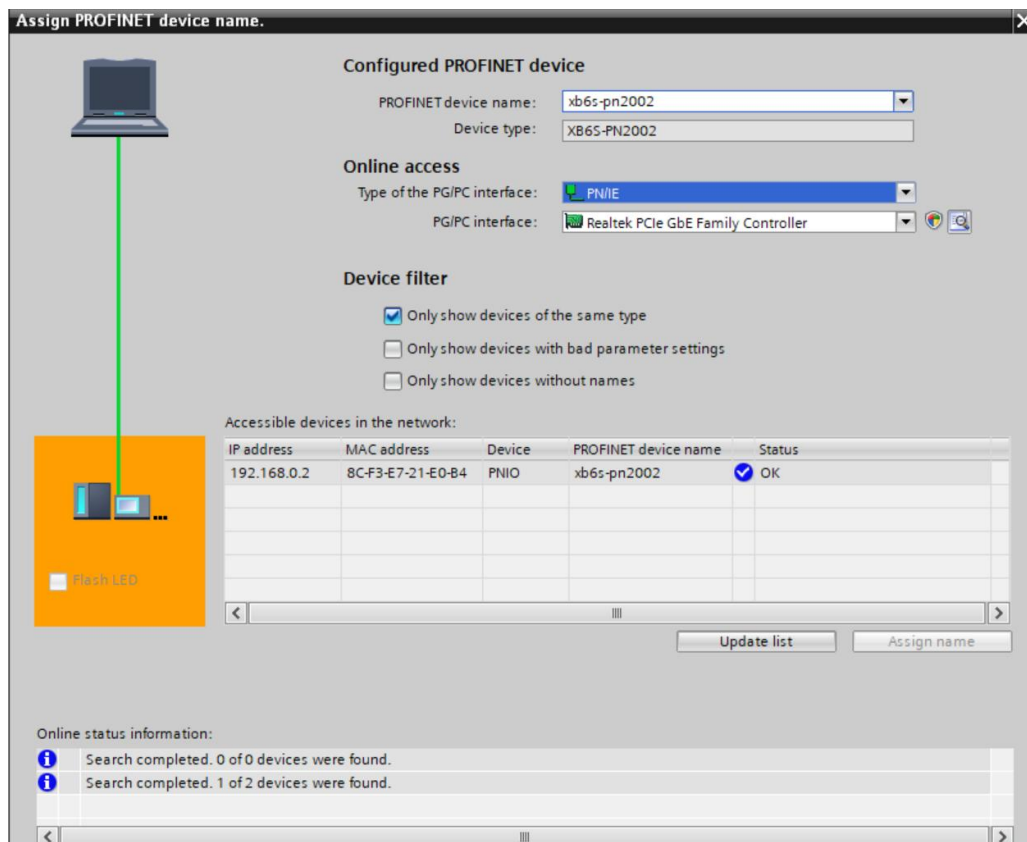
- b. The "Assign PROFINET Device Name" window pops up as shown below.



- c. Select PLC for the device name and click "Update List". After the update is completed, check whether the status of the node in "Accessible nodes in the network" is "OK". If it is not OK, select the device and click "Assign Name", as shown in the following figure.



- d. Select the coupler for the device name, click "Update List", and then assign the name in the same way after updating, as shown in the following figure.

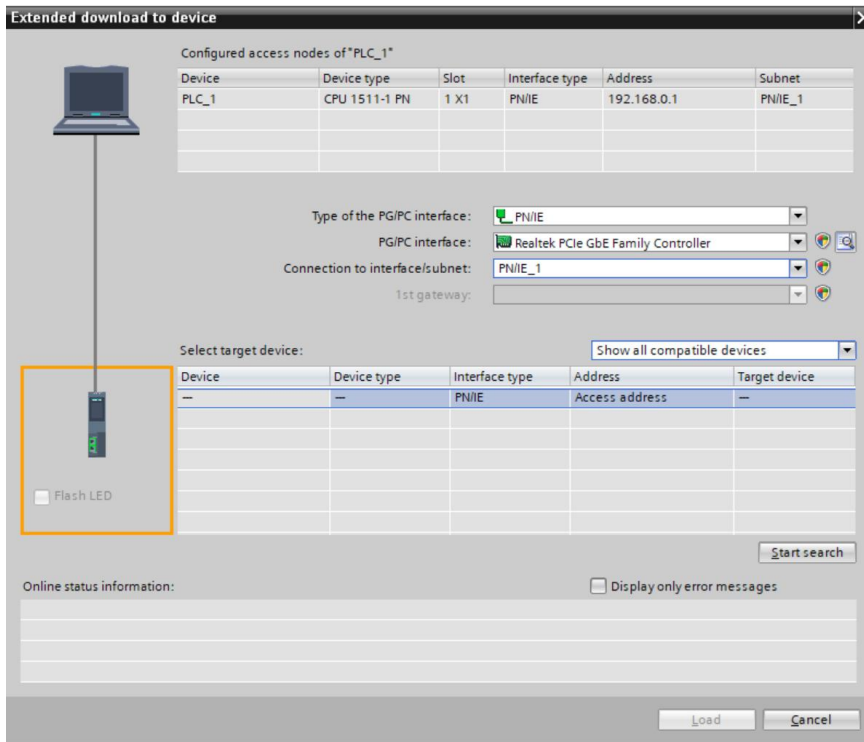


- e. Check to see if the MAC address on the module silkscreen is the same as the MAC address of the assigned device name. Click "Close".

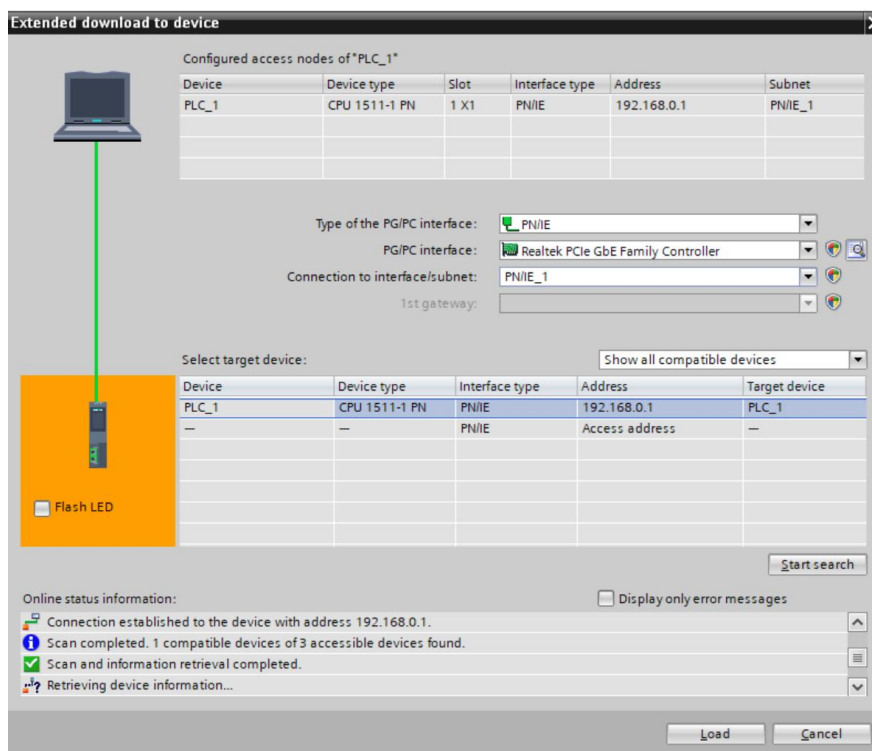


### 8. Download Configuration Structure

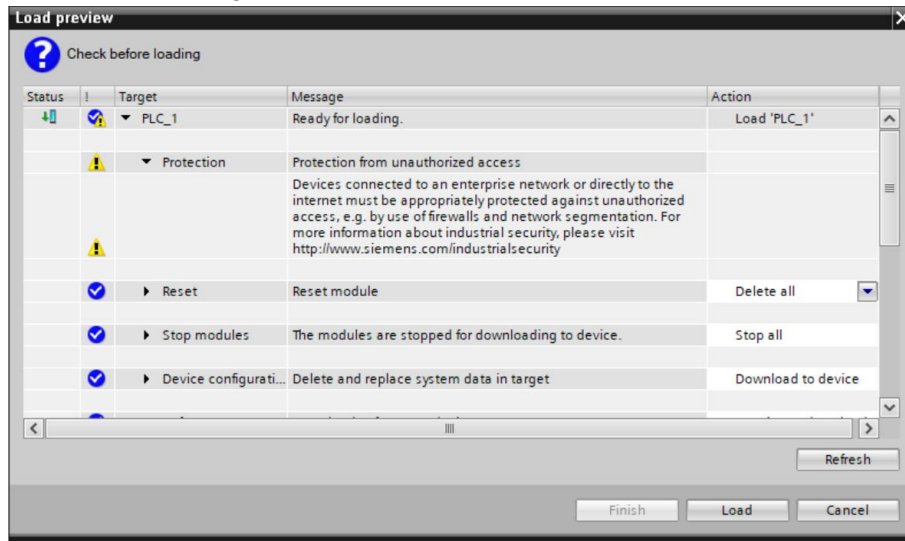
- a. In the network view, select the PLC. first click the Compile button in the menu bar, then click the Download button to download the current configuration to the PLC.
- b. In the pop-up interface of "Extended download to device", the configuration is shown in the following figure.



- c. Click the "Start Search" button as shown below.



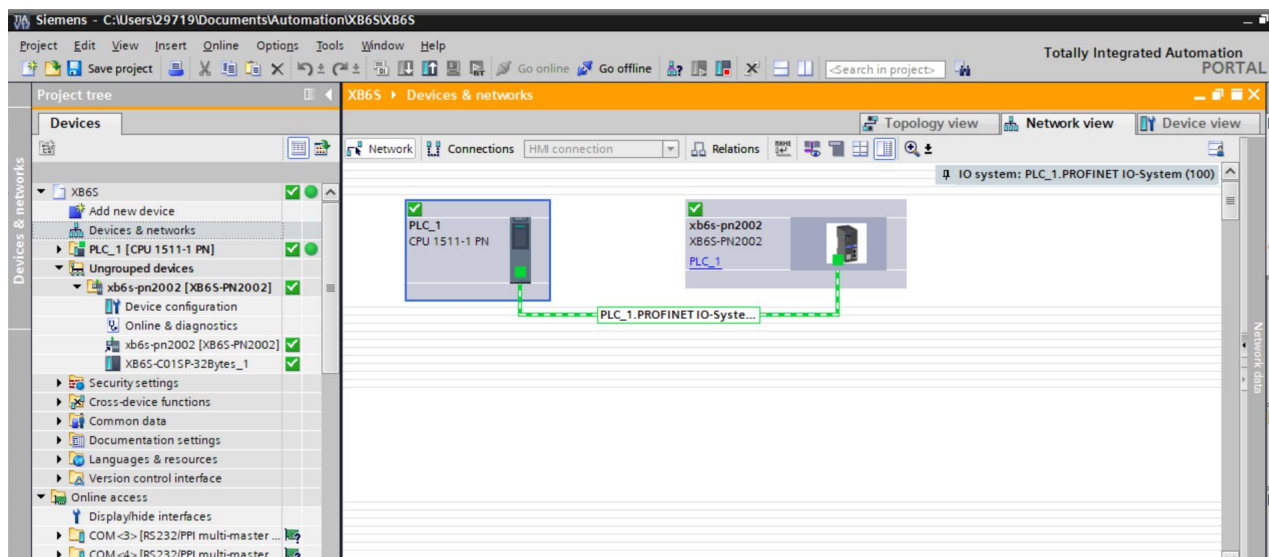
- d. Click "Load" to bring up the load preview window, as shown below.



- e. Click "Load".  
f. Click "Finish".  
g. Power the unit back up.

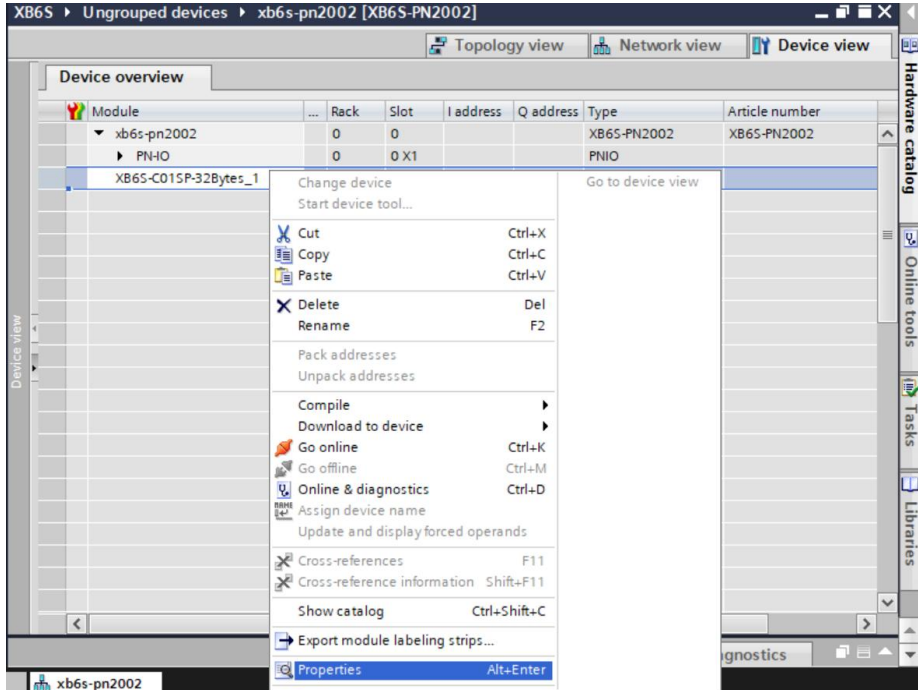
## 9. Communication Link

- a. Click the "Start CPU" button in the menu bar, and then click the "Go Online" button, the icons are green, that is, the connection is successful, as shown in the figure below.

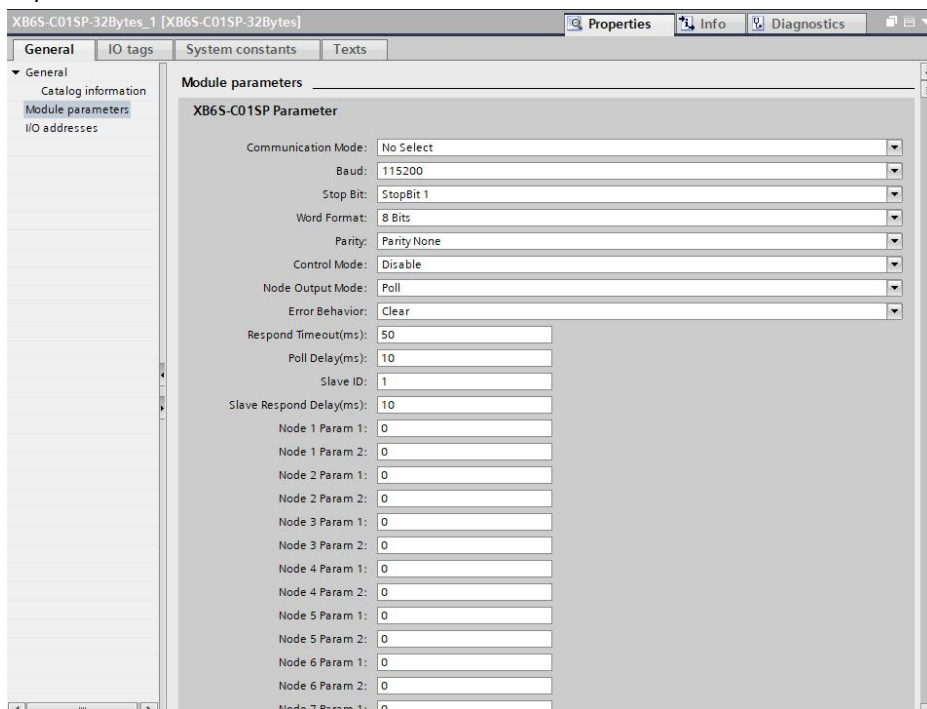


### 10. Parameter configuration

- a. In the offline state, open the "Network View", select the coupler module, switch to the device view, right-click the XB6S-C01SP-32Bytes module, and then click the "Properties" button to view and set the parameters of the module, as shown below. As shown in the figure below.

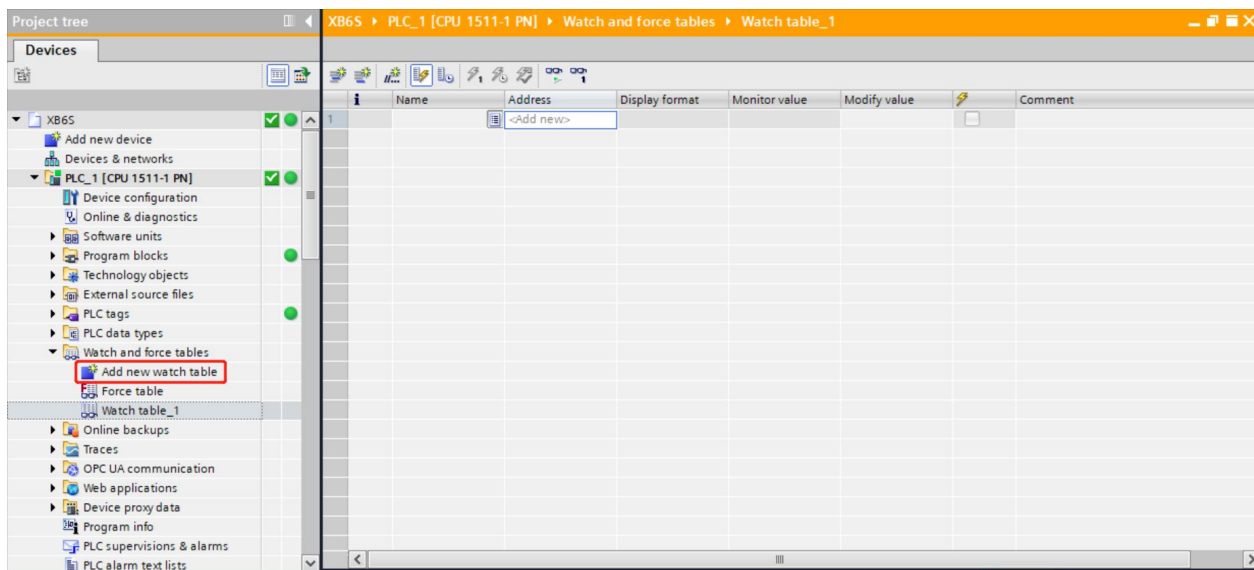


- b. In the property page, click "Module Parameters", as shown in the following figure. Parameters can be configured according to the actual use of the need to configure, after the completion of the configuration, re-download the program to the PLC, the PLC and the module need to be re-powered.

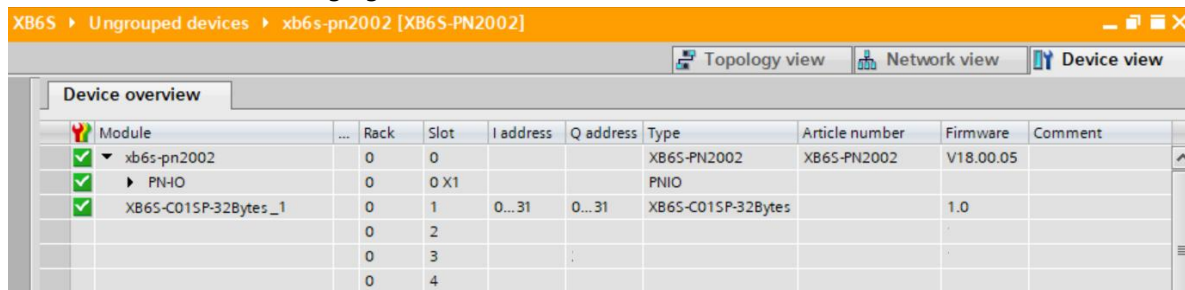




### 11、 Functional verification

- a. Expand the left side of the project navigation, select "Watch and force Table", double-click "Add New Watch Table", the system adds a new watch table, as shown in the following figure.



- b. Open the Device View to view the channel I address (channel address of the input signal) and Q address (channel address of the output signal) of each module in the device overview. For example, the "I address" of XB6S-C01SP-32Bytes module is 0 to 31, and the "Q address" is 0 to 31, as shown in the following figure.



- c. In the address cell of the monitoring table, fill in the input/output channel address, such as writing "IB0" to "IB31", "QB0" to "QB31", press "Enter", after all the information is filled in, click the  button to monitor the data. "QB0" to "QB31", press the "Enter" key, and click  button to watch the data after all the information is filled in.

## 12. RTU Master Mode Function Example

**Example 1: Verify that the module RTU master writes the values of the 2 holding registers in disable mode with a tool or device such as Modbus Slave debugging software.**

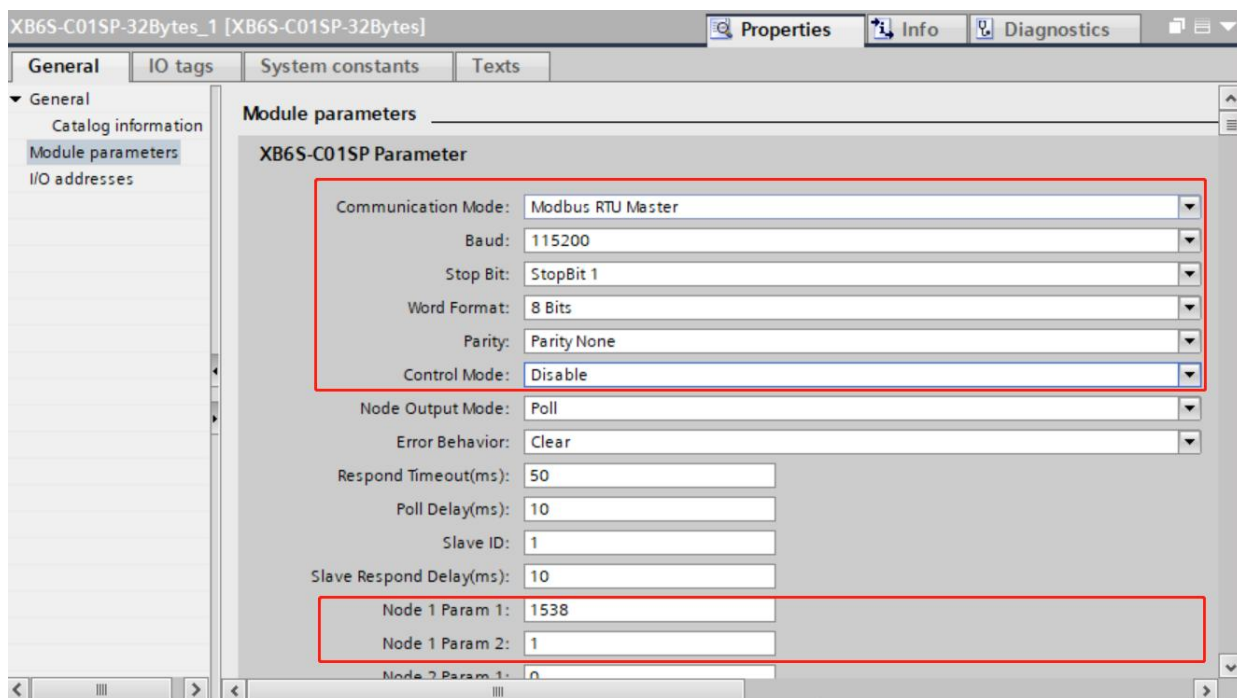
- a. Configure the configuration parameters and select Modbus RTU master mode for the communication mode as shown below.

Communication mode: Select Modbus RTU master;

Control Mode: Select Disable;

Node 1 Parameter 1: Configure 1538 (0x00000602), see [6.2.1 Modbus Master Functions](#) for configuration parameters;

Node 1 Parameter 2: Configuration 1 (0x00000001), see [6.2.1 Modbus Master Functions](#) for details on configuration parameters.



**After all parameters are configured, the program needs to be re-downloaded to the PLC, and the PLC and module need to be re-powered.**

- b. In the downstream data, the values of the 2 holding registers are written as shown below.

|    | Name | Address | Display format | Monitor value | Modify value |                                     | Comment |
|----|------|---------|----------------|---------------|--------------|-------------------------------------|---------|
| 33 |      | %QB0    | DEC+/-         | 1             | 1            | <input checked="" type="checkbox"/> |         |
| 34 |      | %QB1    | DEC+/-         | 1             | 1            | <input checked="" type="checkbox"/> |         |
| 35 |      | %QB2    | DEC+/-         | 1             | 1            | <input checked="" type="checkbox"/> |         |
| 36 |      | %QB3    | DEC+/-         | 1             | 1            | <input checked="" type="checkbox"/> |         |
| 37 |      | %QB4    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 38 |      | %QB5    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 39 |      | %QB6    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 40 |      | %QB7    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 41 |      | %QB8    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 42 |      | %QB9    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 43 |      | %QB10   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 44 |      | %QB11   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 45 |      | %QB12   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 46 |      | %QB13   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 47 |      | %QB14   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 48 |      | %QB15   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 49 |      | %QB16   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 50 |      | %QB17   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 51 |      | %QB18   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 52 |      | %QB19   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 53 |      | %QB20   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 54 |      | %QB21   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 55 |      | %QB22   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 56 |      | %QB23   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 57 |      | %QB24   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 58 |      | %QB25   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 59 |      | %QB26   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 60 |      | %QB27   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |

- c. The debugging software allows you to see the values of the 2 registers received as shown below.

|   | Alias | 00000 |
|---|-------|-------|
| 0 |       | 257   |
| 1 |       | 257   |
| 2 |       | 0     |
| 3 |       | 0     |
| 4 |       | 0     |
| 5 |       | 0     |
| 6 |       | 0     |
| 7 |       | 0     |
| 8 |       | 0     |
| 9 |       | 0     |

**Example 2: Verify that the module RTU master reads the 10 holding registers in Level mode with a tool or device such as Modbus Slave debugging software.**

- a. Configure the configuration parameters and select Modbus RTU master mode for the communication mode as shown below.

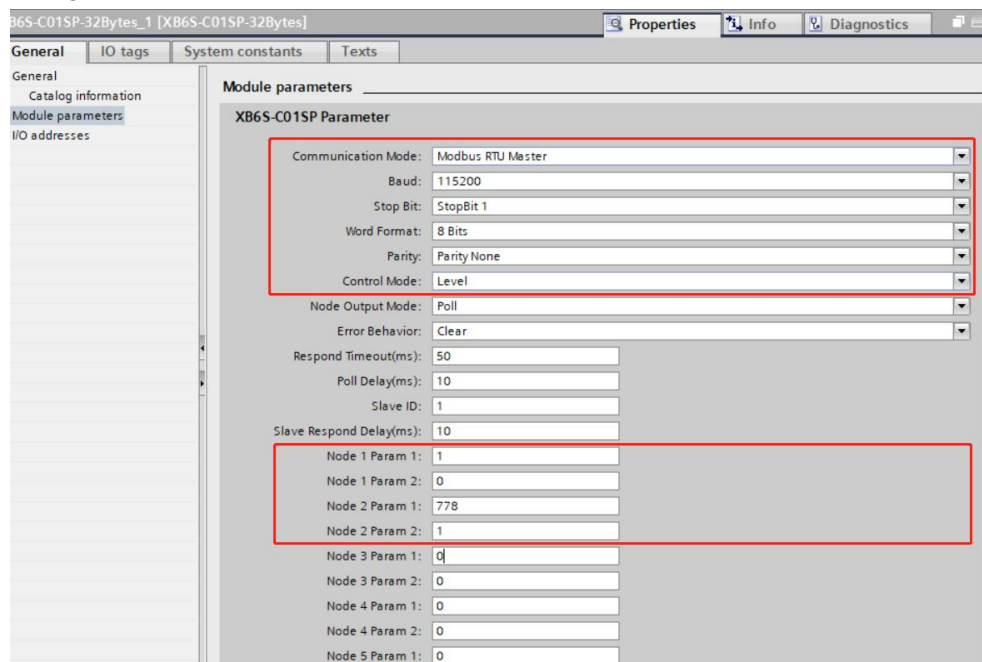
Communication mode: Select Modbus RTU master;

Control Mode: Selects level triggering;

Node 1 Parameter 1: Configuration 1 (0x00000001), see [6.2.8 Control and Status Node Codes](#) for details on configuration parameters;

Node 2 Parameter 1: Configure 778 (0x0000030A), see [6.2.1 Modbus Master Functions](#) for configuration parameters;

Node 2 Parameter 2: Configuration 1 (0x00000001), see [6.2.1 Modbus Master Functions](#) for configuration parameters.



**After all parameters are configured, the program needs to be re-downloaded to the PLC, and the PLC and module need to be re-powered.**

- b. Set the control word to 1 in the Downstream data and open the debugging software to send the data as shown below.

|    | Name | Address | Display format | Monitor value | Modify value |                                     | Comment |
|----|------|---------|----------------|---------------|--------------|-------------------------------------|---------|
| 33 |      | %QB0    | DEC+/-         | 1             | 1            | <input checked="" type="checkbox"/> | !       |
| 34 |      | %QB1    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 35 |      | %QB2    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 36 |      | %QB3    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 37 |      | %QB4    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 38 |      | %QB5    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 39 |      | %QB6    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 40 |      | %QB7    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 41 |      | %QB8    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 42 |      | %QB9    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 43 |      | %QB10   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 44 |      | %QB11   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 45 |      | %QB12   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 46 |      | %QB13   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 47 |      | %QB14   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 48 |      | %QB15   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 49 |      | %QB16   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 50 |      | %QB17   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 51 |      | %QB18   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 52 |      | %QB19   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 53 |      | %QB20   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 54 |      | %QB21   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 55 |      | %QB22   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 56 |      | %QB23   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 57 |      | %QB24   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 58 |      | %QB25   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 59 |      | %QB26   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |

- c. You can see the received data in the Upstream data as shown below.

|  | Name | Address | Display format | Monitor value | Modify value |                          | Comment |
|--|------|---------|----------------|---------------|--------------|--------------------------|---------|
|  |      | %IB0    | DEC+/-         | 5             |              | <input type="checkbox"/> |         |
|  |      | %IB1    | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB2    | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB3    | DEC+/-         | 255           |              | <input type="checkbox"/> |         |
|  |      | %IB4    | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB5    | DEC+/-         | 255           |              | <input type="checkbox"/> |         |
|  |      | %IB6    | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB7    | DEC+/-         | 255           |              | <input type="checkbox"/> |         |
|  |      | %IB8    | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB9    | DEC+/-         | 255           |              | <input type="checkbox"/> |         |
|  |      | %IB10   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB11   | DEC+/-         | 255           |              | <input type="checkbox"/> |         |
|  |      | %IB12   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB13   | DEC+/-         | 255           |              | <input type="checkbox"/> |         |
|  |      | %IB14   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB15   | DEC+/-         | 255           |              | <input type="checkbox"/> |         |
|  |      | %IB16   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB17   | DEC+/-         | 255           |              | <input type="checkbox"/> |         |
|  |      | %IB18   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB19   | DEC+/-         | 255           |              | <input type="checkbox"/> |         |
|  |      | %IB20   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB21   | DEC+/-         | 255           |              | <input type="checkbox"/> |         |
|  |      | %IB22   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB23   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB24   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB25   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB26   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
|  |      | %IB27   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |



### 13. Example of Freeport\_Input function

**Example: Receive an 8-byte data in Level mode using Freeport\_Input through a tool or device such as a serial port debugging assistant.**

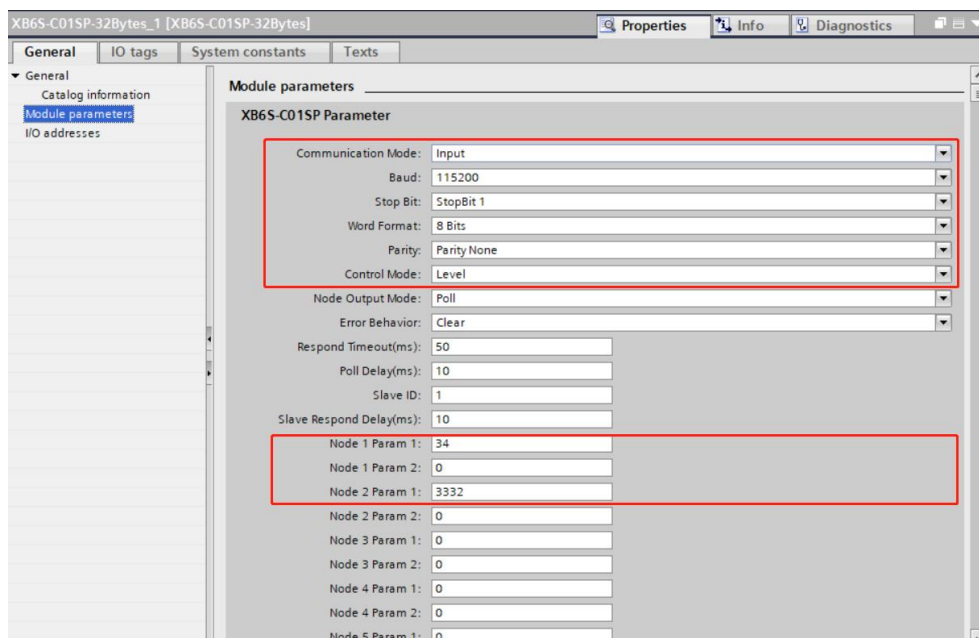
- a. Configure the configuration parameters and select the free port pure input mode for the communication mode as shown below.

Communication Mode: Select Input mode;

Control Mode: Selects Level;

Node 1 Parameter 1: Configuration 34 (0x00000022), see [6.2.8 Control and Status Node Codes](#) for details on configuration parameters;

Node 2 Parameter 1: Configure 3332 (0x00000D04), see [6.2.6 Freeport Freeport Functions](#) for details on configuration parameters.



**After all parameters are configured, the program needs to be re-downloaded to the PLC, and the PLC and module need to be re-powered.**

- b. Set the control word to 1 in the downstream data and open the serial port debugging assistant to send data as shown below.

|    | Name | Address | Display format | Monitor value | Modify value |                                     | Comment |
|----|------|---------|----------------|---------------|--------------|-------------------------------------|---------|
| 33 |      | %QB0    | DEC+/-         | 1             | 1            | <input checked="" type="checkbox"/> |         |
| 34 |      | %QB1    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 35 |      | %QB2    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 36 |      | %QB3    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 37 |      | %QB4    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 38 |      | %QB5    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 39 |      | %QB6    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 40 |      | %QB7    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 41 |      | %QB8    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 42 |      | %QB9    | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 43 |      | %QB10   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 44 |      | %QB11   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 45 |      | %QB12   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 46 |      | %QB13   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 47 |      | %QB14   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 48 |      | %QB15   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 49 |      | %QB16   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 50 |      | %QB17   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 51 |      | %QB18   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 52 |      | %QB19   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 53 |      | %QB20   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 54 |      | %QB21   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 55 |      | %QB22   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 56 |      | %QB23   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 57 |      | %QB24   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 58 |      | %QB25   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |
| 59 |      | %QB26   | DEC+/-         | 0             |              | <input type="checkbox"/>            |         |

- c. The Upstream data can be seen in the received data length of 8, the received data count of 1, and the received 8 bytes of data (IB6~IB13), as shown in the following figure.

|    | Name | Address | Display format | Monitor value | Modify value |                          | Comment |
|----|------|---------|----------------|---------------|--------------|--------------------------|---------|
| 1  |      | %IB0    | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 2  |      | %IB1    | DEC+/-         | 8             |              | <input type="checkbox"/> |         |
| 3  |      | %IB2    | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 4  |      | %IB3    | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 5  |      | %IB4    | DEC+/-         | 1             |              | <input type="checkbox"/> |         |
| 6  |      | %IB5    | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 7  |      | %IB6    | DEC+/-         | 1             |              | <input type="checkbox"/> |         |
| 8  |      | %IB7    | DEC+/-         | 2             |              | <input type="checkbox"/> |         |
| 9  |      | %IB8    | DEC+/-         | 3             |              | <input type="checkbox"/> |         |
| 10 |      | %IB9    | DEC+/-         | 4             |              | <input type="checkbox"/> |         |
| 11 |      | %IB10   | DEC+/-         | 5             |              | <input type="checkbox"/> |         |
| 12 |      | %IB11   | DEC+/-         | 1             |              | <input type="checkbox"/> |         |
| 13 |      | %IB12   | DEC+/-         | 2             |              | <input type="checkbox"/> |         |
| 14 |      | %IB13   | DEC+/-         | 3             |              | <input type="checkbox"/> |         |
| 15 |      | %IB14   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 16 |      | %IB15   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 17 |      | %IB16   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 18 |      | %IB17   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 19 |      | %IB18   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 20 |      | %IB19   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 21 |      | %IB20   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 22 |      | %IB21   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 23 |      | %IB22   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 24 |      | %IB23   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 25 |      | %IB24   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 26 |      | %IB25   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 27 |      | %IB26   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |
| 28 |      | %IB27   | DEC+/-         | 0             |              | <input type="checkbox"/> |         |