



XB6S-C01SP

串行通讯模块

用户手册

s'Dot

南京实点电子科技有限公司

版权所有 © 南京实点电子科技有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明

 和其它实点商标均为南京实点电子科技有限公司的商标。

本文档提及的其它所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受实点公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，实点公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

南京实点电子科技有限公司

地址：江苏省南京市江宁区胜利路 91 号昂鹰大厦 11 楼

邮编：211106

电话：4007788929

网址：<http://www.solidotech.com>

目 录

1	产品概述.....	1
1.1	产品简介.....	1
1.2	产品特性.....	1
2	产品参数.....	2
2.1	通用参数.....	2
3	面板.....	3
3.1	面板结构.....	3
3.2	指示灯功能.....	4
4	安装和拆卸.....	5
4.1	外形尺寸.....	5
4.2	安装指南.....	6
4.3	安装拆卸步骤.....	9
4.4	安装拆卸示意图.....	10
5	接线.....	17
5.1	接线图.....	17
5.2	接线端子定义.....	18
6	使用.....	19
6.1	配置参数定义.....	19
6.2	节点代码参数.....	21
6.2.1	Modbus 主站功能.....	21
6.2.2	Modbus 主站功能节点代码.....	21
6.2.3	Modbus 从站功能.....	24
6.2.4	Modbus 从站功能节点代码.....	24
6.2.5	字节交换.....	27
6.2.6	Freeport 自由口功能.....	27
6.2.7	Freeport 功能节点代码.....	27
6.2.8	控制与状态节点代码.....	28
6.3	过程数据.....	29
6.3.1	Modbus 主站过程数据.....	30
6.3.2	Modbus 从站过程数据.....	31

6.3.3	Freeport_Request 过程数据	32
6.3.4	Freeport_Input 过程数据	33
6.3.5	Freeport_Output 过程数据	34
6.3.6	Freeport_Input_Output 过程数据	35
6.3.7	Modbus 告警码	36
6.4	模块组态说明	37
6.4.1	在 TwinCAT3 软件环境下的应用	37
6.4.2	在 Sysmac Studio 软件环境下的应用	54
6.4.3	在 TIA Portal V17 软件环境下的应用	69

1 产品概述

1.1 产品简介

XB6S-C01SP 是插片式 1 通道串行通讯模块，采用 X-bus 底部总线，适配本司 XB6S 系列耦合器模块，可实现 Modbus 主从站、Freeport 串行通讯功能，模块占用空间小，数据交互处理简单，能够满足不同应用场景的串行通讯需求。

1.2 产品特性

- 支持多种通讯模式
可设置 MRM/MRS/MAM/MAS/FP_Request/FP_Input_Output/FP_Input/FP_Output 八种模式（详情见 [6.1 配置参数](#)）。
- 支持三种通讯接口
RS485/RS422/RS232 三种接口。
- 支持三种通讯协议
Modbus RTU/ Modbus ASCII/Freeport。
- 体积小
结构紧凑，占用空间小。
- 易诊断
创新的通道指示灯设计，紧贴通道，通道状态一目了然，检测、维护方便。
- 易组态
组态配置简单，支持主流主站。
- 易安装
DIN 35 mm 标准导轨安装
采用弹片式接线端子，配线方便快捷。

2 产品参数

2.1 通用参数

接口参数	
产品型号	XB6S-C01SP
总线协议	X-bus
总线输入电源额定电压	5VDC (4.5V~5.5V)
额定电流消耗	230mA
功耗	1.15W
技术参数	
通道数	1 通道
通讯接口类型 ^[1]	RS232、RS485、RS422
通讯协议	Modbus RTU、Modbus ASCII、Freeport
波特率	1200bps~115200bps
重量	90g
尺寸	106.4×25.7×72.3mm
接线方式	免螺丝快速插头
安装方式	DIN 35mm 导轨安装
工作温度	-20°C~+60°C
存储温度	-40°C~+80°C
相对湿度	95%，无冷凝
防护等级	IP20

注[1]: RS232、RS485、RS422 接口同时只能使用一种。

3 面板

3.1 面板结构

产品各部位名称



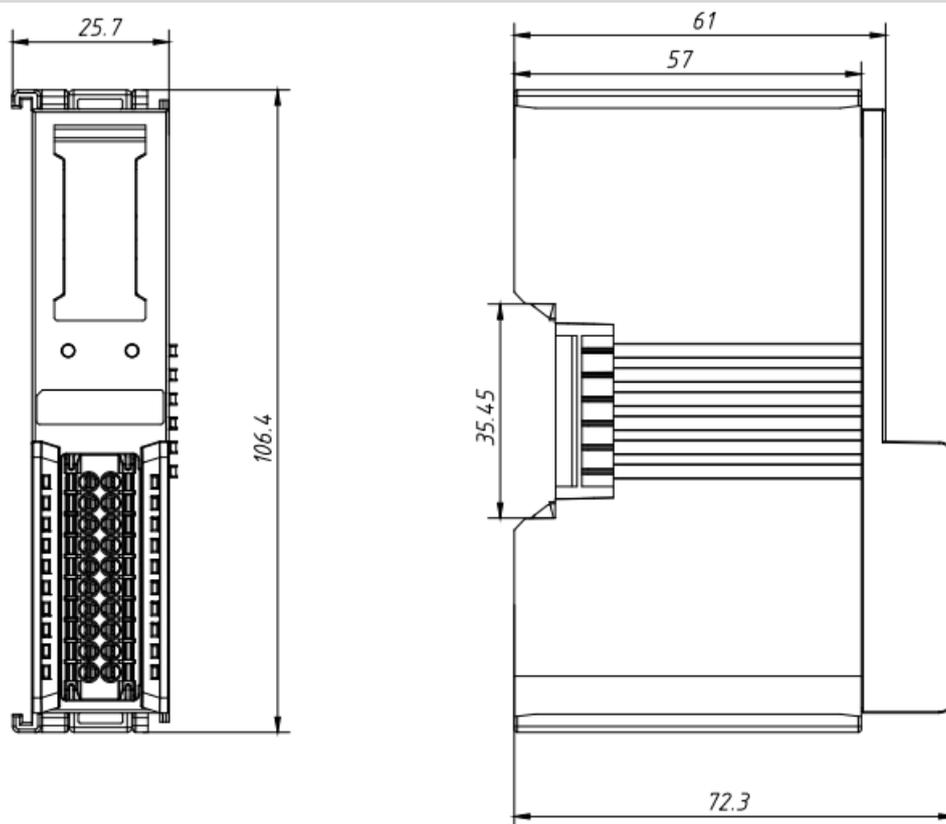
3.2 指示灯功能

名称	标识	颜色	状态	状态描述
电源指示灯	PWR	绿色	常亮	电源供电正常
			熄灭	产品未上电或电源供电异常
通信指示灯	SYS	绿色	常亮	系统运行正常
			闪烁 1Hz	无业务数据交互, 等待建立业务数据交互
			闪烁 10Hz	固件升级
			熄灭	系统未工作
输入通道指示灯	0 (右侧)	绿色	闪烁	通道有数据接收
			熄灭	通道无数据接收
输出通道指示灯	0 (左侧)	绿色	闪烁	通道有数据发送
			熄灭	通道无数据发送

4 安装和拆卸

4.1 外形尺寸

外形规格 (单位 mm)



4.2 安装指南

安装\拆卸注意事项

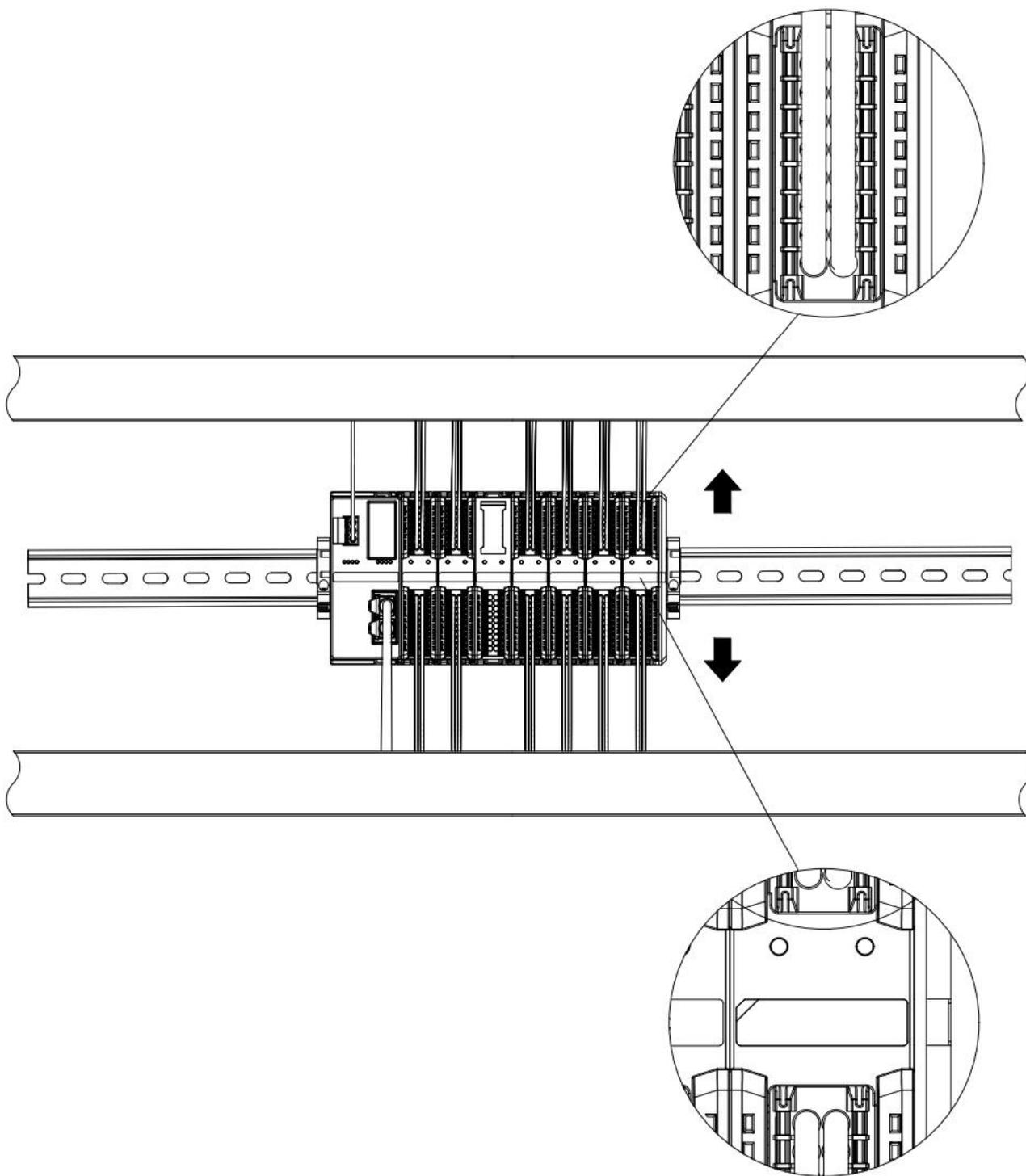
- 模块防护等级为 IP20，模块需在机柜内安装，室内使用。
- 确保机柜有良好的通风措施（如机柜加装排风扇）。
- 请勿将本设备安装在可能引起过热的设备旁边或者上方。
- 务必将模块竖直安装在固定导轨上，并保持周围空气流通（模块上下至少有 50mm 的空气流通空间）。
- 模块安装后，务必在两端安装导轨固定件将模块固定。
- 安装\拆卸务必在切断电源的状态下进行。
- 模块安装后，建议按照上下走线的方式进行接线和布线。



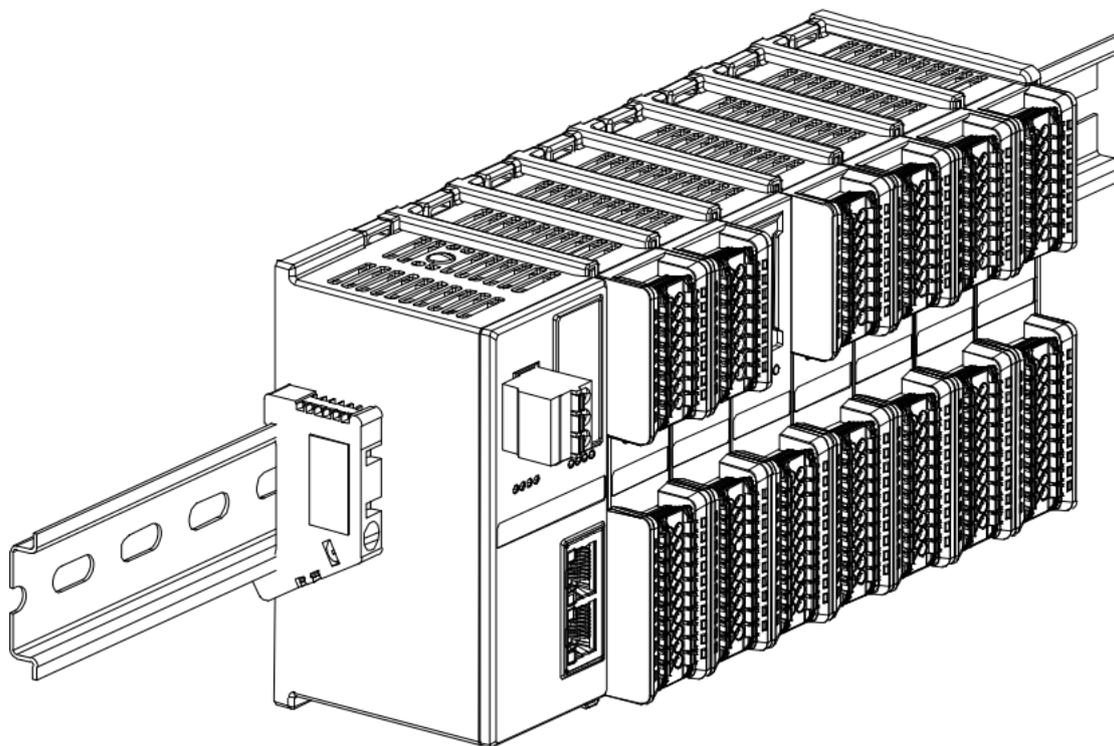
警告

- 如果不按照产品用户手册进行使用，设备提供的保护可能会受到损害。

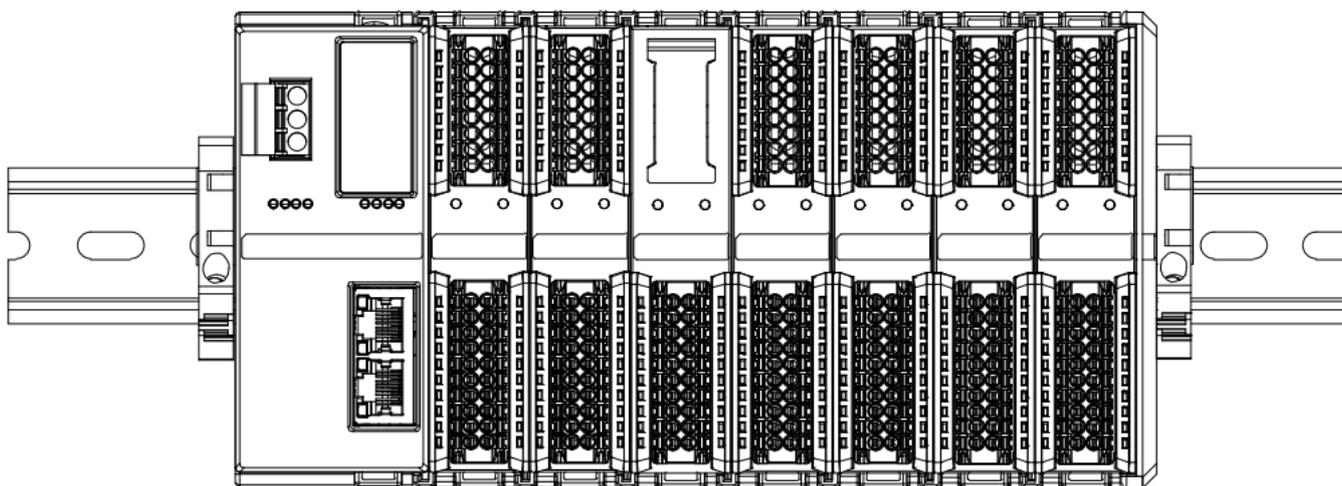
模块安装示意图，上下最小间隙 ($\geq 50\text{mm}$)



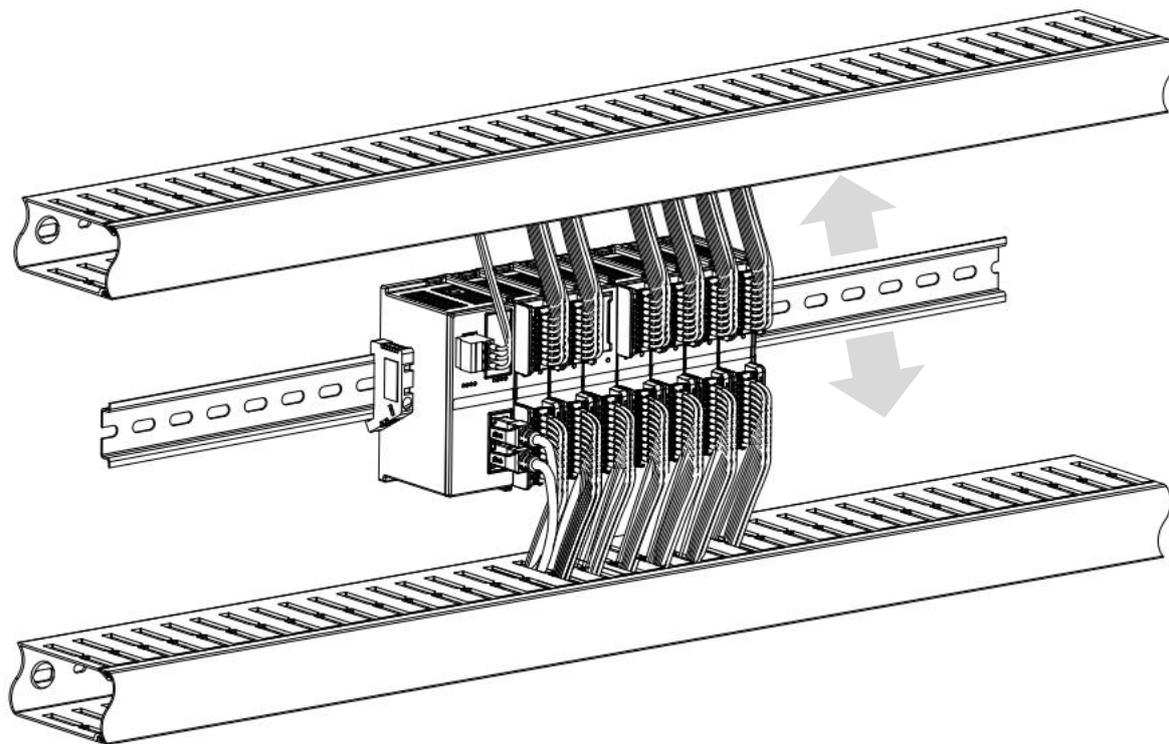
确保模块竖直安装于固定导轨



务必安装导轨固定件



模块上下布线示意图



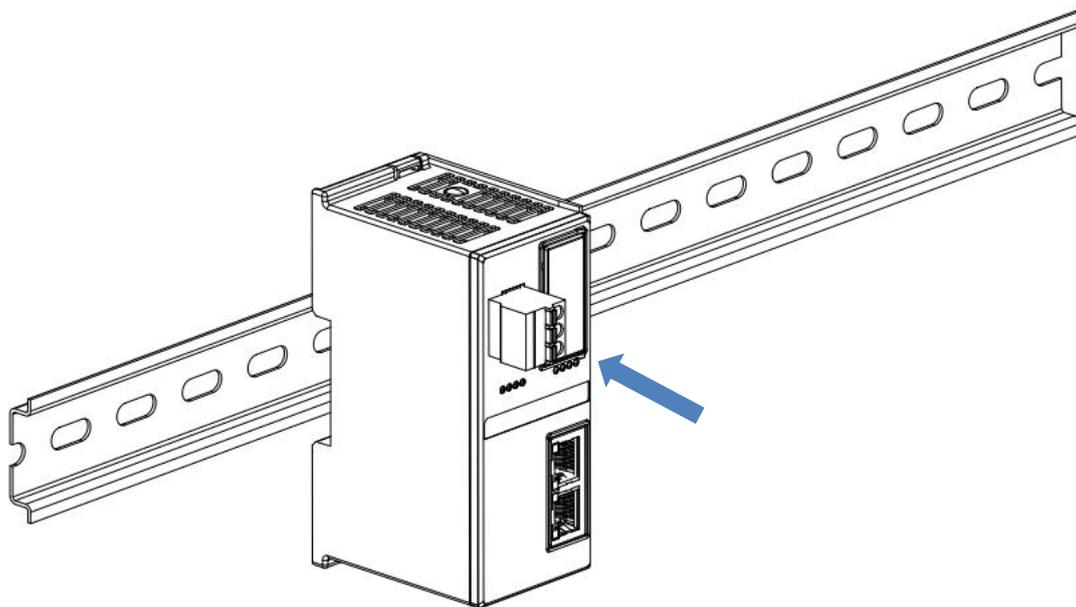
4.3 安装拆卸步骤

模块安装及拆卸	
模块安装步骤	1、在已固定的导轨上先安装耦合器模块。
	2、在耦合器模块的右边依次安装所需要的 I/O 模块或功能模块。
	3、安装所有需要的模块后，安装终端盖板，完成模块的组装。
	4、在耦合器模块、终端盖板的两端安装导轨固定件，将模块固定。
模块拆卸步骤	1、松开模块两端的导轨固定件。
	2、用一字螺丝刀撬开模块卡扣。
	3、拔出拆卸的模块。

4.4 安装拆卸示意图

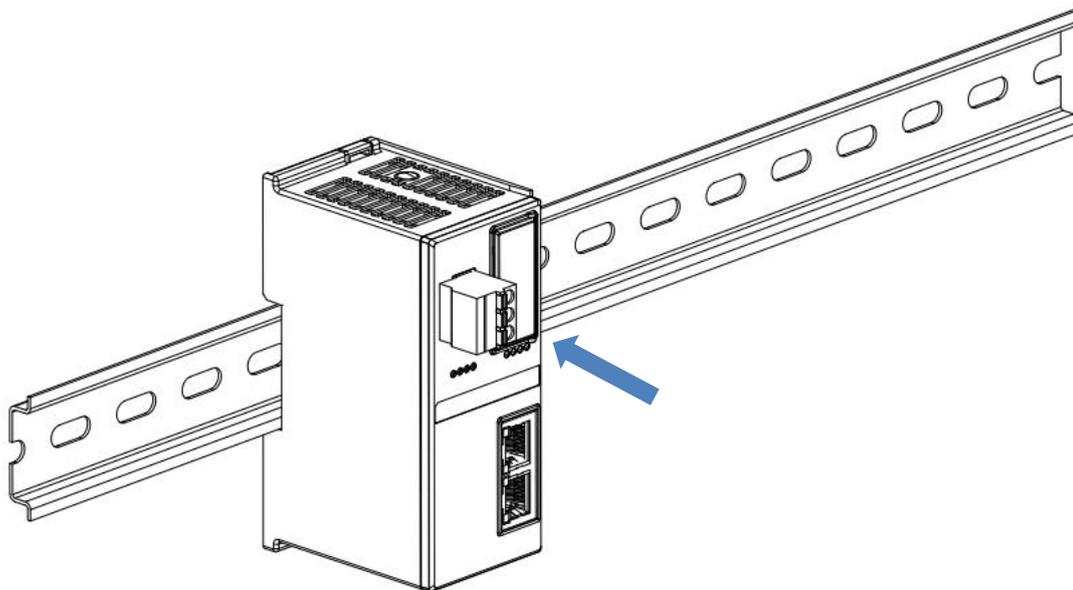
耦合器模块安装

- 将耦合器模块垂直对准导轨卡槽，如下图①所示。



①

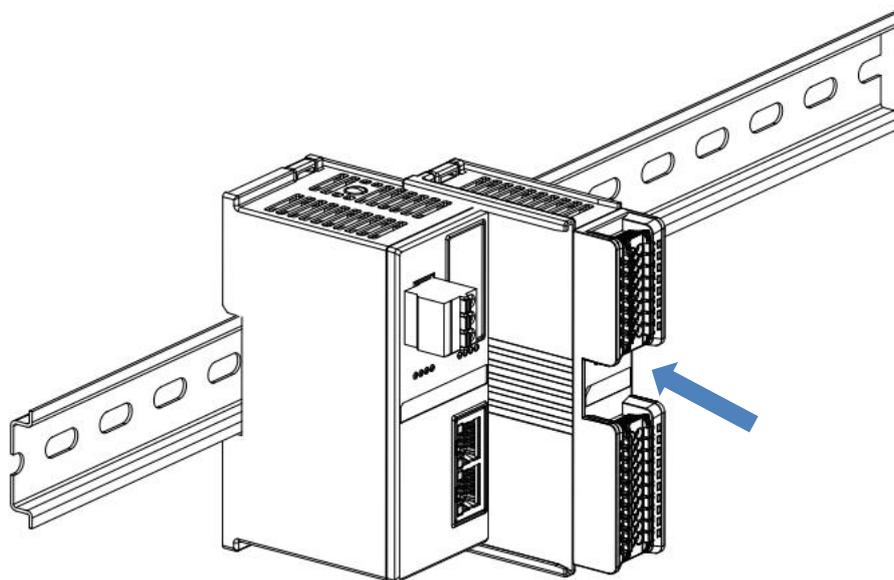
- 用力向导轨方向压耦合器模块，听到“咔哒”声，模块即安装到位，如下图②所示。



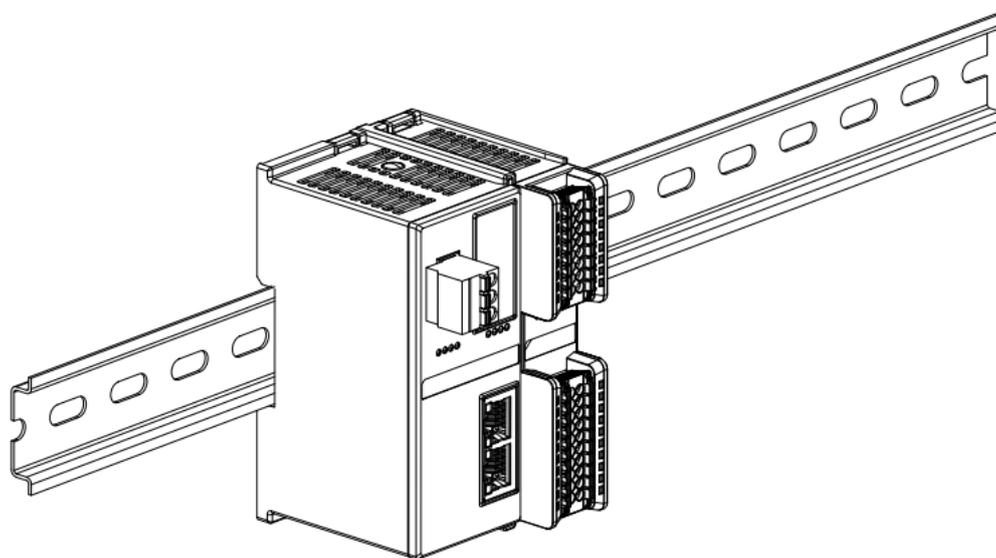
②

I/O 模块安装

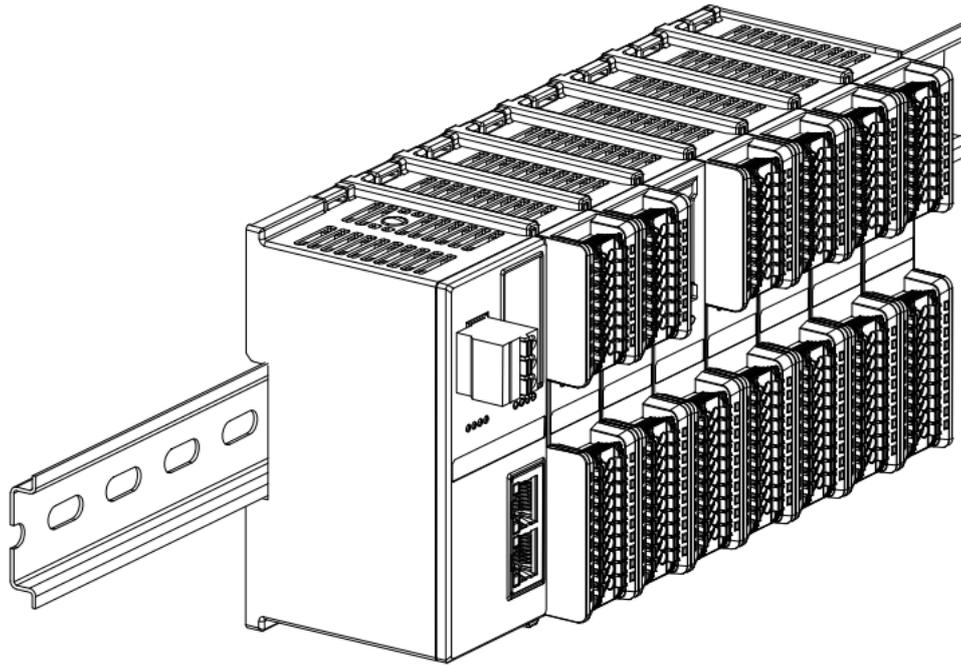
- 按照上述安装耦合器模块的步骤，逐个安装所需要的 I/O 模块或功能模块，如下图③、图④和图⑤所示推入，听到“咔哒”声，模块即安装到位。



③



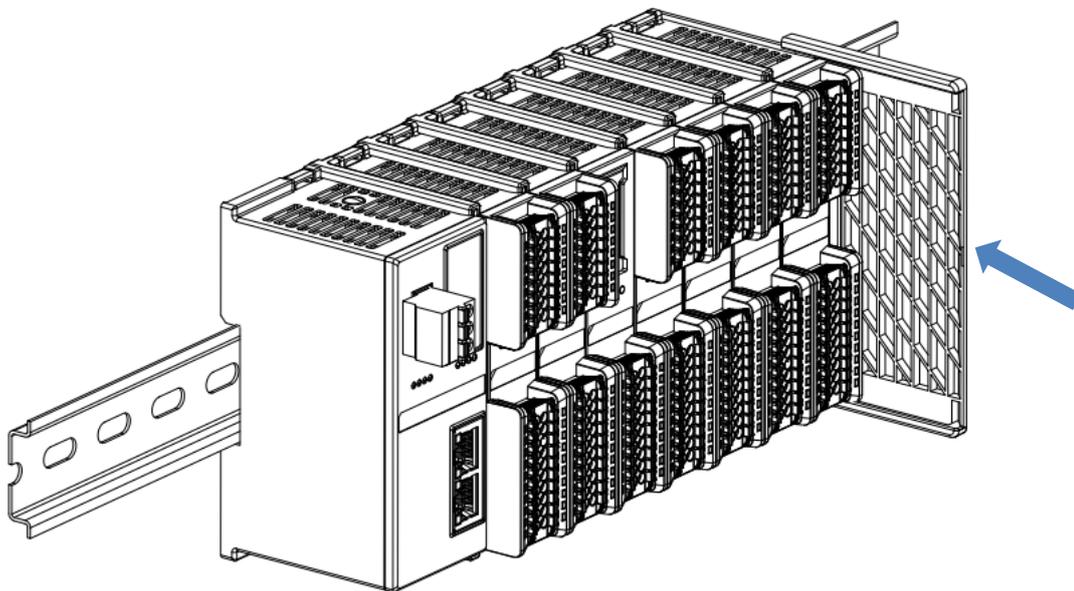
④



⑤

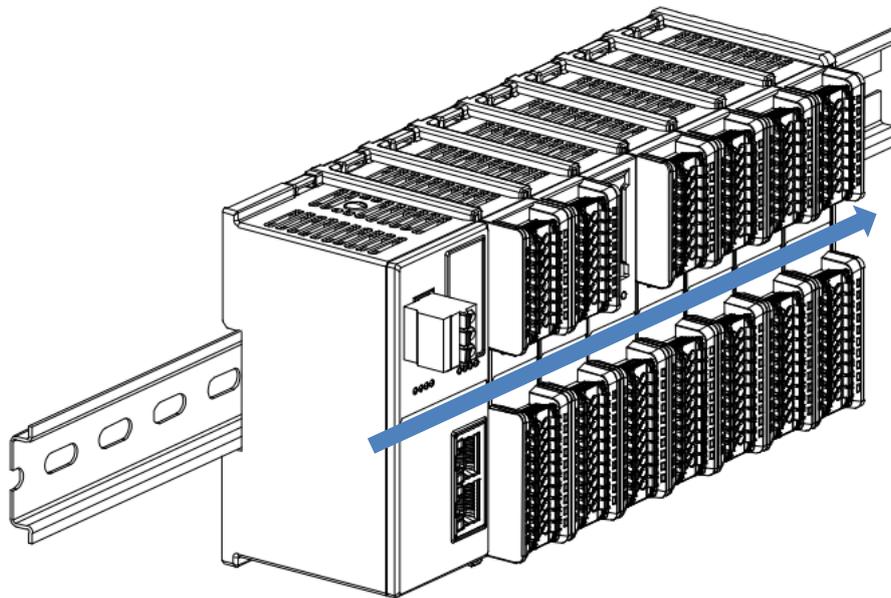
终端盖板安装

- 在最后一个模块的右侧安装终端盖板，终端盖板凹槽一侧对准导轨，安装方式请参照 I/O 模块的安装方法，将终端盖板内推到位，如下图⑥所示。



⑥

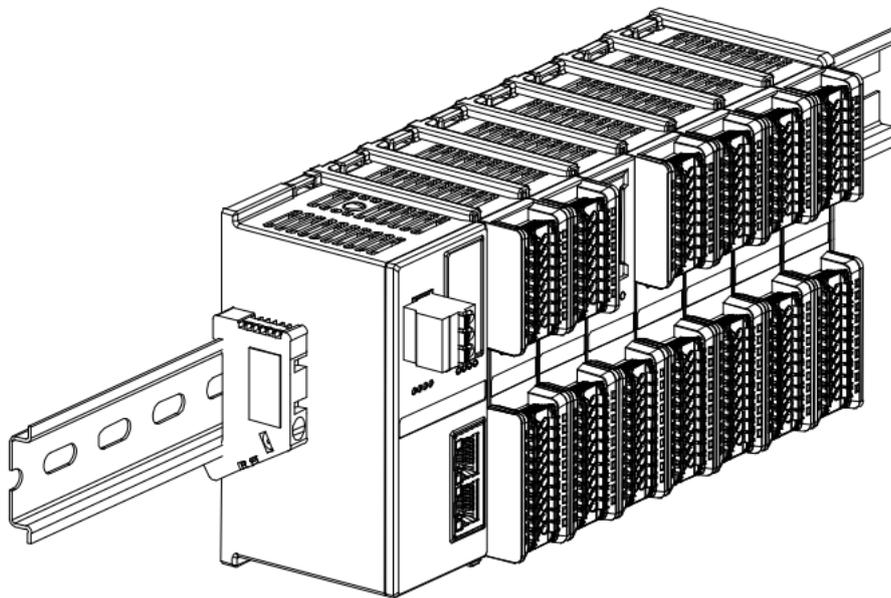
- 终端盖板安装完成后，检查整个模组正面是否平整，确保所有模块和端盖都安装到位，正面平齐，如下图⑦所示。



⑦

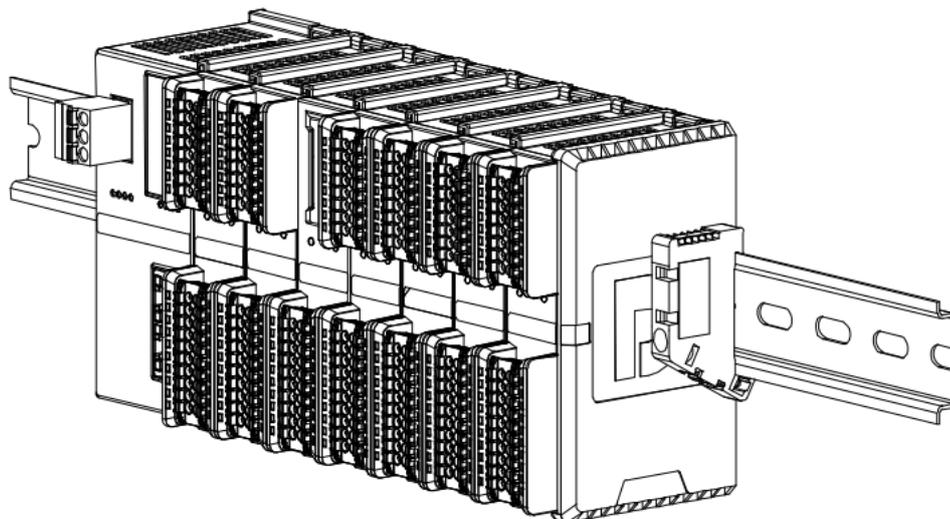
导轨固定件安装

- 紧贴耦合器左侧面安装并拧紧导轨固定件，如下图⑧所示。



⑧

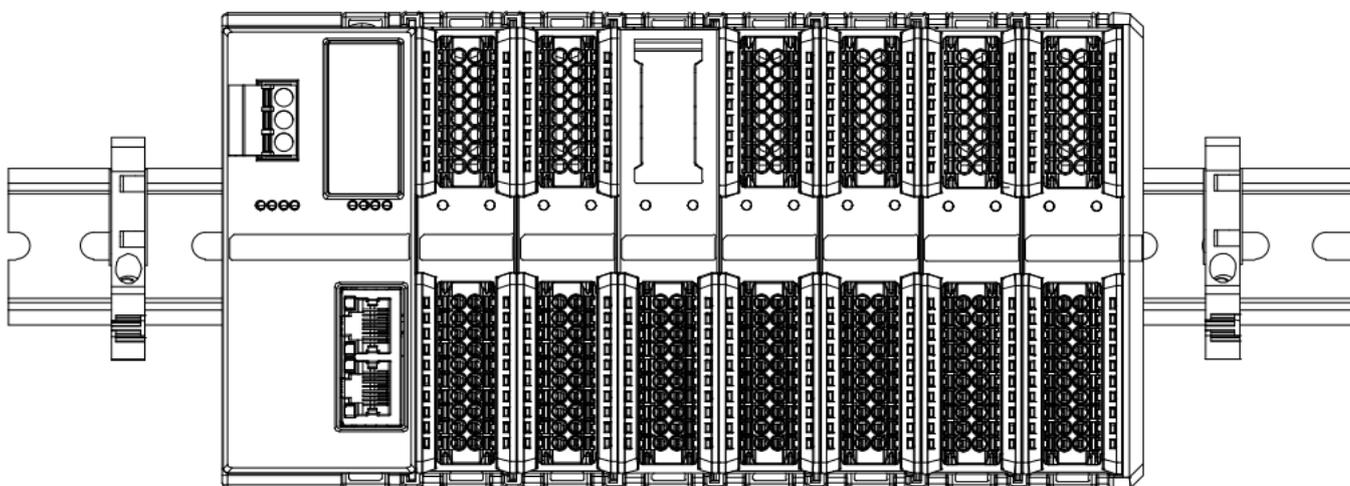
- 在终端盖板右侧安装导轨固定件，先将导轨固定件向耦合器的方向用力推，确保模块安装紧固，并用螺丝刀锁紧导轨固定件，如下图⑨所示。



⑨

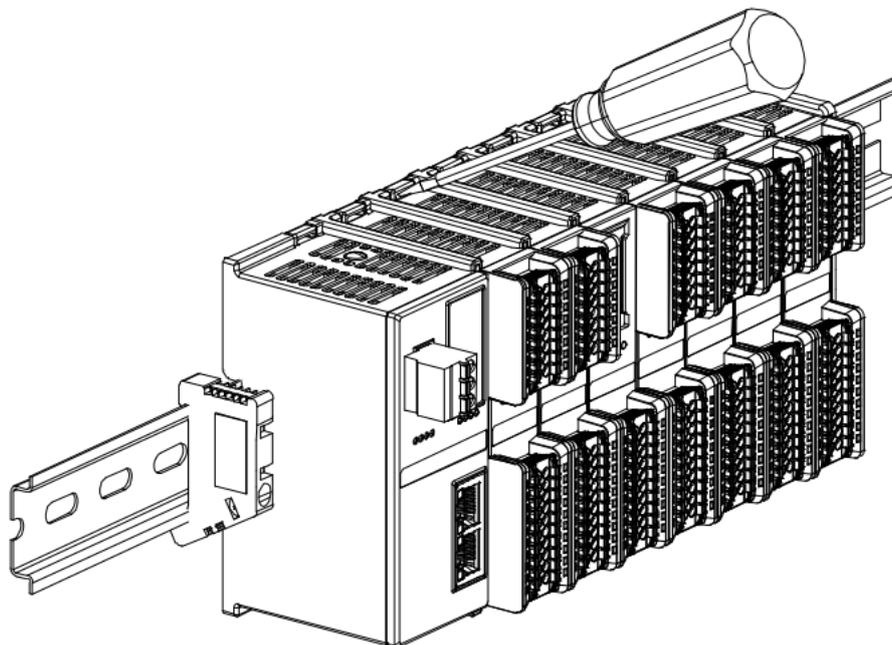
拆卸

- 用螺丝刀松开模块一端导轨固定件，并向一侧移开，确保模块和导轨固定件之间有间隙，如下图⑩所示。

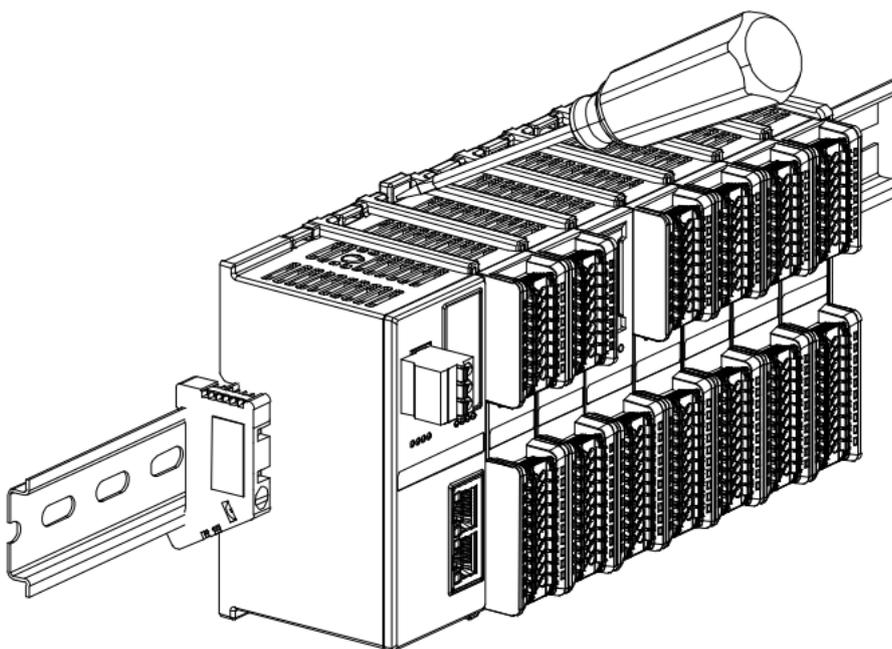


⑩

- 将一字平头起插入待拆卸模块的卡扣，侧向模块的方向用力（听到响声），如下图⑪和⑫所示。**注：每个模块上下各有一个卡扣，均按此方法操作。**

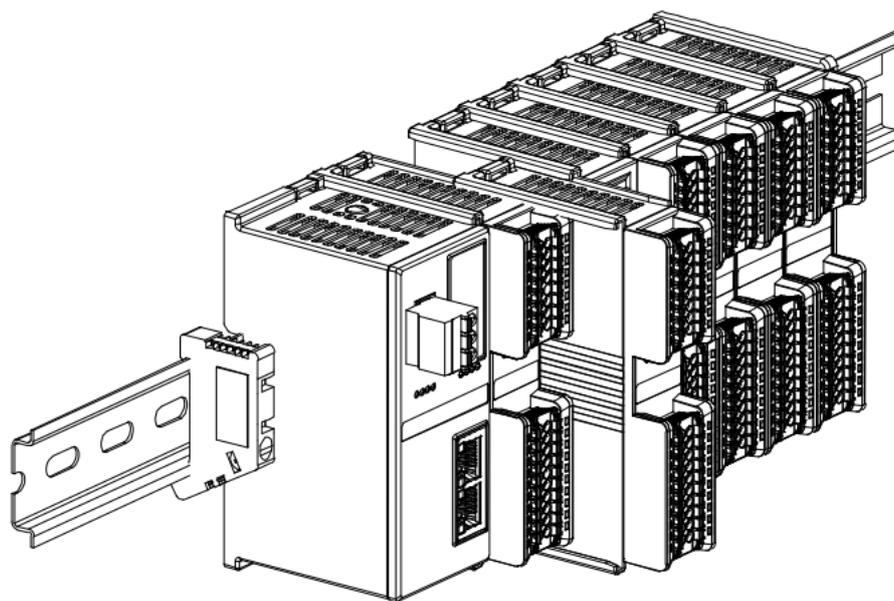


⑪



⑫

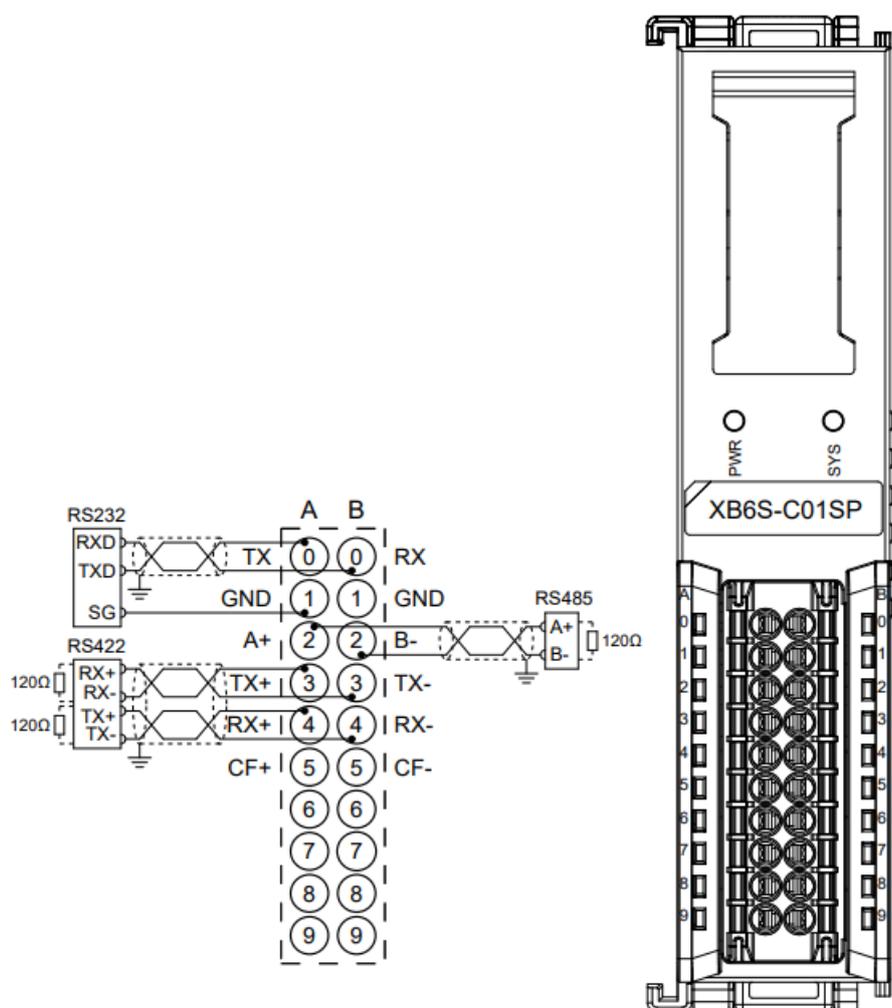
- 按安装模块相反的操作，拆卸模块，如下图⑬所示。



⑬

5 接线

5.1 接线图



- *GND 为 RS232 信号地，内部导通
- *电缆应采用屏蔽双绞线，并可靠接地
- *CF+/CF-为模块预留调试接口，暂未启用

5.2 接线端子定义

A			B		
端子标识	端子定义	说明	端子标识	端子定义	说明
0	TX	RS232 发送端	0	RX	RS232 接收端
1	GND	RS232 信号地	1	GND	RS232 信号地
2	A+	RS485 发送+	2	B-	RS485 发送-
3	TX+	RS422 发送+	3	TX-	RS422 发送-
4	RX+	RS422 接收+	4	RX-	RS422 接收-
5	CF+	预留	5	CF-	预留
6	NC	空端子	6	NC	空端子
7	NC	空端子	7	NC	空端子
8	NC	空端子	8	NC	空端子
9	NC	空端子	9	NC	空端子

6 使用

6.1 配置参数定义

参数名称	描述	取值范围	说明
Communication Mode	通讯模式	0	NoSelect 即无选择
		1	Modbus RTU Master 即 RTU 主站模式
		2	Modbus RTU Slave 即 RTU 从站模式
		3	Modbus ASCII Master 即 ASCII 主站模式
		4	Modbus ASCII Slave 即 ASCII 从站模式
		5	FP_Request 即自由口请求模式
		6	FP_Input_Output 即自由口输入输出模式
		7	FP_Input 即自由口输入模式
		8	FP_Output 即自由口输出模式
		9	SetPDOLength32Bytes
		10	SetPDOLength64Bytes
		11	SetPDOLength128Bytes
		12	SetPDOLength255Bytes
Baud Rate	串行端口波特率	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	停止位	0	1 Bit
		1	2 Bits
Word Format	字符格式	0	8 Bits
		1	7 Bits
Parity	奇偶校验位	0	Parity None 无校验

		1	Parity Odd 奇校验
		2	Parity Even 偶校验
Control Mode	控制与状态模式	0	Disable 即控制与状态失能
		1	Level 即控制与状态使能-高电平触发
		2	Rising Edge 即控制与状态使能-上升沿触发
Node Output Mode	节点输出模式	0	Poll 即轮询
		1	Change Trigger 即逢变
Communication Error Behavior	通讯错误行为	0	Clear 即清空
		1	Hold 即保持
Respond Timeout	响应时间	5~3000	单位 ms, 默认 50
Poll Delay	轮询延时	5~3000	单位 ms, 默认 10
Slave ID	Modbus 从站站号	1~247	在从站模式下有效, 默认 1
Slave Respond Delay	从站响应延时	0~3000	单位 ms, 默认 50
Frame Interval	帧间隔	0	1.5T
		1	3.5T
		2	5T
		3	10T
		4	20T
		5	50T
		6	100T
		7	200T
Node_1_Param_01	节点 1 参数 1	0~FFFFFFFF	默认 0, 详见 节点代码参数
Node_1_Param_02	节点 1 参数 2	0~FFFFFFFF	默认 0, 详见 节点代码参数
...
Node_16_Param_01	节点 16 参数 1	0~FFFFFFFF	默认 0, 详见 节点代码参数
Node_16_Param_02	节点 16 参数 2	0~FFFFFFFF	默认 0, 详见 节点代码参数

6.2 节点代码参数

6.2.1 Modbus 主站功能

Modbus 主站模式下，参数 1 (Param_01) 是 2 字节的起始地址参数和 2 字节的节点代码参数构成，参数 2 (Param_02) 由 2 位的字节交换参数和 1 字节的从站站号参数构成。

Param_01			
BIT31~BIT16		BIT15~BIT0	
起始地址 Start Address		节点代码 Node Code 详见 Modbus 主站节点代码	
Param_02			
BIT31~BIT16		BIT15~BIT10	BIT9~BIT8
Reserve		见 字节交换 ExByte Flag	从站站号 Slave Station Number

注：起始地址参数范围为 0000~FFFFH，从站站号参数范围为 1~247。

6.2.2 Modbus 主站功能节点代码

SubModule 分组	SubModule	SubModule ID (10#)	SubModule ID (16#)
Modbus Master Read Coils 即 Modbus 主站读取线圈	Read 8 bits	257	0x0101
	Read 16 bits	258	0x0102
	Read 24 bits	259	0x0103
	Read 32 bits	260	0x0104
	Read 40 bits	261	0x0105
	Read 48 bits	262	0x0106
	Read 56 bits	263	0x0107
	Read 64 bits	264	0x0108
	Read 72 bits	265	0x0109
	Read 80 bits	266	0x010A
	Read 88 bits	267	0x010B
	Read 96 bits	268	0x010C
	Read 104 bits	269	0x010D
	Read 112 bits	270	0x010E
Read 120 bits	271	0x010F	
Read 128 bits	272	0x0110	
Modbus Master Read Discrete 即 Modbus 主站读取离散输入	Read 8 bits	513	0x0201
	Read 16 bits	514	0x0202
	Read 24 bits	515	0x0203
	Read 32 bits	516	0x0204
	Read 40 bits	517	0x0205
	Read 48 bits	518	0x0206
	Read 56 bits	519	0x0207
Read 64 bits	520	0x0208	

	Read 72 bits	521	0x0209
	Read 80 bits	522	0x020A
	Read 88 bits	523	0x020B
	Read 96 bits	524	0x020C
	Read 104 bits	525	0x020D
	Read 112 bits	526	0x020E
	Read 120 bits	527	0x020F
	Read 128 bits	528	0x0210
Modbus Master Read Hold 即 Modbus 主站读保持寄存器	Read 1 Reg	769	0x0301
	Read 2 Regs	770	0x0302
	Read 3 Regs	771	0x0303
	Read 4 Regs	772	0x0304
	Read 5 Regs	773	0x0305
	Read 6 Regs	774	0x0306
	Read 7 Regs	775	0x0307
	Read 8 Regs	776	0x0308
	Read 9 Regs	777	0x0309
	Read 10 Regs	778	0x030A
	Read 11 Regs	779	0x030B
	Read 12 Regs	780	0x030C
	Read 13 Regs	781	0x030D
	Read 14 Regs	782	0x030E
	Read 15 Regs	783	0x030F
	Read 16 Regs	784	0x0310
Modbus Master Read Input 即 Modbus 主站读输入寄存器	Read 1 Reg	1025	0x0401
	Read 2 Regs	1026	0x0402
	Read 3 Regs	1027	0x0403
	Read 4 Regs	1028	0x0404
	Read 5 Regs	1029	0x0405
	Read 6 Regs	1030	0x0406
	Read 7 Regs	1031	0x0407
	Read 8 Regs	1032	0x0408
	Read 9 Regs	1033	0x0409
	Read 10 Regs	1034	0x040A
	Read 11 Regs	1035	0x040B
	Read 12 Regs	1036	0x040C
	Read 13 Regs	1037	0x040D
	Read 14 Regs	1038	0x040E
	Read 15 Regs	1039	0x040F
	Read 16 Regs	1040	0x0410
Modbus Master Write Coils 即 Modbus 主站写线圈	Write 8 bits	1281	0x0501
	Write 16 bits	1282	0x0502
	Write 24 bits	1283	0x0503

	Write 32 bits	1284	0x0504
	Write 40 bits	1285	0x0505
	Write 48 bits	1286	0x0506
	Write 56 bits	1287	0x0507
	Write 64 bits	1288	0x0508
	Write 72 bits	1289	0x0509
	Write 80 bits	1290	0x050A
	Write 88 bits	1291	0x050B
	Write 96 bits	1292	0x050C
	Write 104 bits	1293	0x050D
	Write 112 bits	1294	0x050E
	Write 120 bits	1295	0x050F
	Write 128 bits	1296	0x0510
	Write Single bit	1535	0x05FF
Modbus Master Write Hold 即 Modbus 主站写保持寄存器	Write 1 Reg	1537	0x0601
	Write 2 Regs	1538	0x0602
	Write 3 Regs	1539	0x0603
	Write 4 Regs	1540	0x0604
	Write 5 Regs	1541	0x0605
	Write 6 Regs	1542	0x0606
	Write 7 Regs	1543	0x0607
	Write 8 Regs	1544	0x0608
	Write 9 Regs	1545	0x0609
	Write 10 Regs	1546	0x060A
	Write 11 Regs	1547	0x060B
	Write 12 Regs	1548	0x060C
	Write 13 Regs	1549	0x060D
	Write 14 Regs	1550	0x060E
	Write 15 Regs	1551	0x060F
	Write 16 Regs	1552	0x0610
Write Single Reg	1791	0x06FF	

6.2.3 Modbus 从站功能

Modbus 从站模式下，参数 1 (Param_01) 是 2 字节的起始地址参数和 2 字节的节点代码参数构成，参数 2 (Param_02) 由 2 位的字节交换参数构成。

Param_01		
BIT31~BIT16	BIT15~BIT0	
起始地址 Start Address	节点代码 Node Code 详见 Modbus 从站节点代码	
Param_02		
BIT31~BIT16	BIT15~BIT2	BIT1~BIT0
Reserve		见 字节交换 ExByte Flag

注：起始地址参数范围为 0000~FFFFH

6.2.4 Modbus 从站功能节点代码

SubModule 分组	SubModule	SubModule ID (10#)	SubModule ID (16#)
Modbus Slave Read Coils 即 Modbus 从站读线圈	Read 8 bits	1793	0x0701
	Read 16 bits	1794	0x0702
	Read 24 bits	1795	0x0703
	Read 32 bits	1796	0x0704
	Read 40 bits	1797	0x0705
	Read 48 bits	1798	0x0706
	Read 56 bits	1799	0x0707
	Read 64 bits	1800	0x0708
	Read 72 bits	1801	0x0709
	Read 80 bits	1802	0x070A
	Read 88 bits	1803	0x070B
	Read 96 bits	1804	0x070C
	Read 104 bits	1805	0x070D
	Read 112 bits	1806	0x070E
Read 120 bits	1807	0x070F	
Read 128 bits	1808	0x0710	
Modbus Slave Read Hold 即 Modbus 从站读保持寄存器	Read 1 Reg	2049	0x0801
	Read 2 Regs	2050	0x0802
	Read 3 Regs	2051	0x0803
	Read 4 Regs	2052	0x0804
	Read 5 Regs	2053	0x0805
	Read 6 Regs	2054	0x0806
	Read 7 Regs	2055	0x0807
	Read 8 Regs	2056	0x0808
	Read 9 Regs	2057	0x0809
	Read 10 Regs	2058	0x080A
	Read 11 Regs	2059	0x080B

	Read 12 Regs	2060	0x080C
	Read 13 Regs	2061	0x080D
	Read 14 Regs	2062	0x080E
	Read 15 Regs	2063	0x080F
	Read 16 Regs	2064	0x0810
Modbus Slave Write Coils 即 Modbus 从站写线圈	Write 8 bits	2305	0x0901
	Write 16 bits	2306	0x0902
	Write 24 bits	2307	0x0903
	Write 32 bits	2308	0x0904
	Write 40 bits	2309	0x0905
	Write 48 bits	2310	0x0906
	Write 56 bits	2311	0x0907
	Write 64 bits	2312	0x0908
	Write 72 bits	2313	0x0909
	Write 80 bits	2314	0x090A
	Write 88 bits	2315	0x090B
	Write 96 bits	2316	0x090C
	Write 104 bits	2317	0x090D
	Write 112 bits	2318	0x090E
Write 120 bits	2319	0x090F	
Write 128 bits	2320	0x0910	
Modbus Slave Write Discrete 即 Modbus 从站写离散输入	Write 8 bits	2561	0x0A01
	Write 16 bits	2562	0x0A02
	Write 24 bits	2563	0x0A03
	Write 32 bits	2564	0x0A04
	Write 40 bits	2565	0x0A05
	Write 48 bits	2566	0x0A06
	Write 56 bits	2567	0x0A07
	Write 64 bits	2568	0x0A08
	Write 72 bits	2569	0x0A09
	Write 80 bits	2570	0x0A0A
	Write 88 bits	2571	0x0A0B
	Write 96 bits	2572	0x0A0C
	Write 104 bits	2573	0x0A0D
	Write 112 bits	2574	0x0A0E
Write 120 bits	2575	0x0A0F	
Write 128 bits	2576	0x0A10	
Modbus Slave Write Hold 即 Modbus 从站写保持寄存器	Write 1 Reg	2817	0x0B01
	Write 2 Regs	2818	0x0B02
	Write 3 Regs	2819	0x0B03
	Write 4 Regs	2820	0x0B04
	Write 5 Regs	2821	0x0B05
	Write 6 Regs	2822	0x0B06

	Write 7 Regs	2823	0x0B07
	Write 8 Regs	2824	0x0B08
	Write 9 Regs	2825	0x0B09
	Write 10 Regs	2826	0x0B0A
	Write 11 Regs	2827	0x0B0B
	Write 12 Regs	2828	0x0B0C
	Write 13 Regs	2829	0x0B0D
	Write 14 Regs	2830	0x0B0E
	Write 15 Regs	2831	0x0B0F
	Write 16 Regs	2832	0x0B10
Modbus Slave Write Input 即 Modbus 从站写输入寄存器	Write 1 Reg	3073	0x0C01
	Write 2 Regs	3074	0x0C02
	Write 3 Regs	3075	0x0C03
	Write 4 Regs	3076	0x0C04
	Write 5 Regs	3077	0x0C05
	Write 6 Regs	3078	0x0C06
	Write 7 Regs	3079	0x0C07
	Write 8 Regs	3080	0x0C08
	Write 9 Regs	3081	0x0C09
	Write 10 Regs	3082	0x0C0A
	Write 11 Regs	3083	0x0C0B
	Write 12 Regs	3084	0x0C0C
	Write 13 Regs	3085	0x0C0D
	Write 14 Regs	3086	0x0C0E
	Write 15 Regs	3087	0x0C0F
	Write 16 Regs	3088	0x0C10

6.2.5 字节交换

字节编号	功能	长度
0	不交换	2 位 Bits
1	2Bytes Exchange	2 位 Bits
2	4Bytes Register Exchange	2 位 Bits
3	4Bytes Big-Little-Endian Exchange	2 位 Bits

6.2.6 Freeport 自由口功能

在 Freeport 自由口模式下, 参数 1 (Param_01) 是 2 字节的节点代码参数构成。

Param_01	
BIT31~BIT16	BIT15~BIT0
Reserve	节点代码 Node Code 详见 Freeport 节点代码
Param_02	
BIT31~BIT0	
Reserve	

6.2.7 Freeport 功能节点代码

SubModule 分组	SubModule	SubModule ID (10#)	SubModule ID (16#)
Freeport Input 即自由口输入	Input 1 byte	3329	0x0D01
	Input 2 bytes	3330	0x0D02
	Input 4 bytes	3331	0x0D03
	Input 8 bytes	3332	0x0D04
	Input 16 bytes	3333	0x0D05
	Input 32 bytes	3334	0x0D06
	Input 64 bytes	3335	0x0D07
	Input 128 bytes	3336	0x0D08
	Input 160 bytes	3337	0x0D09
	Input 192 bytes	3338	0x0D0A
	Input 224 bytes	3339	0x0D0B
Input 255 bytes	3340	0x0D0C	
Freeport Output 即自由口输出	Output 1 byte	3585	0x0E01
	Output 2 bytes	3586	0x0E02
	Output 4 bytes	3587	0x0E03
	Output 8 bytes	3588	0x0E04
	Output 16 bytes	3589	0x0E05
	Output 32 bytes	3590	0x0E06
	Output 64 bytes	3591	0x0E07
	Output 128 bytes	3592	0x0E08
	Output 160 bytes	3593	0x0E09
Output 192 bytes	3594	0x0E0A	

	Output 224 bytes	3595	0x0E0B
	Output 255 bytes	3596	0x0E0C
Freeport Parity 即自由口 奇偶校验	CRC8	3841	0x0F01
	CRC16	3842	0x0F02
	XOR	3843	0x0F03
	SUM8	3844	0x0F04
	LRC	3845	0x0F05

6.2.8 控制与状态节点代码

当配置参数 Control Mode 控制与状态模式设置为 disable 时，无需配置控制与状态节点代码；当配置参数 Control Mode 控制与状态模式设置为 Level 或 Rising Edge 时，必需先在节点 1 参数 1 (Node_1_Param_01) 配置节点代码。

节点代码如下表所示：

SubModule 分组	SubModule	SubModule ID (10#)	SubModule ID (16#)	备注
Control&Status	1 Node	1	0x0001	主站模式下配置
	2 Node	2	0x0002	
	3 Node	3	0x0003	
	4 Node	4	0x0004	
	5 Node	5	0x0005	
	6 Node	6	0x0006	
	7 Node	7	0x0007	
	8 Node	8	0x0008	
	9 Node	9	0x0009	
	10 Node	10	0x000A	
	11 Node	11	0x000B	
	12 Node	12	0x000C	
	13 Node	13	0x000D	
	14 Node	14	0x000E	
	15 Node	15	0x000F	
	Freeport Req	32	0x0020	自由口模式下配置
	Freeport Input_Output	33	0x0021	
	Freeport Input	34	0x0022	
	Freeport Output	35	0x0023	从站模式下配置
Modbus Slave	48	0x0030		

6.3 过程数据

模块出厂默认为 XB6S-C01SP-32Bytes，上下行数据长度各 32Bytes，当模块的通讯节点参数配置所产生的 PDO 数据量达到一定范围时（上行数据量或下行数据量任一到达范围限制），耦合器模块会告警 0x5002，手动切换相应的 PDO 数据长度，重新上电后扫描模块，具体步骤详见 [6.4.1 切换 PDO 数据长度示例](#)。（用户最好提前预估 PDO 数据长度，并切换相应的长度）

模块 PDO 长度范围与型号匹配表：

模块 PDO 长度范围与型号匹配表		
模块型号	上行数据长度 (Bytes)	下行数据长度 (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

下面以型号 XB6S-C01SP-32Bytes 为例介绍配置参数。

➤ 当配置参数 **Control Mode** 控制与状态模式设置为 **disable** 时，上下行过程数据如下表所示。

上行数据 (32 字节)			
字节编号	功能	含义	地址范围
1	输入数据	串口通讯模块输入数据	32 个字节
2	输入数据		
...	...		
32	输入数据		
下行数据 (32 字节)			
字节编号	功能	含义	地址范围
1	输出数据	串口通讯模块输出数据	32 个字节
2	输出数据		
...	...		
32	输出数据		

- 当配置参数 Control Mode 控制与状态模式设置为 Level 或 Rising Edge 时，上下行过程数据如下表所示。

6.3.1 Modbus 主站过程数据

上行数据 (32 字节)			
字节编号	功能	含义	地址范围
1	状态字	Bit0: 空闲状态	1 个字节
		Bit1: busy 状态, 即传输数据中	
		Bit2: 完成状态, 即传输数据完成	
2	告警码	见 告警码	1 个字节
3	输入数据	串口通讯模块输入数据	后 30 个字节
4	输入数据		
...	...		
...	...		
32	输入数据		
下行数据 (32 字节)			
字节编号	功能	含义	地址范围
1	控制字	当配置参数 Control Mode 为 Level, 置 1 发送数据	1 个字节
		当配置参数 Control Mode 为 Rising Edge, 0->1 发送数据	
2	输出数据	串口通讯模块输出数据	后 31 个字节
3	输出数据		
...	...		
...	...		
32	输出数据		

注：上述表格过程数据是 Level 或 Rising Edge 模式下配置 1 个节点，如配置两个节点，上行数据的字节编号 1 和 2 为节点 1 的状态位和告警码，字节编号 3 和 4 为节点 2 的状态位和告警码，剩余字节为数据位；下行数据的字节编号 1 为节点 1 的控制位，字节编号 2 为节点 2 的控制位，剩余字节为数据位，最多可配置 15 个节点。

6.3.2 Modbus 从站过程数据

上行数据 (32 字节)			
字节编号	功能	含义	地址范围
1	告警码	见 告警码	1 个字节
2	输入数据	串口通讯模块输入数据	后 31 个字节
3	输入数据		
...	...		
32	输入数据		
下行数据 (32 字节)			
字节编号	功能	含义	地址范围
1	控制字	Bit0: 告警码清除	1 个字节
2	输出数据	串口通讯模块输出数据	后 31 个字节
3	输出数据		
...	...		
32	输出数据		

6.3.3 Freeport_Request 过程数据

上行数据 (32 字节)			
字节编号	功能	含义	地址范围
1	状态字	Bit0: 空闲状态	1 个字节
		Bit1: busy 状态, 即传输数据中	
		Bit2: 完成状态, 即传输数据完成	
2	告警码	Bit0: 接收数据长度错误	1 个字节
		Bit1: 校验错误	
		Bit2: 超时错误	
3	接收数据长度	0~255	1 个字节
4~5	接收异常计数	0~65535	2 个字节
6~7	接收数据计数	0~65535	2 个字节
8	输入数据	串口通讯模块输入数据	后 25 个字节
9	输入数据		
...	...		
...	...		
32	输入数据		
下行数据 (32 字节)			
字节编号	功能	含义	地址范围
1	控制字	Bit0: 使能	1 个字节
		Bit1: 状态字清空	
		Bit2: 接收数据清空	
2	发送数据长度	0~255	1 个字节
3	计数清零	Bit0: 接收异常计数清零	1 个字节
		Bit1: 接收数据计数清零	
		Bit2: 接收数据长度清零	
4	错误码清零	Bit0: 长度异常清除	1 个字节
		Bit1: 校验错误清除	
		Bit2: 超时错误清除	
5	输出数据	串口通讯模块输出数据	后 28 个字节
6	输出数据		
...	...		
...	...		
32	输出数据		

6.3.4 Freeport_Input 过程数据

上行数据 (32 字节)			
字节编号	功能	含义	地址范围
1	告警码	Bit0: 接收数据长度错误	1 个字节
		Bit1: 校验错误	
2	接收数据长度	0~255	1 个字节
3~4	接收异常计数	0~65535	2 个字节
5~6	接收数据计数	0~65535	2 个字节
7	输入数据	串口通讯模块输入数据	后 26 个字节
8	输入数据		
...	...		
32	输入数据		
下行数据 (32 字节)			
字节编号	功能	含义	地址范围
1	控制字	Bit0: 使能	1 个字节
		Bit1: 状态字清空	
		Bit2: 接收数据清空	
2	计数清零	Bit0: 接收异常计数清零	1 个字节
		Bit1: 接收数据计数清零	
		Bit2: 接收数据长度清零	
3	错误码清零	Bit0: 接收长度错误清除	1 个字节
		Bit1: 校验错误清除	
4	输出数据	串口通讯模块输出数据	后 29 个字节
5	输出数据		
...	...		
32	输出数据		

6.3.5 Freeport_Output 过程数据

下行数据 (32 字节)			
字节编号	功能	含义	地址范围
1	控制字	Bit0: 使能	1 个字节
2	发送数据长度	0~255	1 个字节
3	输出数据	串口通讯模块输出数据	后 30 个字节
4	输出数据		
...	...		
32	输出数据		

6.3.6 Freeport_Input_Output 过程数据

上行数据 (32 字节)			
字节编号	功能	含义	地址范围
1	告警码	Bit0: 接收数据长度错误	1 个字节
		Bit1: 校验错误	
2	接收数据长度	0~255	1 个字节
3~4	接收异常计数	0~65535	2 个字节
5~6	接收数据计数	0~65535	2 个字节
7	输入数据	串口通讯模块输入数据	后 26 个字节
8	输入数据		
...	...		
32	输入数据		
下行数据 (32 字节)			
字节编号	功能	含义	地址范围
1	控制字	Bit0: 使能	1 个字节
		Bit1: 状态字清空	
		Bit2: 接收数据清空	
2	发送数据长度	0~255	1 个字节
3	计数清零	Bit0: 接收异常计数清零	1 个字节
		Bit1: 接收数据计数清零	
		Bit2: 接收数据长度清零	
4	错误码清零	Bit0: 接收长度错误清除	1 个字节
		Bit1: 校验错误清除	
5	输出数据	串口通讯模块输出数据	后 28 个字节
6	输出数据		
...	...		
32	输出数据		

6.3.7 Modbus 告警码

故障码	注释
0x00	无错误
0x01	从设备地址错误
0x02	功能码错误
0x03	CRC 校验错误
0x04	响应字节数错误
0x05	线圈寄存器数据错误
0x06	保持寄存器数据错误
0x07	非法数据地址
0x08	从设备故障
0x09	LRC 校验错误
0x0C	数据长度错误
0x0D	数据转换错误
0x0E	ASCII 帧错误
0x0F	响应超时

6.4 模块组态说明

6.4.1 在 TwinCAT3 软件环境下的应用

1、准备工作

- 硬件环境

- 模块型号 XB6S-C01SP
- EtherCAT 总线耦合器模块，端盖
本说明以 XB6S-EC2002 耦合器模块为例
- 计算机一台，预装 TwinCAT3 软件
- EtherCAT 专用屏蔽电缆
- 开关电源一台
- 模块安装导轨及导轨固定件
- 设备配置文件

配置文件获取地址：<https://www.solidotech.com/documents/configfile>

- 硬件组态及接线

请按照“4 安装和拆卸”“5 接线”要求操作

2、预置配置文件

将 ESI 配置文件 (EcatTerminal-XB6S_V1.19.13_ENUM.xml) 放置于 TwinCAT 的安装目录“C:\TwinCAT\3.1\Config\Io\EtherCAT”下，如下图所示。

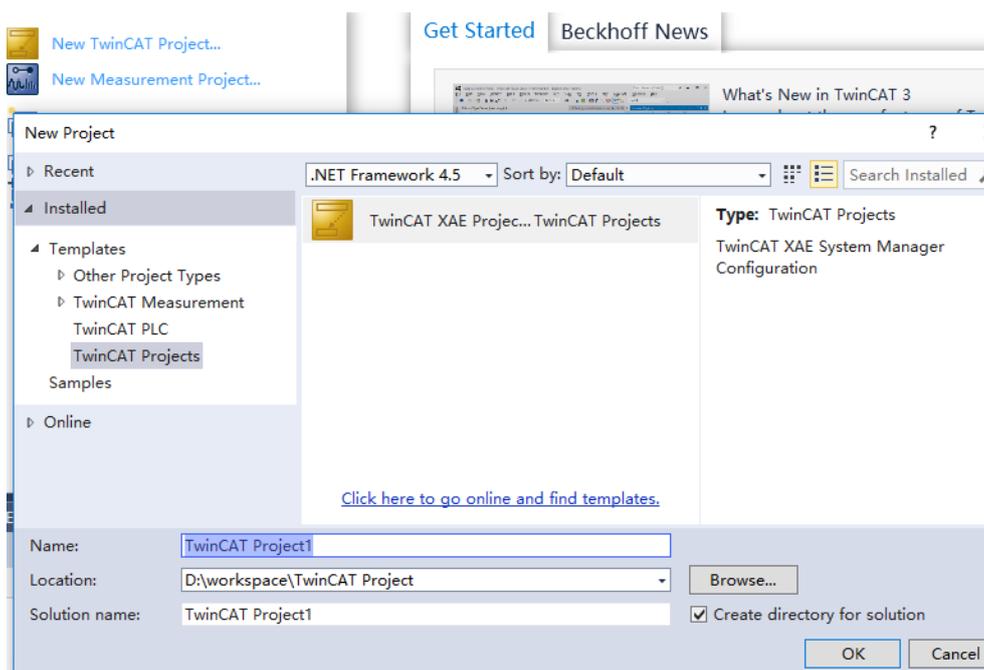
名称	修改日期	类型	大小
Beckhoff EL32xx.xml	2017/10/25 15:43	XML 文档	5,997 KB
Beckhoff EL66xx.xml	2017/10/27 8:55	XML 文档	1,820 KB
Beckhoff EKx9xx.xml	2017/11/3 9:53	XML 文档	1,223 KB
Beckhoff EP7xx.xml	2017/11/8 9:46	XML 文档	9,290 KB
Beckhoff ATH2xx.xml	2017/11/23 13:22	XML 文档	439 KB
Beckhoff EPP3xx.xml	2017/12/8 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、创建工程

- a. 单击桌面右下角的 TwinCAT 图标，选择 “TwinCAT XAE (VS xxxx) ” ，打开 TwinCAT 软件，如下图所示。

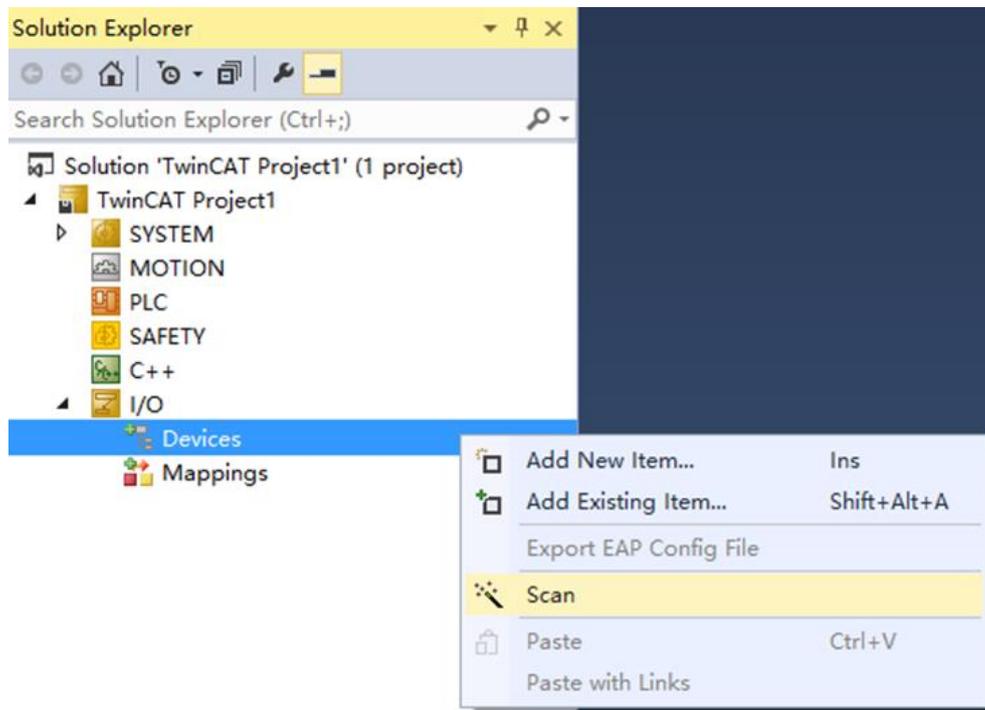


- b. 单击 “New TwinCAT Project” ，在弹窗内 “Name” 和 “Solution name” 分别对应项目名称和解决方案名称，“Location” 对应项目路径，此三项可选择默认，然后单击 “OK” ，项目创建成功，如下图所示。

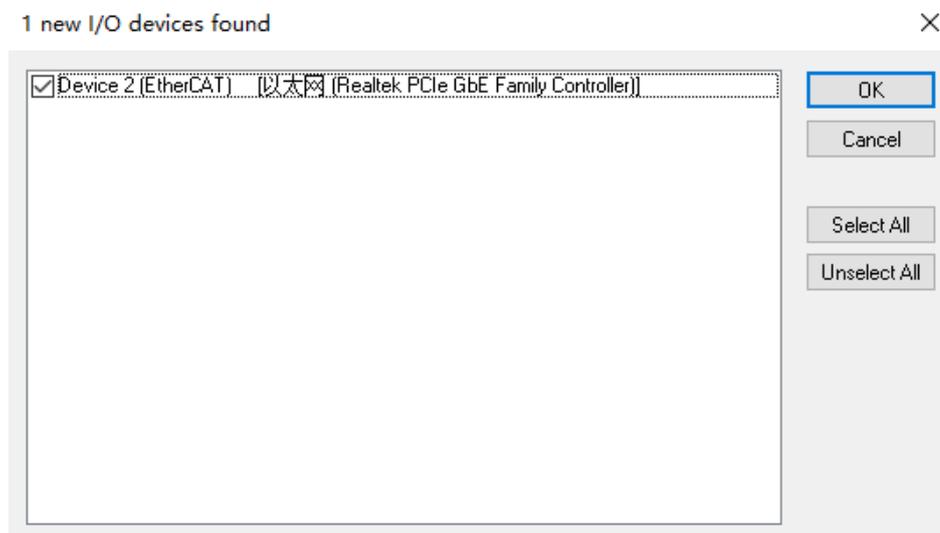


4、扫描设备

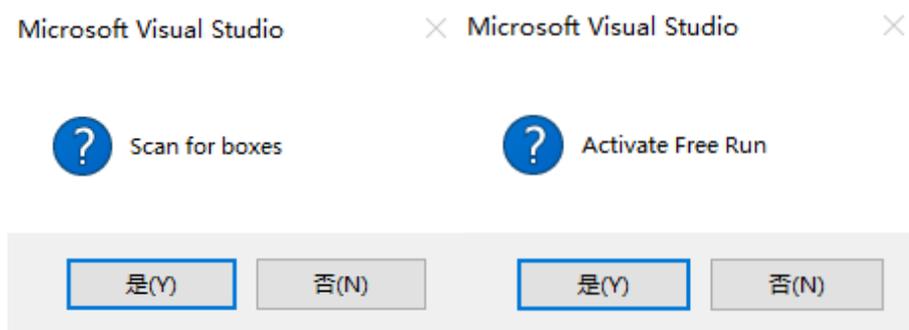
- a. 创建项目后，在 “I/O -> Devices” 下右击 “Scan” 选项，进行从站设备扫描，如下图所示。



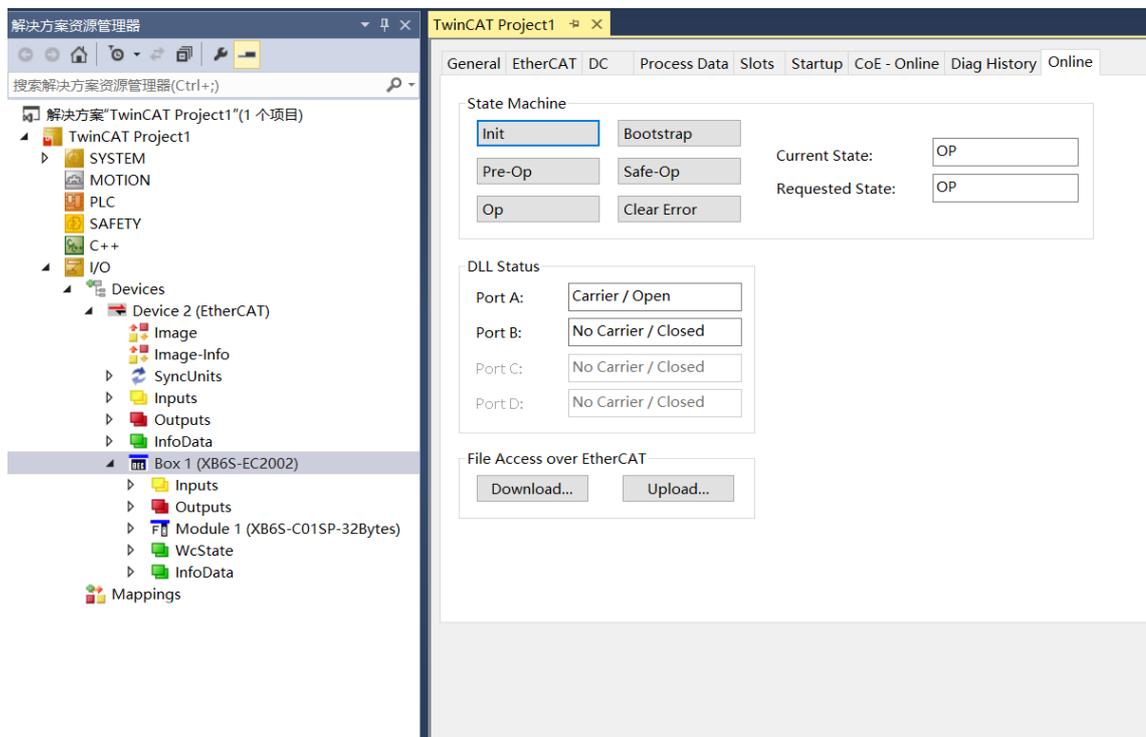
- b. 勾选 “本地连接” 网卡，如下图所示。



- c. 弹窗 “Scan for boxes” ，单击选择 “是” ；弹窗 “Activate Free Run” 单击选择 “是” ，如下图所示。

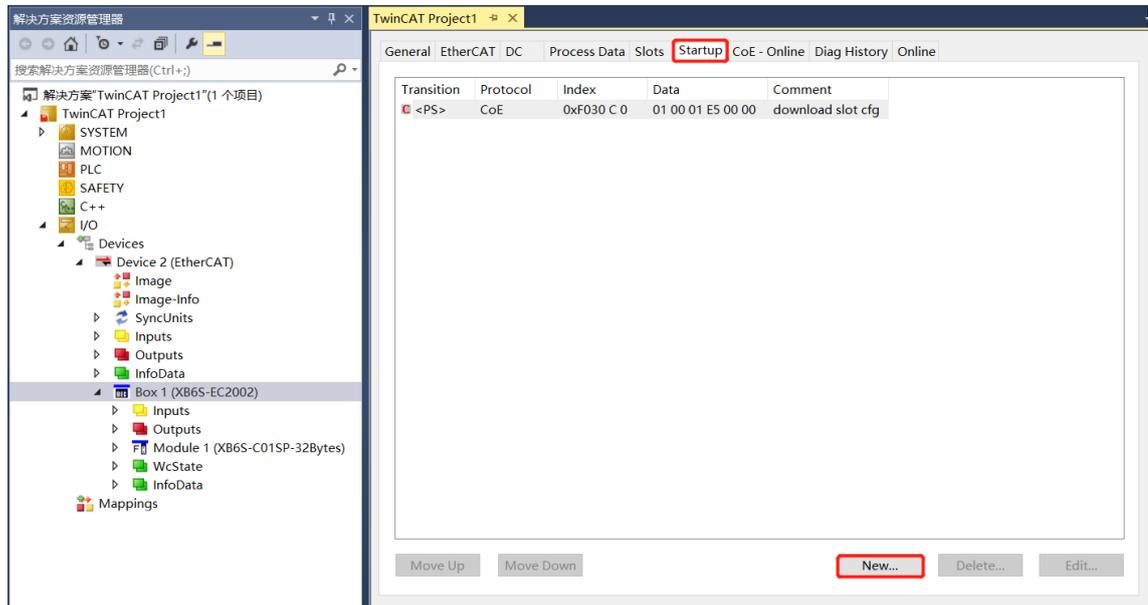


- d. 扫描到设备后，左侧导航树可以看到 Box1 (XB6S-EC2002) 和 Module1 (XB6S-C01SP-32Bytes) ，在 “Online” 处可以看到 TwinCAT 在 “OP” 状态，可以观察到从站设备 RUN 灯常亮，如下图所示。

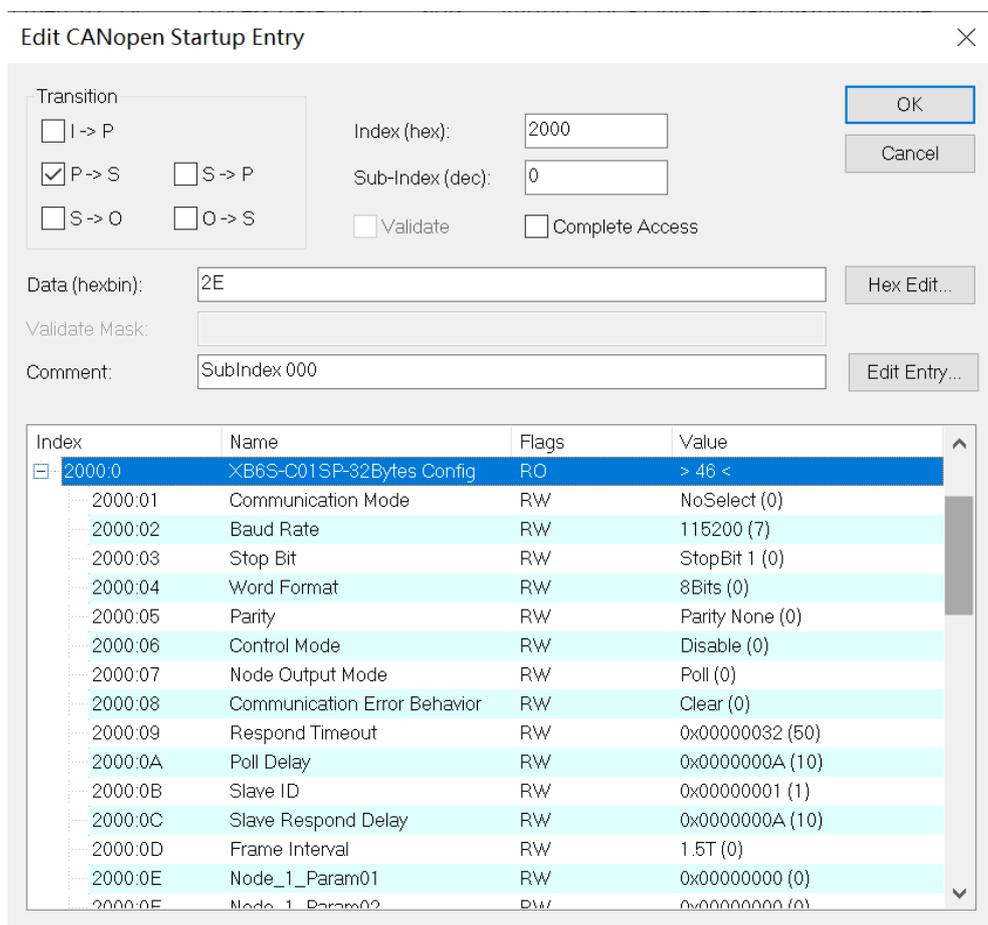


5、验证基本功能

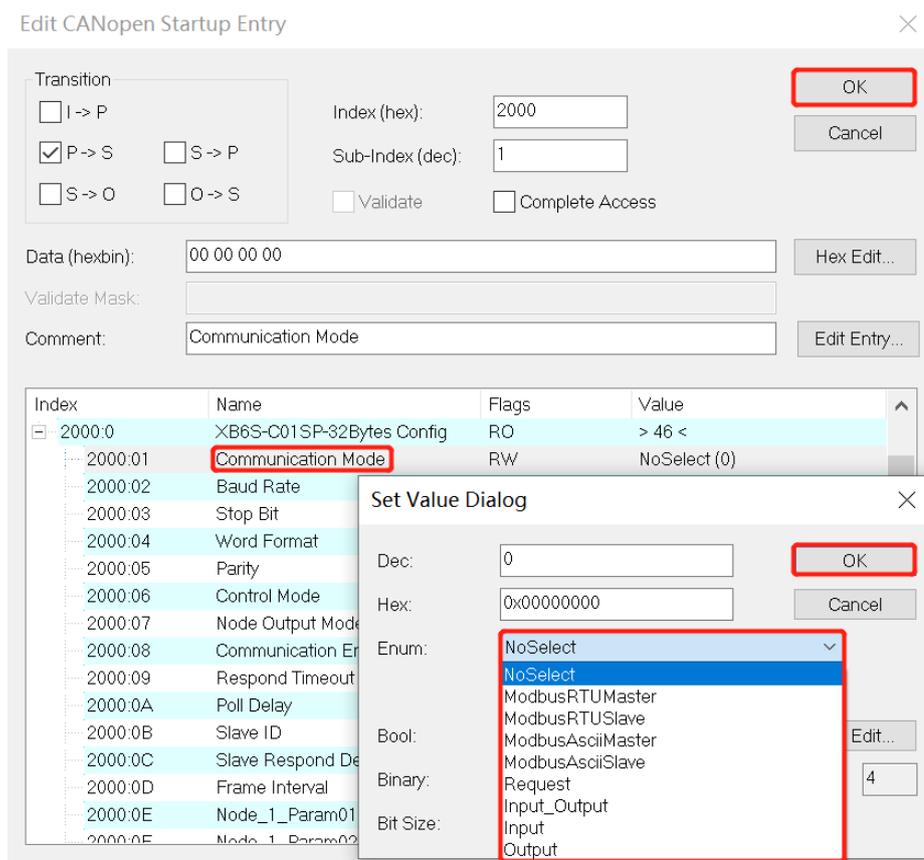
- a. 单击左侧导航树“Box1 -> Startup -> New”可以进入配置参数编辑页面，如下图所示。



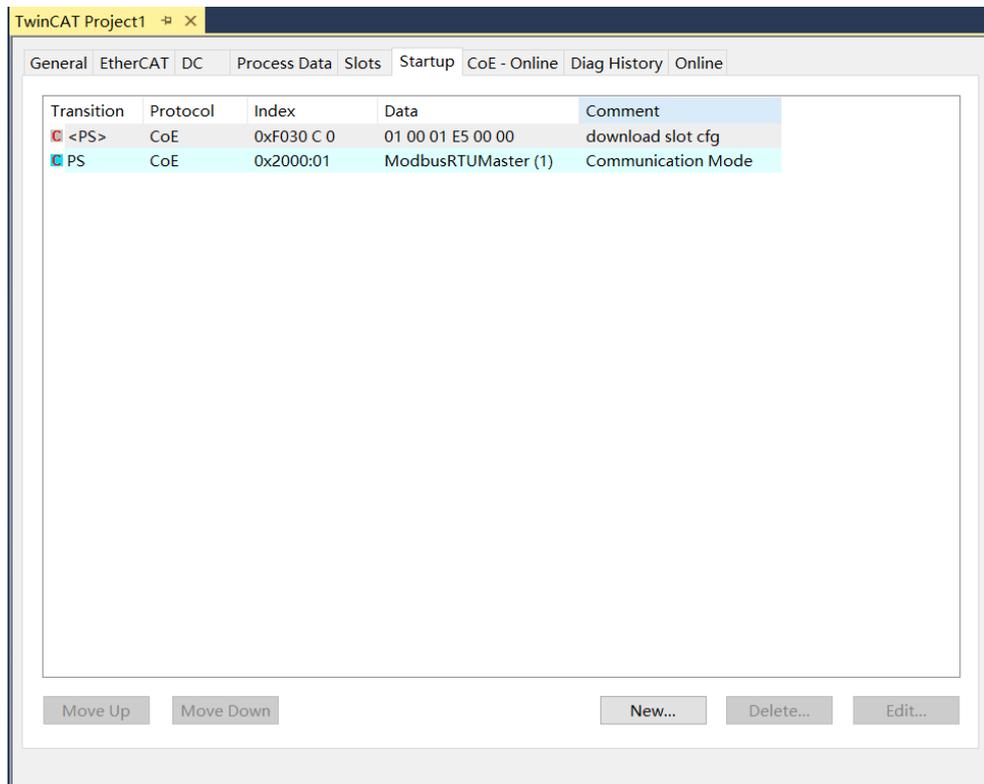
- b. 在 Edit CANopen Startup Entry 弹窗中，单击 Index 2000:0 前面的“+”，展开配置参数菜单，可以看到 45 个配置参数，点击任意一个参数，可以设置相关的配置，如下图所示。



- c. 例如修改通讯模式参数，可以双击“Communication Mode”，修改参数值，如下图所示。



- d. 参数修改完成后，可在 Startup 下方看到修改后的参数项和参数值，如下图所示。参数设置完成后，需进行 Reload 操作及模块重新上电，实现主站自动下发参数设定。



e. 左侧导航树 “Module 1 -> Inputs” 显示通讯模块的上行数据，用于监视模块的状态，如下图所示。

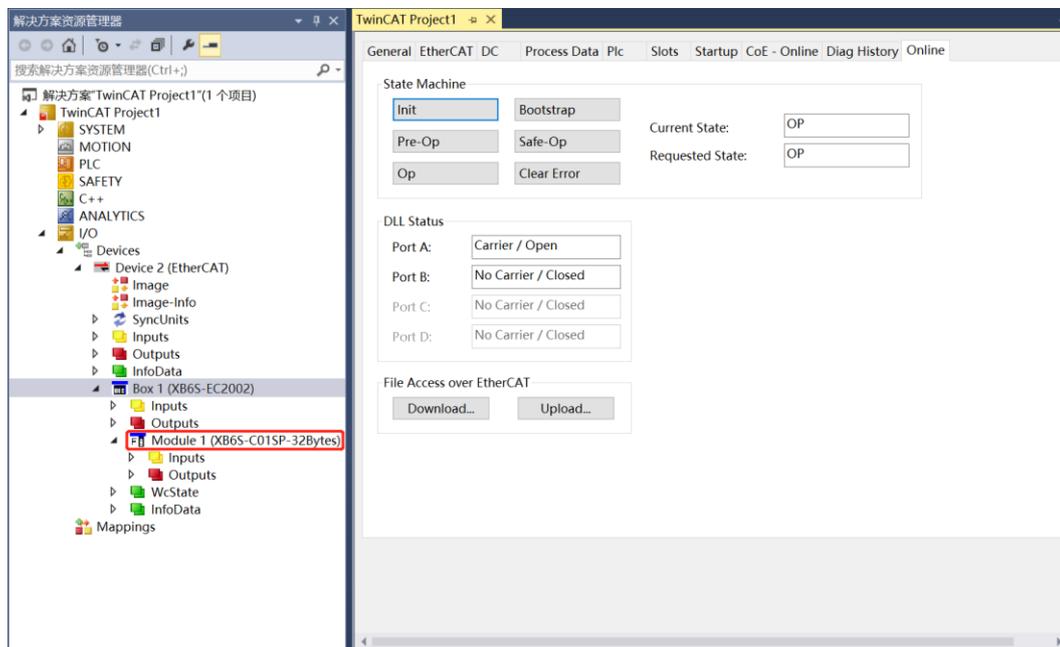
Name	Online	Type	Size	>Address	In/Out	User ID	Linked to
RX 1	0	USINT	1.0	41.0	Input	0	
RX 2	0	USINT	1.0	42.0	Input	0	
RX 3	0	USINT	1.0	43.0	Input	0	
RX 4	0	USINT	1.0	44.0	Input	0	
RX 5	0	USINT	1.0	45.0	Input	0	
RX 6	0	USINT	1.0	46.0	Input	0	
RX 7	0	USINT	1.0	47.0	Input	0	
RX 8	0	USINT	1.0	48.0	Input	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. 左侧导航树 “Module 1 -> Outputs” 显示通讯模块的下行数据，用于监视模块的输出状态，如下图所示。

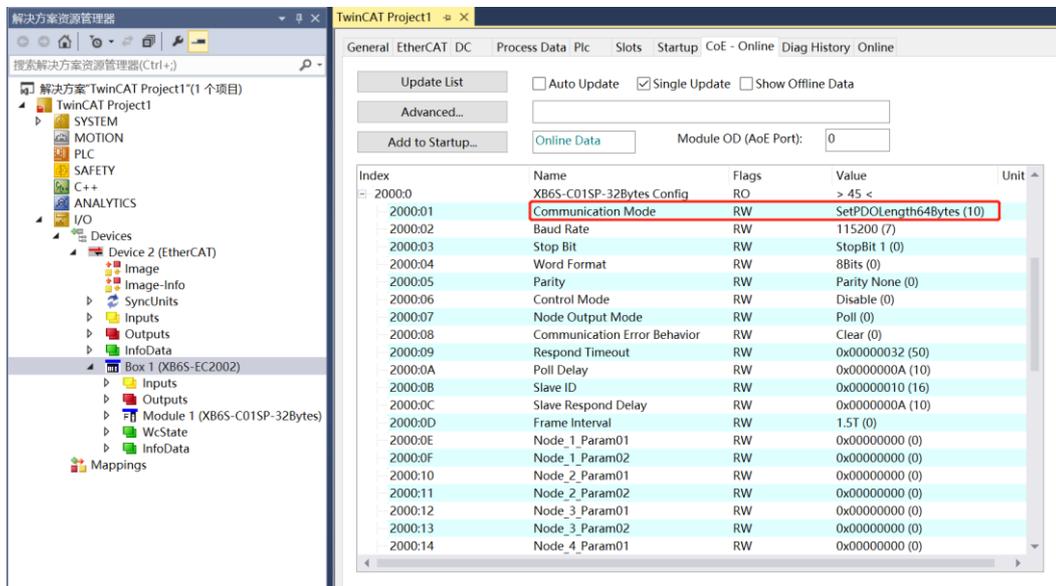
Name	Online	Type	Size	>Address	In/Out	User ID	Linked to
TX 1	0	USINT	1.0	41.0	Output	0	
TX 2	0	USINT	1.0	42.0	Output	0	
TX 3	0	USINT	1.0	43.0	Output	0	
TX 4	0	USINT	1.0	44.0	Output	0	
TX 5	0	USINT	1.0	45.0	Output	0	
TX 6	0	USINT	1.0	46.0	Output	0	
TX 7	0	USINT	1.0	47.0	Output	0	
TX 8	0	USINT	1.0	48.0	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、切换 PDO 数据长度功能示例

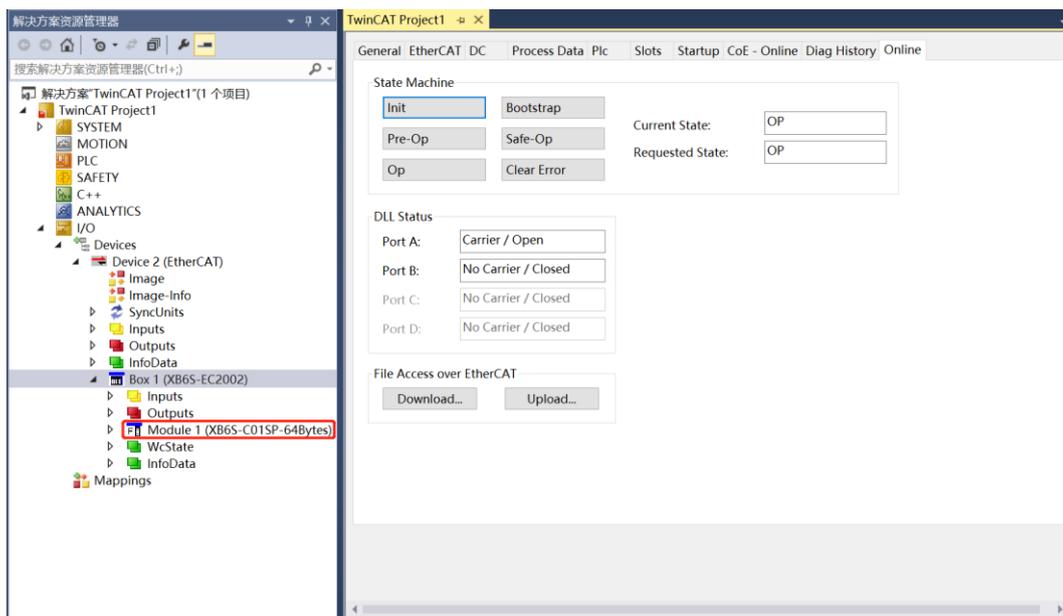
- a. 右击 “I/O -> Devices -> Scan” 扫描设备，左侧导航树可以看到扫描的设备为 XB6S-C01SP-32Bytes，如下图所示。



- b. 单击左侧导航树 “Box1 -> CoE-Online ” 可以进入配置参数编辑页面，通讯模式选择 10 即设置 PDO 数据长度为 64Bytes，如下图所示。



- c. 此时模块通讯模式处于 No_Select 状态，需要重新上下电模块后扫描设备，扫描的设备为 XB6S-C01SP-64Bytes，如下图所示。



7、RTU 主站模式功能示例

示例 1：通过 Modbus Slave 调试软件等工具或设备验证模块 RTU 主站在 disable 模式下写 2 个保持寄存器的值。

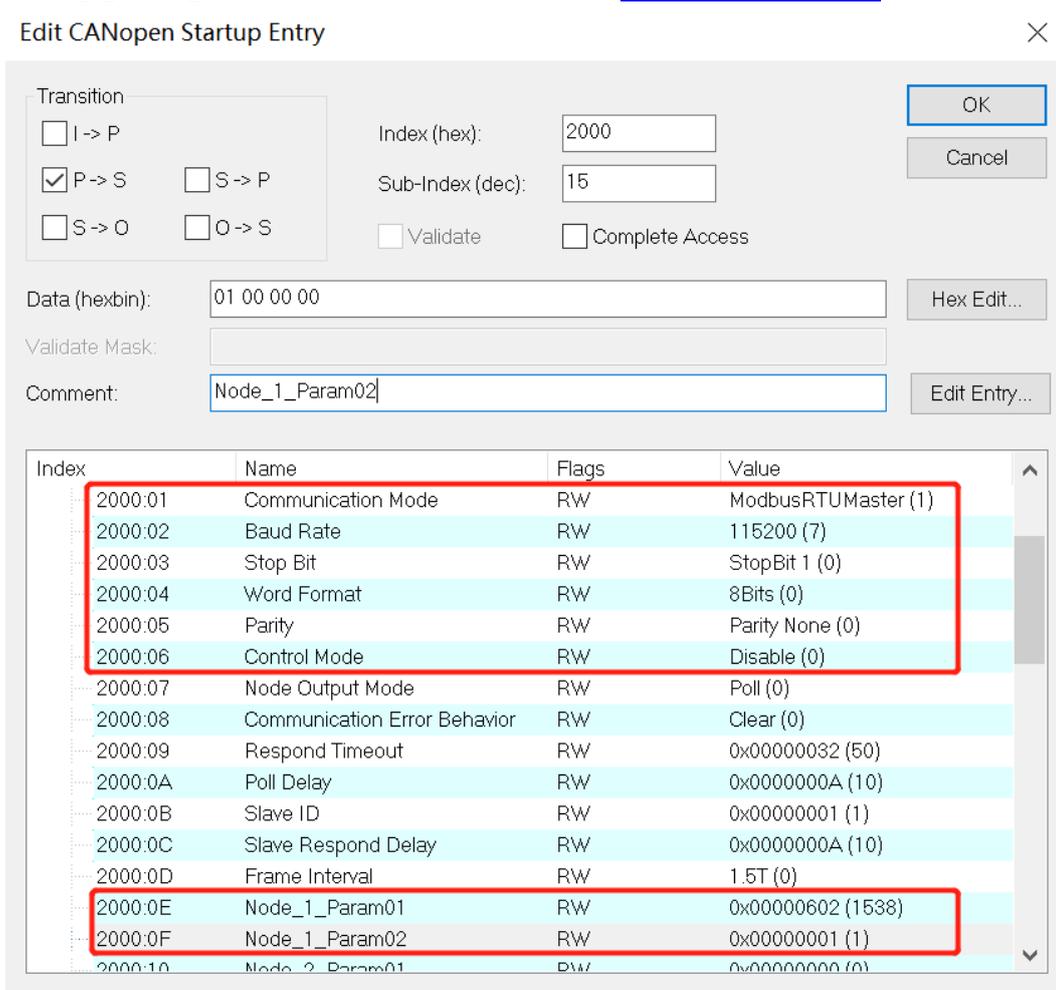
- a. 对配置参数进行配置，通讯模式选择 1 即 RTU 主站模式，如下图所示。

Communication Mode: 选择 ModbusRTUMaster;

Control Mode: 选择 Disable;

Node_1_Param_01: 配置 0x00000602, 配置参数详见 [6.2.1 Modbus 主站功能](#);

Node_1_Param_02: 配置 0x00000001, 配置参数详见 [6.2.1 Modbus 主站功能](#)。



- b. 在下行数据中，写入 2 个保持寄存器的值，如下图所示。

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. 通过调试软件可以看到接收的 2 个寄存器的值，如下图所示。

ID	Alias	F
0		257
1		257
2		
3		
4		
5		
6		
7		
8		
9		

For Help, press F1.

示例 2：通过 Modbus Slave 调试软件等工具或设备验证模块 RTU 主站在 Level 模式下读取 10 个保持寄存器。

- a. 对配置参数进行配置，通讯模式选择 1 即 RTU 主站模式，如下图所示。

Communication Mode: 选择 ModbusRTUMaster;

Control Mode: 选择 Level;

Node_1_Param_01: 配置 0x00000001, 配置参数详见 [6.2.8 控制与状态节点代码](#);

Node_2_Param_01: 配置 0x0000030A, 配置参数详见 [6.2.1 Modbus 主站功能](#);

Node_2_Param_02: 配置 0x00000001, 配置参数详见 [6.2.1 Modbus 主站功能](#)。

Edit CANopen Startup Entry

Transition
 I->P
 P->S S->P
 S->O O->S

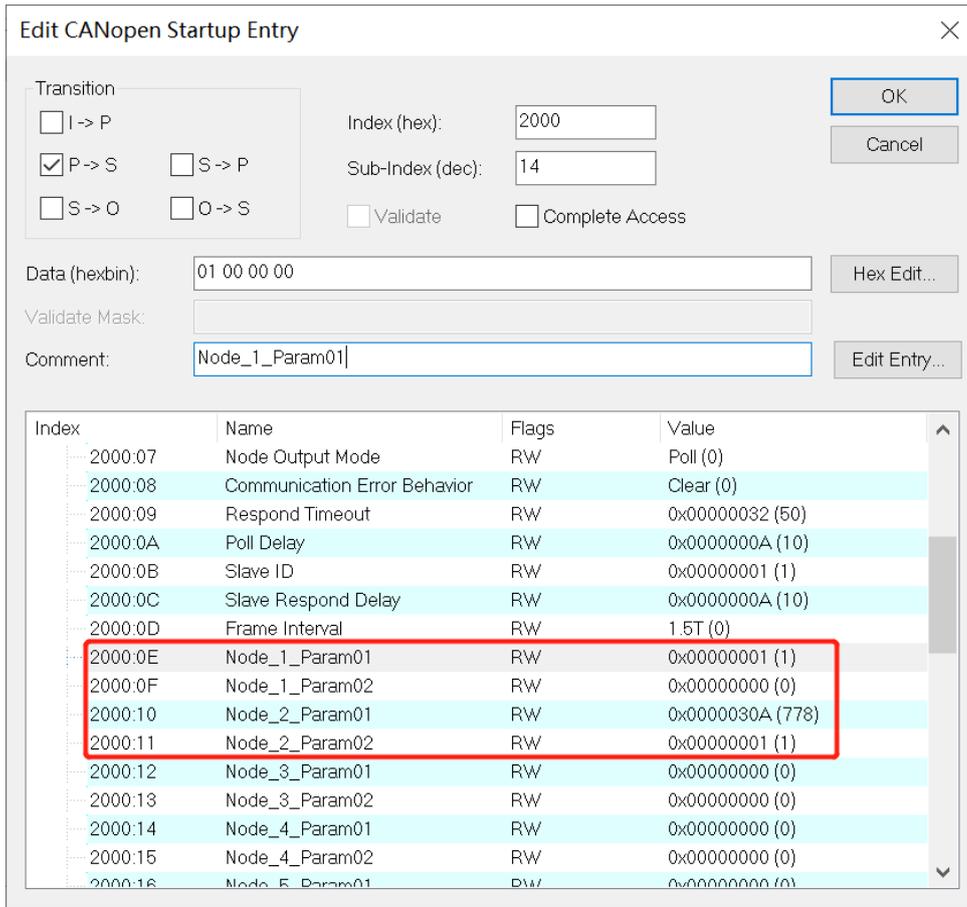
Index (hex): 2000
 Sub-Index (dec): 1
 Validate Complete Access

Data (hexbin): 01 00 00 00 Hex Edit...

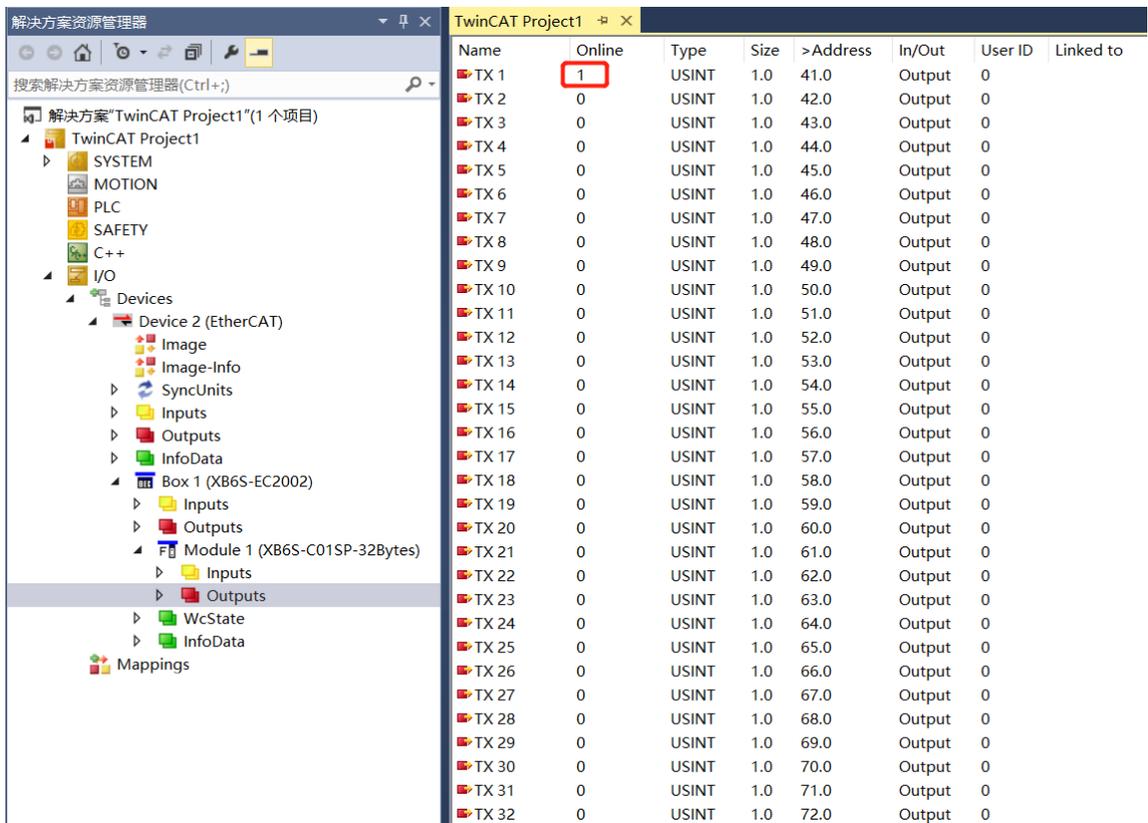
Validate Mask:

Comment: Communication Mode Edit Entry...

Index	Name	Flags	Value
2000:01	Communication Mode	RW	ModbusRTUMaster (1)
2000:02	Baud Rate	RW	115200 (7)
2000:03	Stop Bit	RW	StopBit 1 (0)
2000:04	Word Format	RW	8Bits (0)
2000:05	Parity	RW	Parity None (0)
2000:06	Control Mode	RW	Level (1)
2000:07	Node Output Mode	RW	Poll (0)
2000:08	Communication Error Behavior	RW	Clear (0)
2000:09	Respond Timeout	RW	0x00000032 (50)
2000:0A	Poll Delay	RW	0x0000000A (10)
2000:0B	Slave ID	RW	0x00000001 (1)
2000:0C	Slave Respond Delay	RW	0x0000000A (10)
2000:0D	Frame Interval	RW	1.5T (0)
2000:0E	Node_1_Param01	RW	0x00000602 (1538)
2000:0F	Node_1_Param02	RW	0x00000001 (1)
2000:10	Node_2_Param01	RW	0x00000000 (0)



b. 在下行数据中将控制字置为 1，并打开调试软件发送数据，如下图所示。



c. 在上行数据中可以看到接收到的数据，如下图所示。

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	

8、Freeport_Input 功能示例

示例：通过串口调试助手等工具或设备利用 Freeport_Input 在 Level 模式下接收一个 8 字节数据。

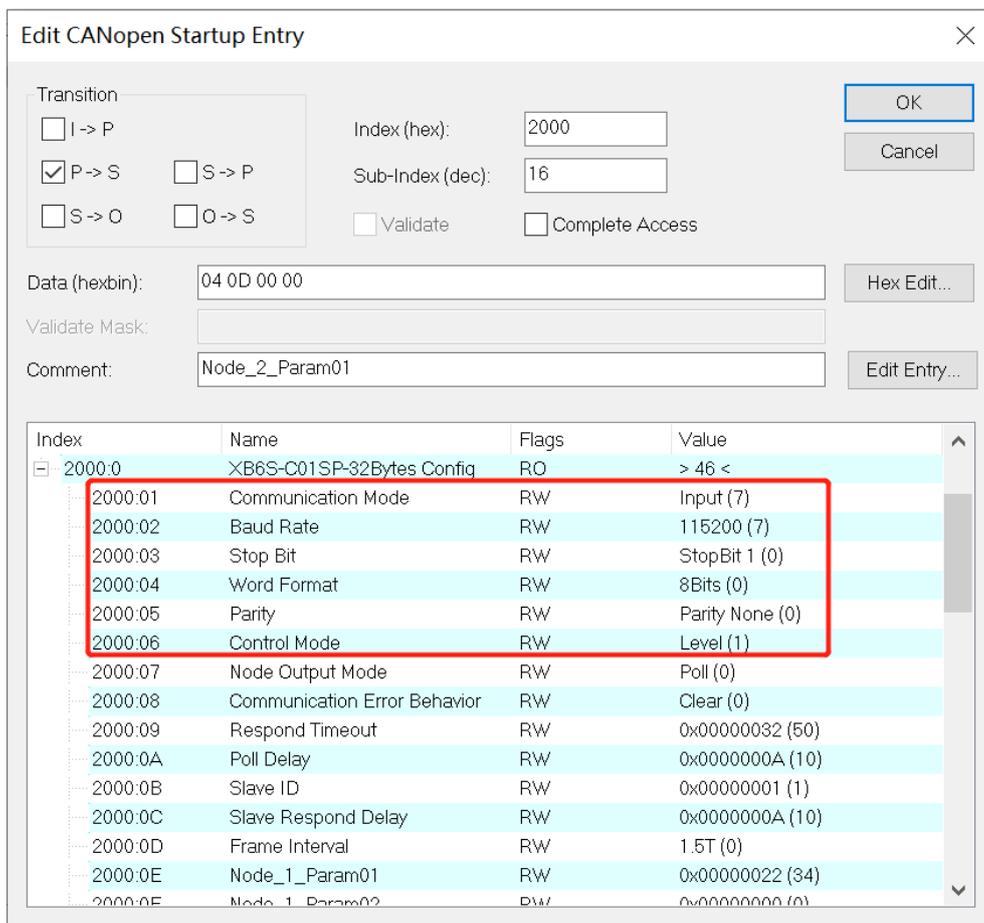
- a. 对配置参数进行配置，通讯模式选择 7 即 Input 模式，如下图所示。

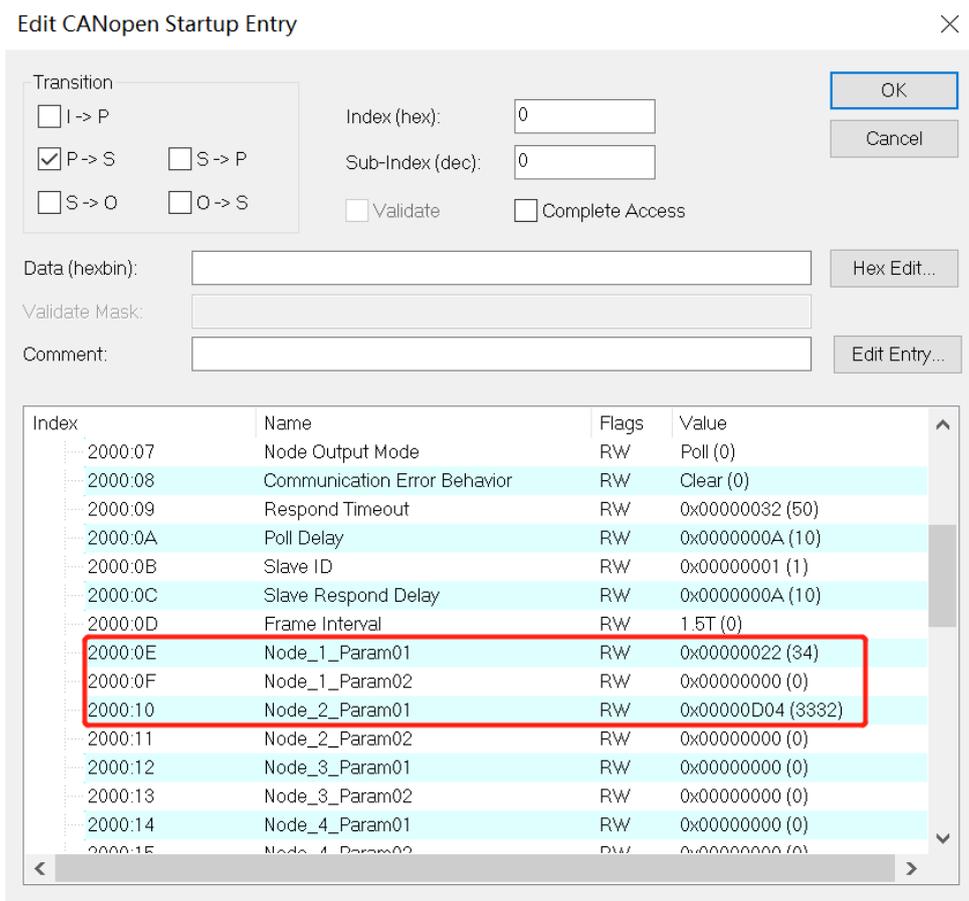
Communication Mode: 选择 Input;

Control Mode: 选择 Level;

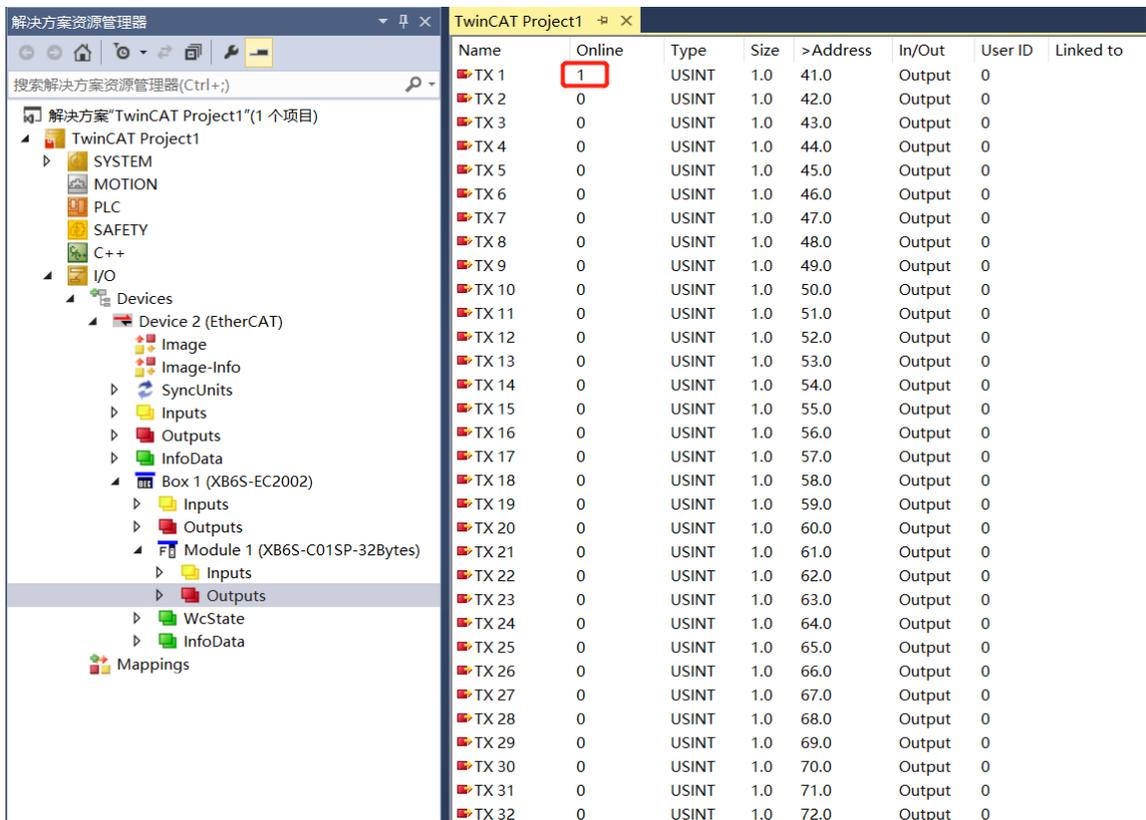
Node_1_Param_01: 配置 0x00000022, 配置参数详见 [6.2.8 控制与状态节点代码](#);

Node_2_Param_01: 配置 0x00000D04, 配置参数详见 [6.2.6 Freeport 自由口功能](#)。





- b. 在下行数据中将控制字置为 1，并打开串口调试助手发送数据，如下图所示。



- c. 上行数据可以看到接收的数据长度为 8，接收的数据计数为 1，接收的 8 个字节的数 (RX7~RX14)，如下图所示。

Name	Online	Type	Size	>Address	In/Out	User ID	Linked to
RX 1	0	USINT	1.0	41.0	Input	0	
RX 2	8	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	1	USINT	1.0	45.0	Input	0	
RX 6	0	USINT	1.0	46.0	Input	0	
RX 7	1	USINT	1.0	47.0	Input	0	
RX 8	2	USINT	1.0	48.0	Input	0	
RX 9	3	USINT	1.0	49.0	Input	0	
RX 10	4	USINT	1.0	50.0	Input	0	
RX 11	5	USINT	1.0	51.0	Input	0	
RX 12	1	USINT	1.0	52.0	Input	0	
RX 13	2	USINT	1.0	53.0	Input	0	
RX 14	3	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	

6.4.2 在 Sysmac Studio 软件环境下的应用

1、准备工作

● 硬件环境

- 模块型号 XB6S-C01SP
- EtherCAT 耦合器, 端盖
本说明以 XB6S-EC2002 耦合器为例
- 计算机一台, 预装 Sysmac Studio 软件
- 欧姆龙 PLC 一台, 本说明以型号 NX1P2-9024DT 为例
- EtherCAT 专用屏蔽电缆
- 开关电源一台
- 设备配置文件

配置文件获取地址: <https://www.solidotech.com/documents/configfile>

● 硬件组态及接线

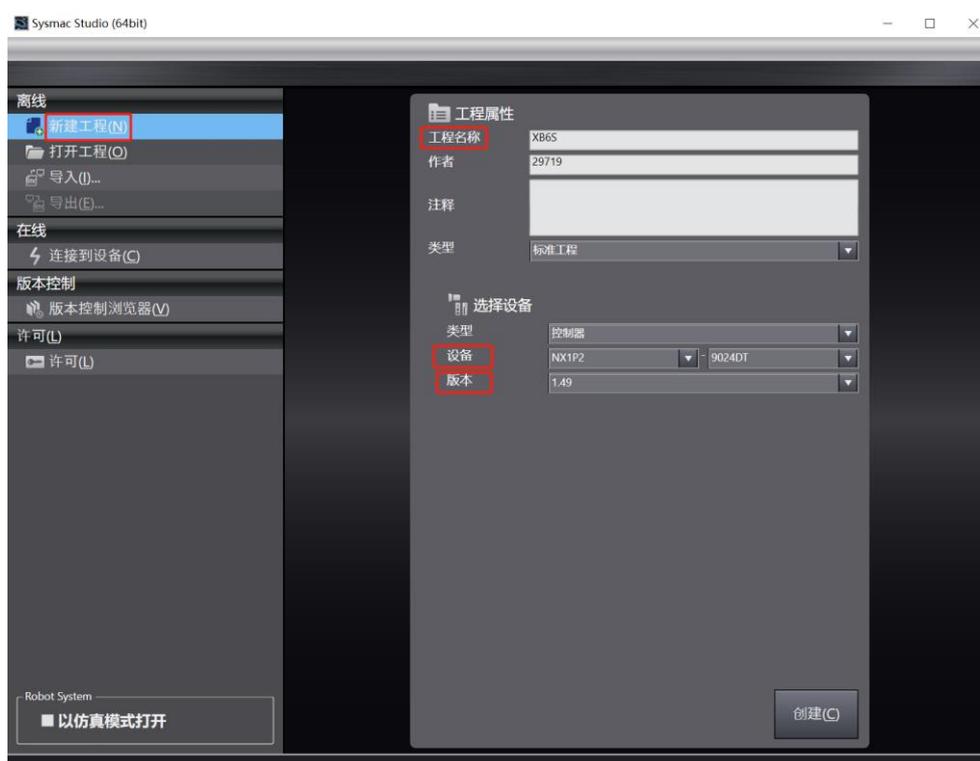
请按照“[4 安装和拆卸](#)”和“[5 接线](#)”要求操作

● 计算机 IP 要求

设置电脑的 IP 地址和 PLC 的 IP 地址, 确保其在同一网段。

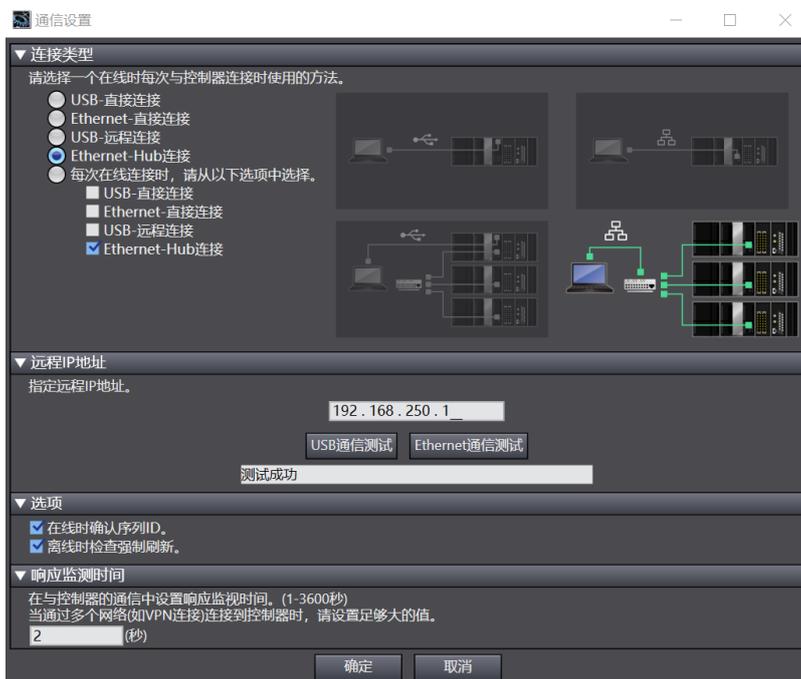
2、新建工程

- 打开 Sysmac Studio 软件, 单击“新建工程”。



- 工程名称: 自定义。
- 选择设备: “设备”选择对应的 PLC 型号, “版本”选择 PLC 对应的版本号。

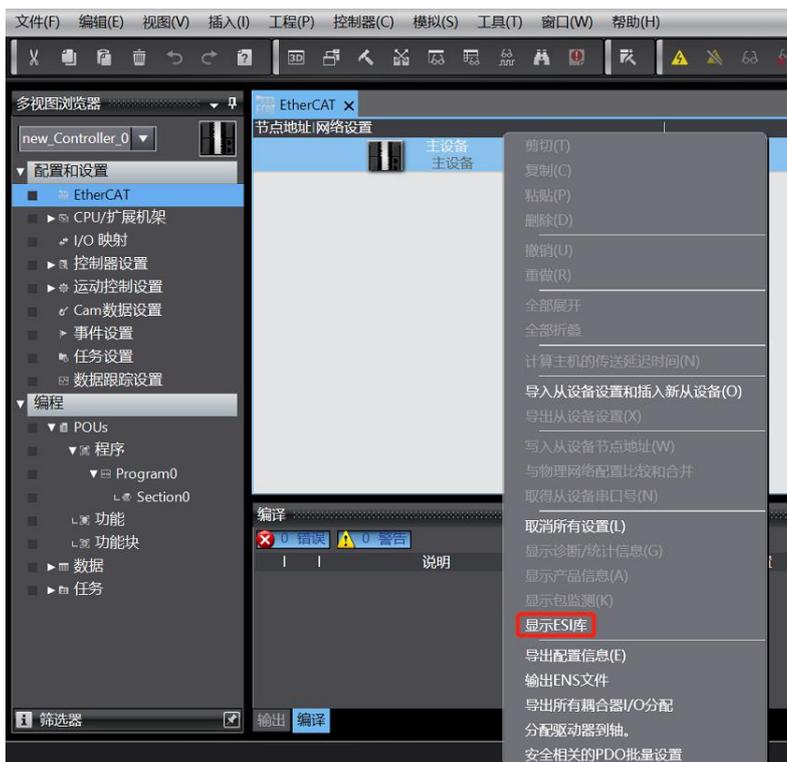
- b. 工程属性输入完成后，单击“创建”。
- c. 单击菜单栏“控制器 -> 通信设置”，选择在线时每次与控制器连接时使用的方法，输入“远程 IP 地址”，如下图所示。



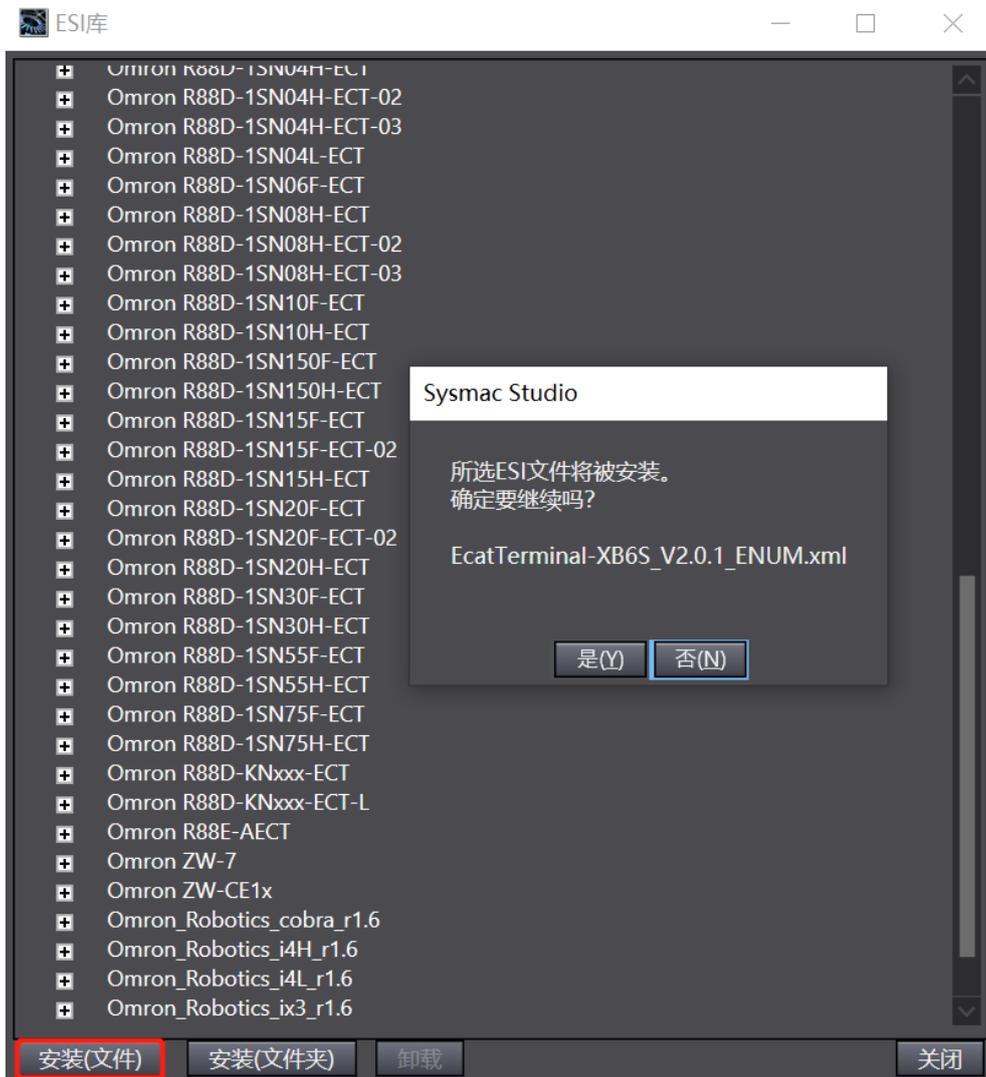
- d. 单击“Ethernet 通信测试”，系统显示测试成功。

3. 安装 XML 文件

- a. 在左侧导航树展开“配置和设置”，双击“EtherCAT”。
- b. 右击“主设备”，选择“显示 ESI 库”，如下图所示。



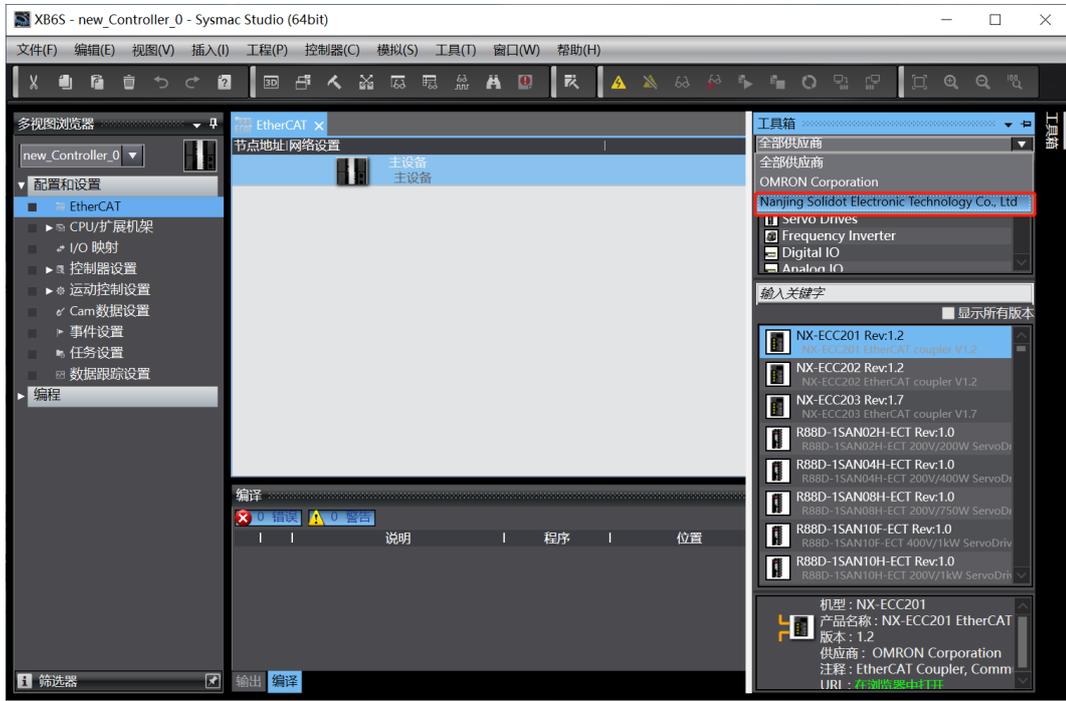
- c. 在弹出的“ESI库”窗口中单击“安装(文件)”，选择模块的XML文件路径，单击“是”完成安装，如下图所示。



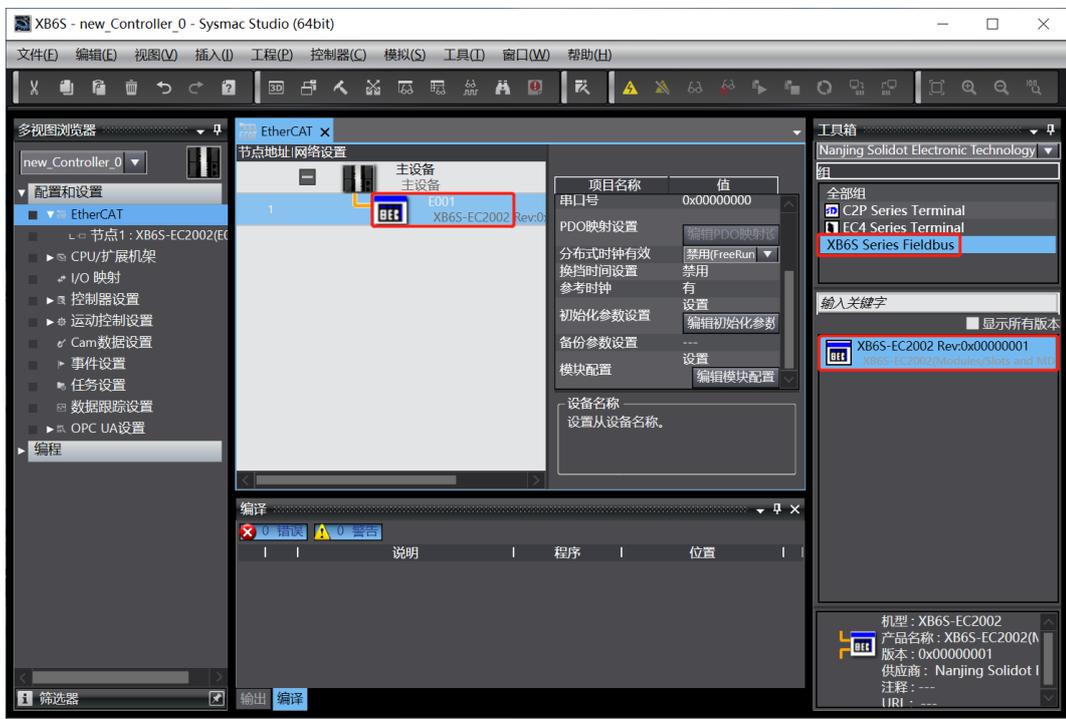
4、添加设备

添加设备有在线扫描和离线添加两种方式，本说明以离线添加为例进行介绍。

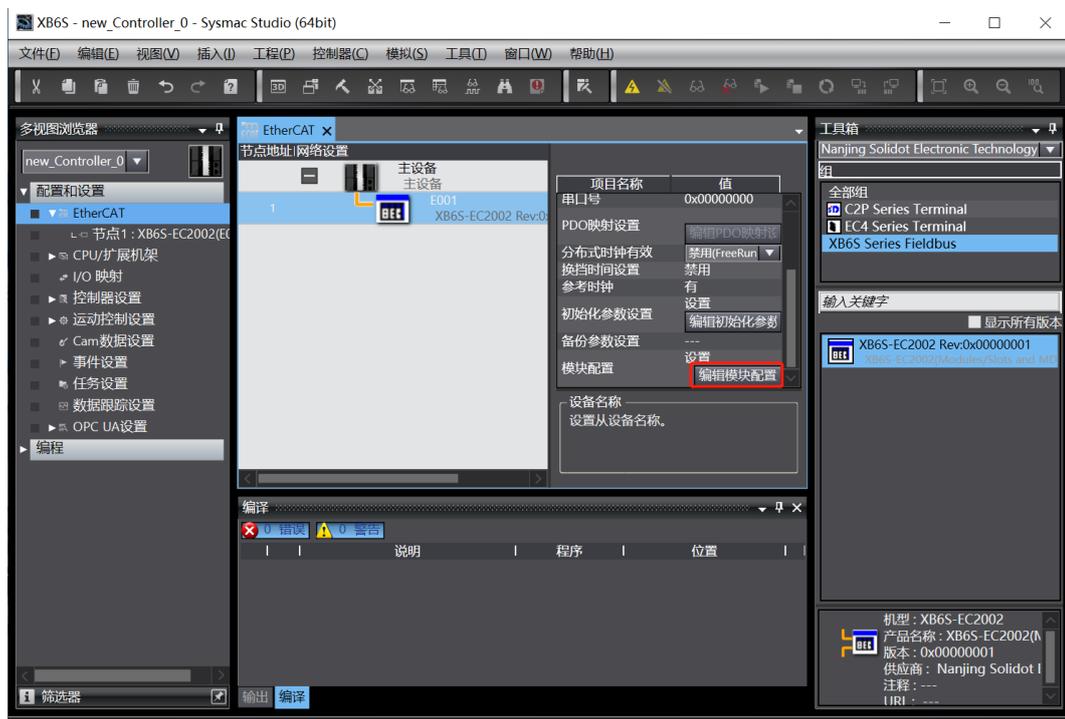
- a. 在右侧“工具箱”栏下，单击展开全部供应商，选择“Nanjing Solidot Electronic Technology Co., Ltd.”，如下图所示。



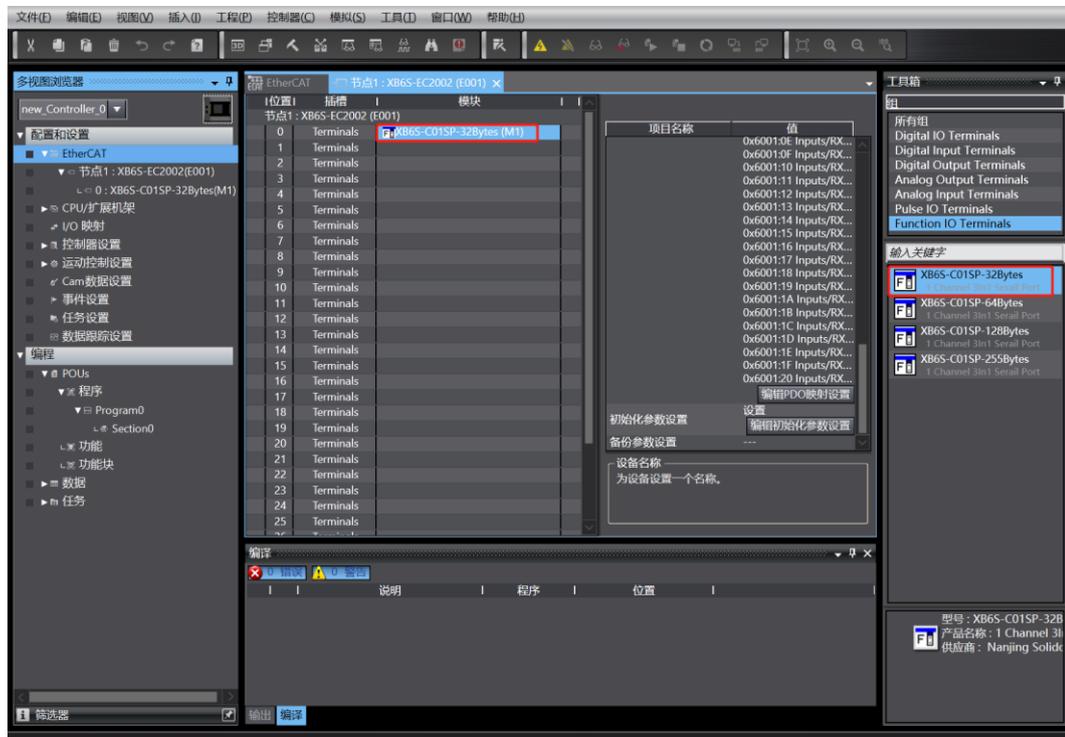
- b. 单击选择 XB6S Series Fieldbus，双击 XB6S-EC2002 耦合器模块，添加从设备，如下图所示。



- c. 在 EtherCAT 主页面，选中刚添加的 XB6S-EC2002 耦合器模块，选择“编辑模块配置”，如下图所示。

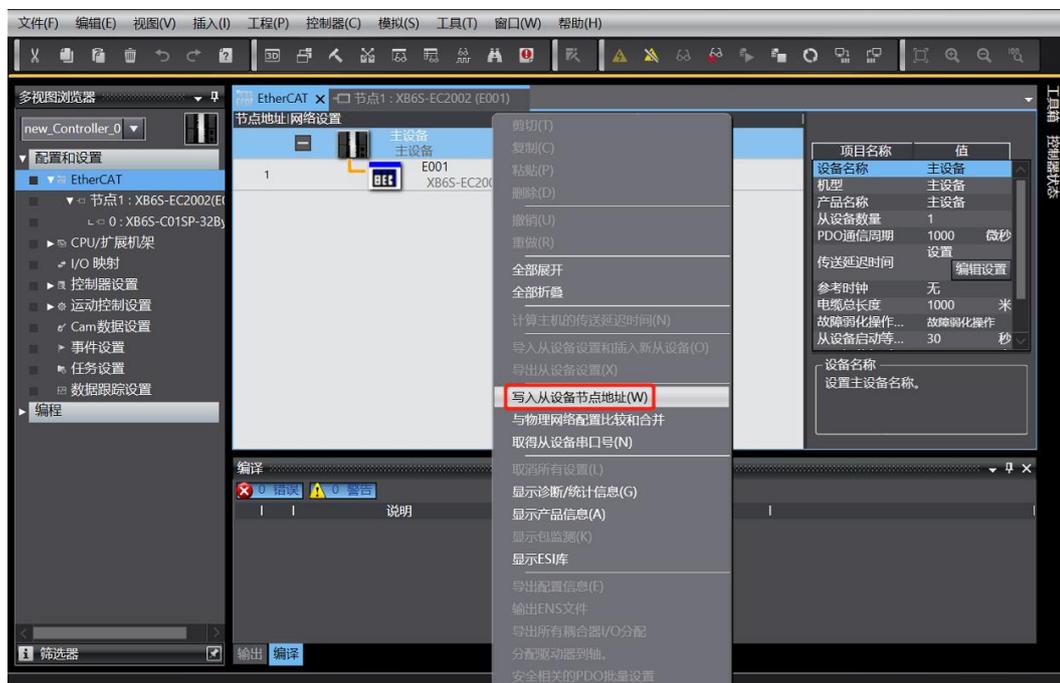


- d. 光标定位到“模块”中，在右侧工具箱模块列表中单击模块，按 I/O 模块组态的顺序，逐个添加 I/O 模块。注意：顺序及型号必须与物理拓扑一致！

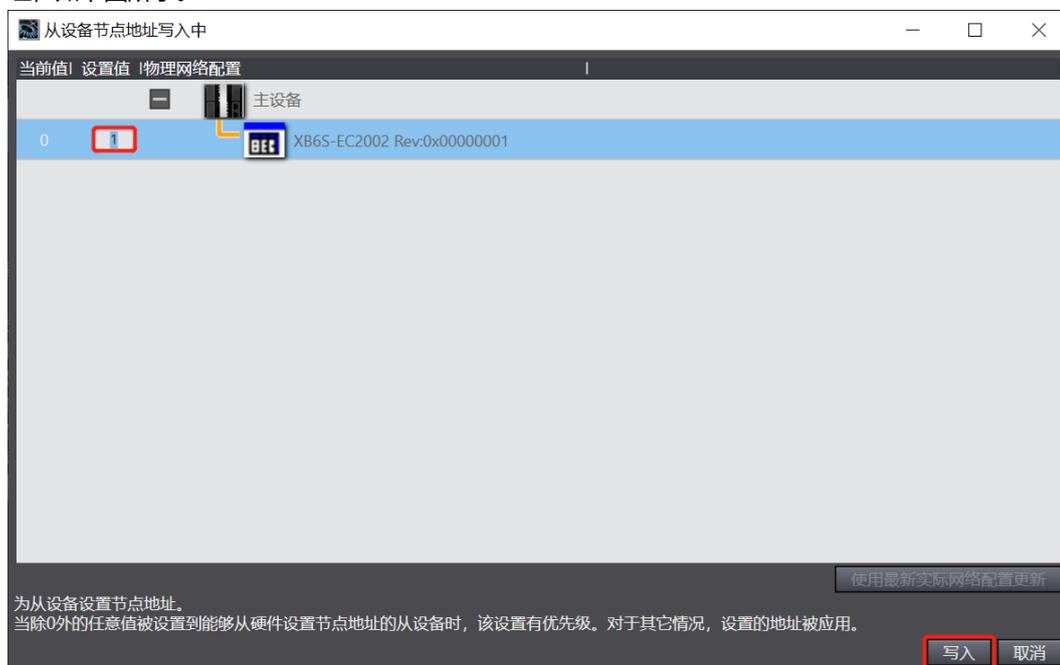


5、设置节点地址

- a. 单击菜单栏“控制器 -> 在线”，将控制器转至在线状态。右击主设备，单击选择“写入从设备节点地址”，如下图所示。



- b. 在设置节点地址的窗口中，单击设置值下的数值，输入节点地址，单击“写入”，更改从设备节点地址，如下图所示。

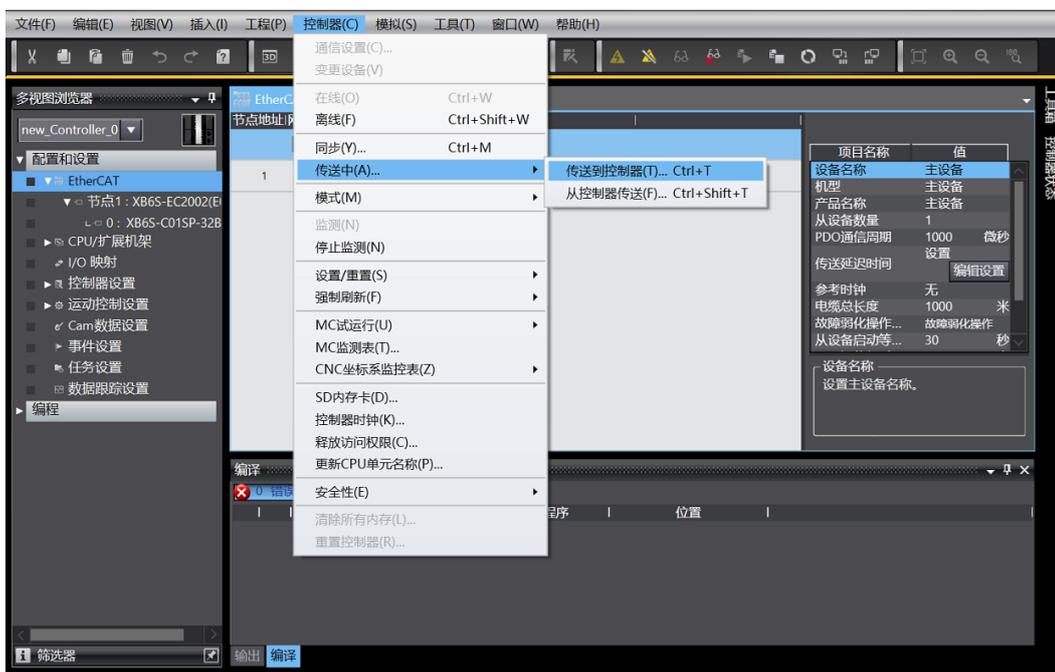


- c. 写入之后，弹出重新上电提示，如下图所示，单击“写入”，再根据提示重启从设备电源。

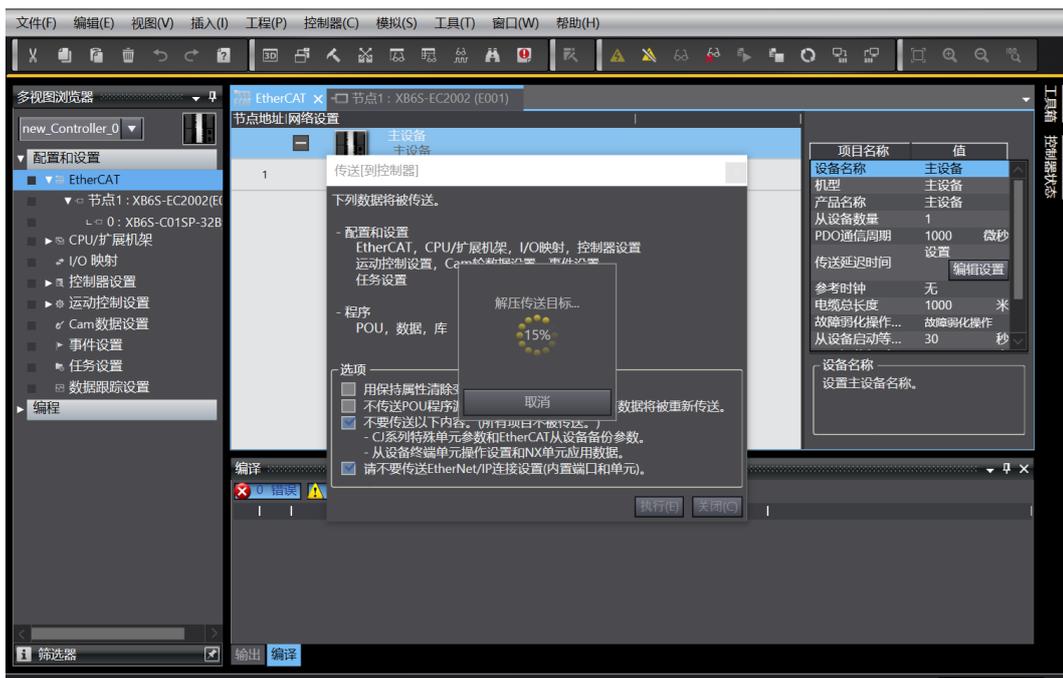


6、将组态下载到 PLC

- a. 单击菜单栏“控制器 -> 传送中 (A) -> 传送到控制器 (T)”按钮，如下图所示。

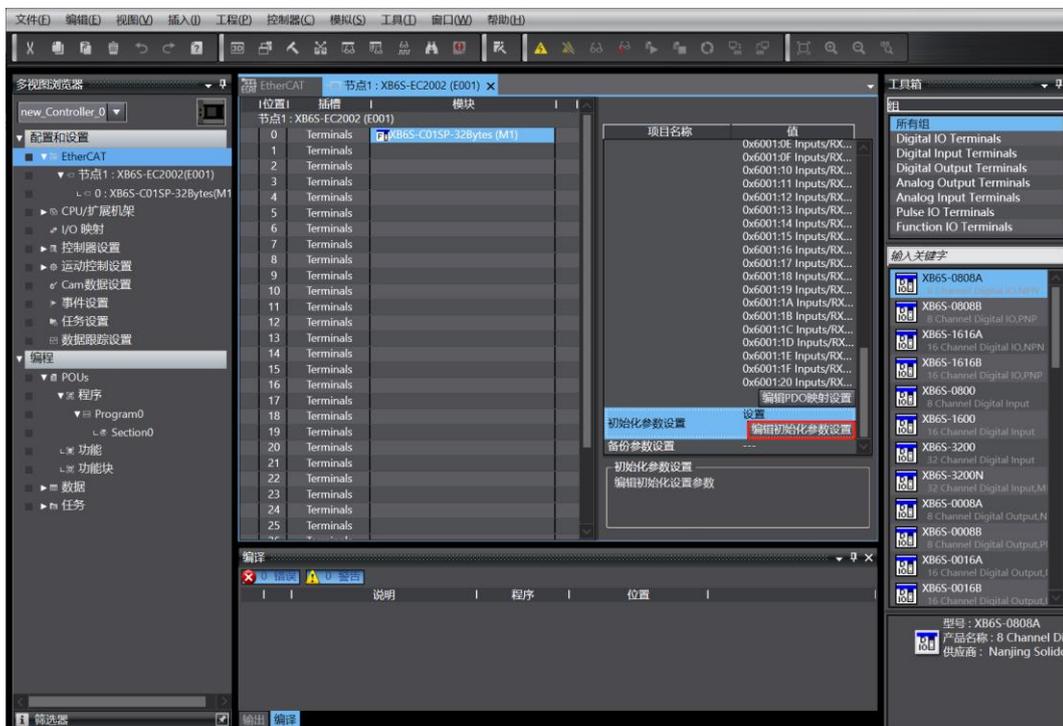


- b. 将组态下载到 PLC，弹出传送确认弹窗，单击“执行”，后续弹窗依次单击“是/确定”，如下图所示，下载完成后，需要重新上电。



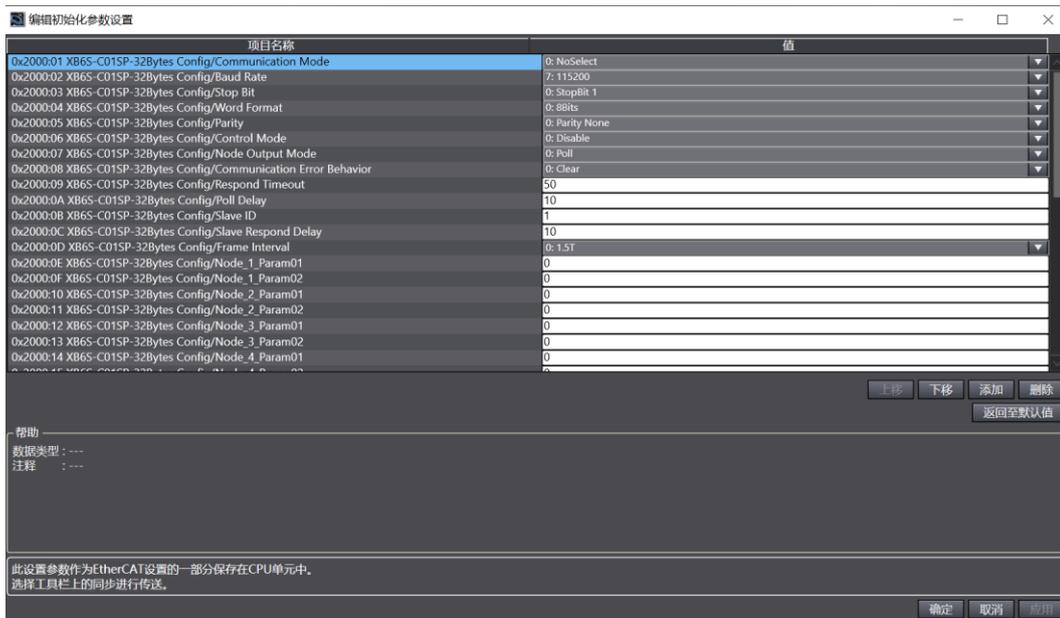
7、参数设置

- a. 将组态切换到离线状态，在节点 1 编辑模块配置页面，选择 XB6S-C01SP-32Bytes 模块，单击“编辑初始化参数设置”，如下图所示。

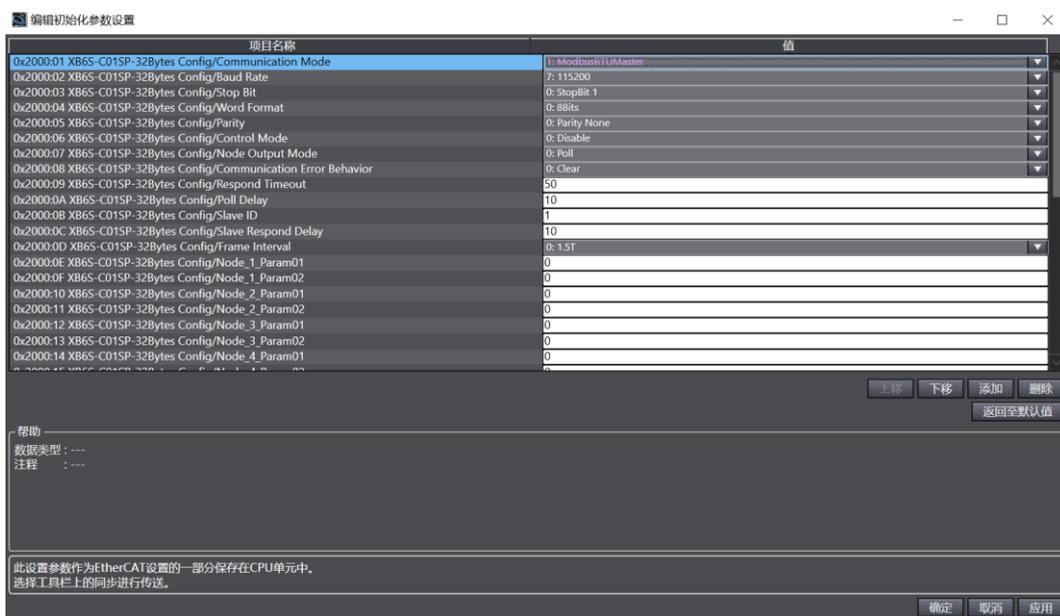


注：若 PLC 固件版本过低，需要用 EC_CoESDOWrite、EC_CoESDORed 指令进行 SDO 地址的写入和读取。

- b. 在 XB6S-C01SP 参数设置页面，可以看到 45 个配置参数，点击任意一个参数，可以设置相关的配置，如下图所示。

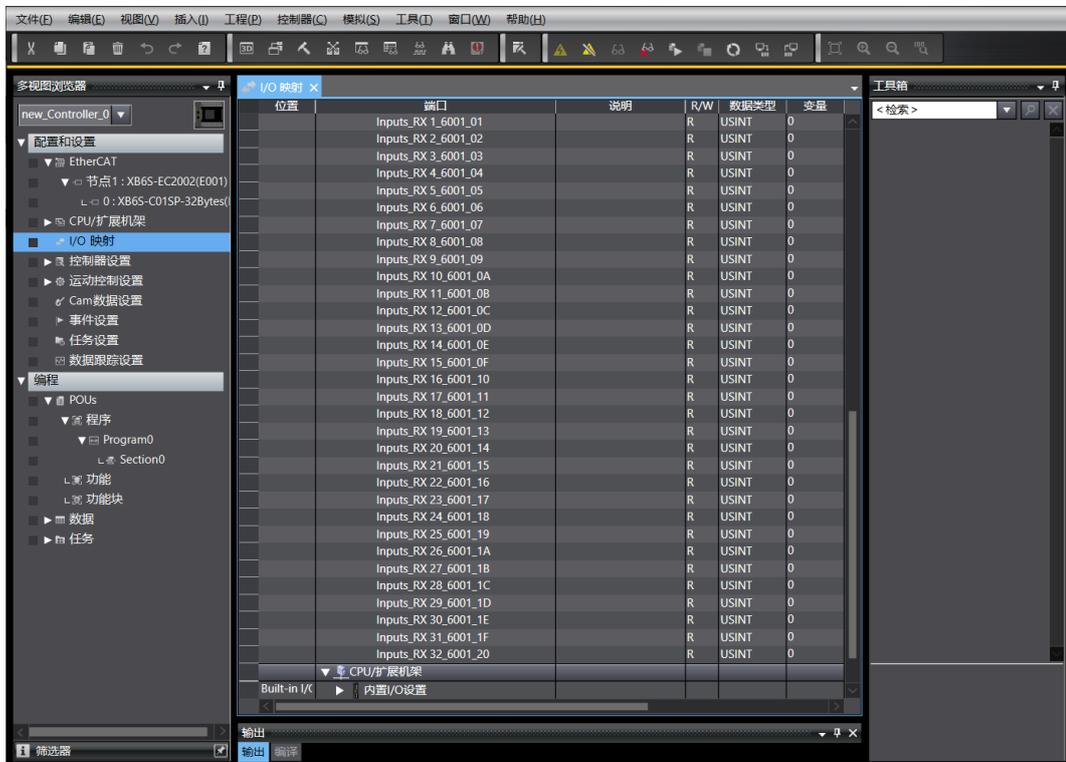
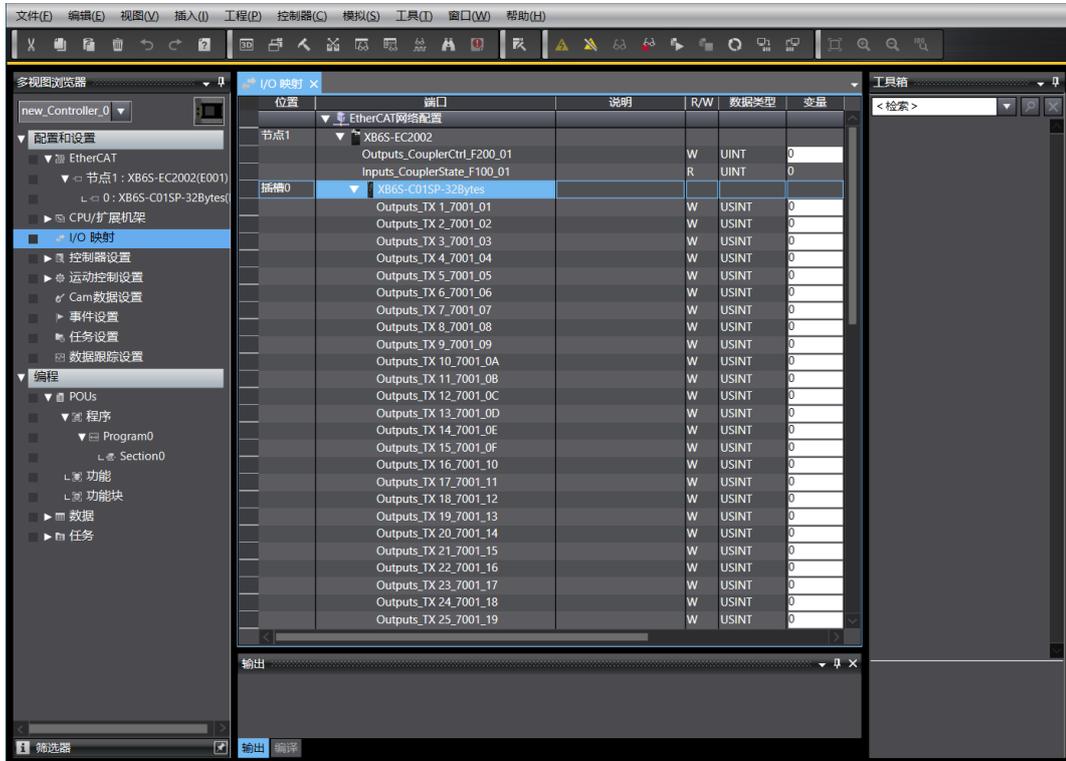


- c. 例如修改通讯模式参数，可以单击“Communication Mode”，修改参数值，如下图所示。参数全部配置完成后，需重新下载程序至 PLC 中，PLC 与模块需要重新上电。



8、I/O 功能

- a. 在左侧导航树中双击“I/O 映射”，可以看到拓扑中模块的映射表，从而对通讯模块输入输出值进行监控，如下图所示。



9、RTU 主站模式功能示例

示例 1：通过 Modbus Slave 调试软件等工具或设备验证模块 RTU 主站在 disable 模式下写 2 个保持寄存器的值。

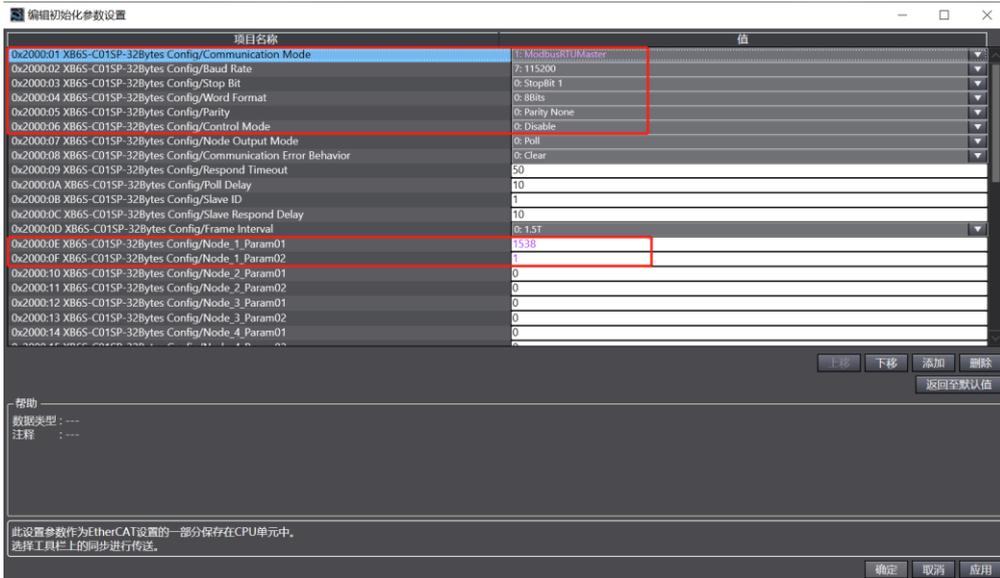
- a. 对配置参数进行配置，通讯模式选择 1 即 RTU 主站模式，如下图所示。

Communication Mode: 选择 ModbusRTUMaster;

Control Mode: 选择 Disable;

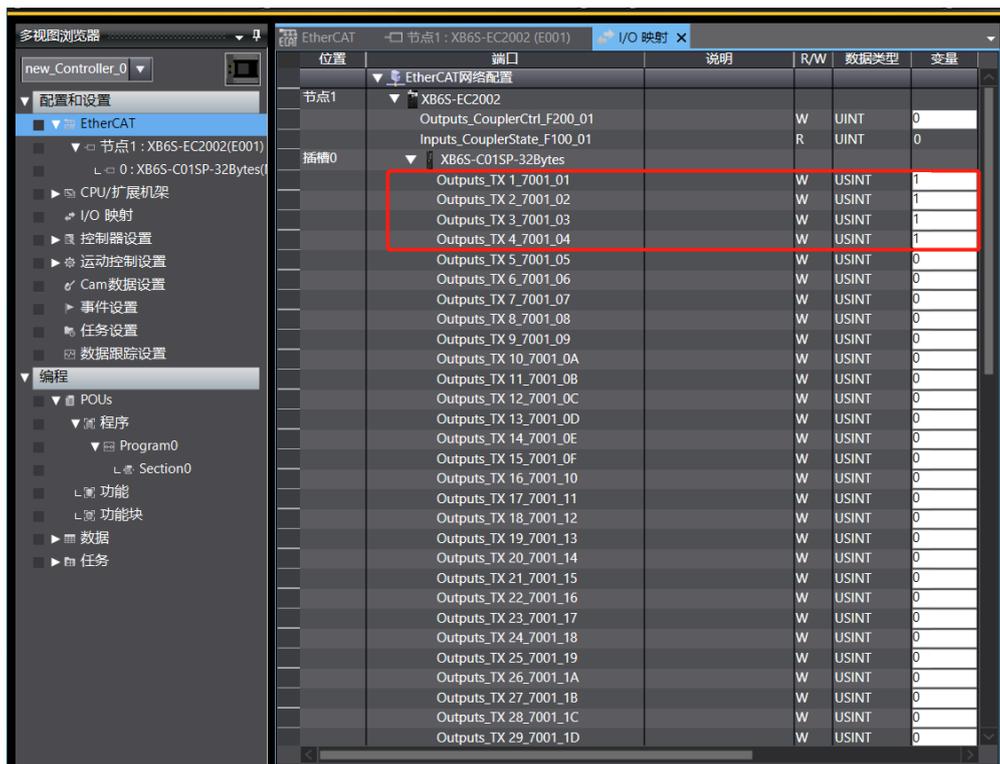
Node_1_Param_01: 配置 1538 (0x00000602), 配置参数详见 [6.2.1 Modbus 主站功能](#);

Node_1_Param_02: 配置 1 (0x00000001), 配置参数详见 [6.2.1 Modbus 主站功能](#)。

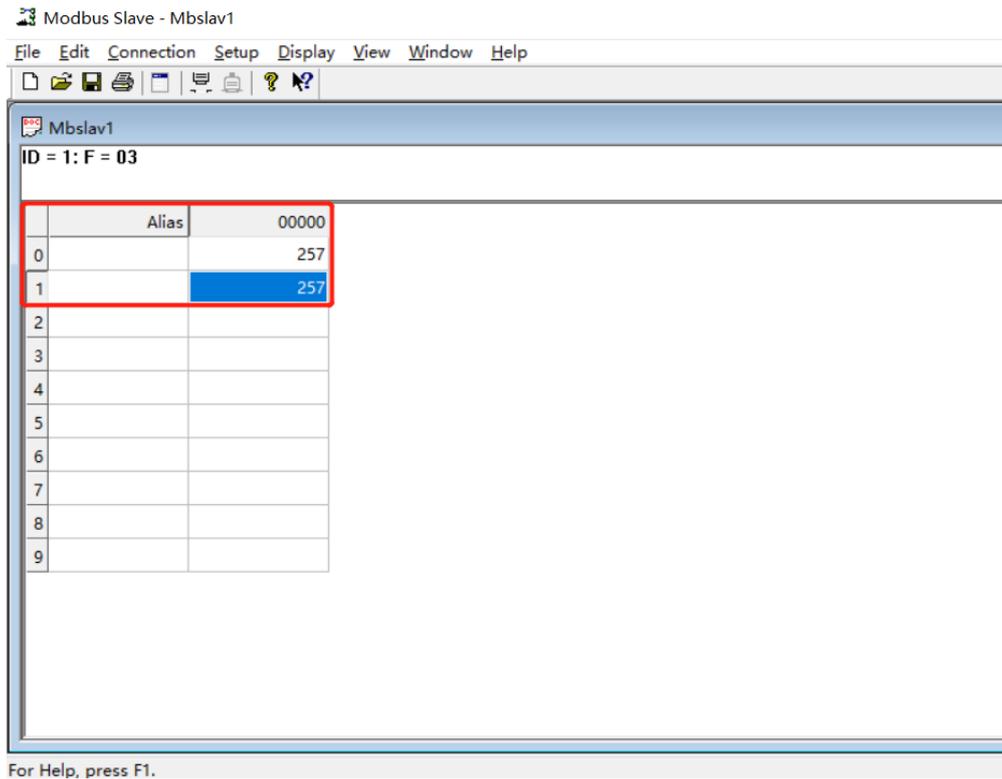


参数全部配置完成后，需重新下载程序至 PLC 中，PLC 与模块需要重新上电。

- b. 在下行数据中，写入 2 个保持寄存器的值，如下图所示。



c. 通过调试软件可以看到接收的 2 个寄存器的值，如下图所示。



示例 2：通过 Modbus Slave 调试软件等工具或设备验证模块 RTU 主站在 Level 模式下读取 10 个保持寄存器。

a. 对配置参数进行配置，通讯模式选择 1 即 RTU 主站模式，如下图所示。

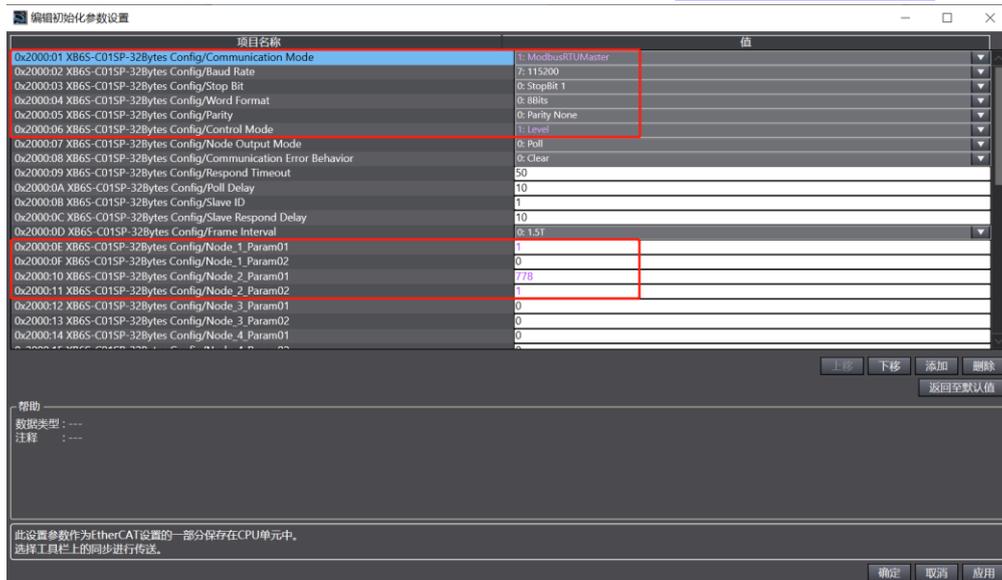
Communication Mode: 选择 ModbusRTUMaster;

Control Mode: 选择 Level;

Node_1_Param_01: 配置 1 (0x00000001) , 配置参数详见 [6.2.8 控制与状态节点代码](#);

Node_2_Param_01: 配置 778 (0x000030A) , 配置参数详见 [6.2.1 Modbus 主站功能](#);

Node_2_Param_02: 配置 1 (0x00000001) , 配置参数详见 [6.2.1 Modbus 主站功能](#)。



参数全部配置完成后，需重新下载程序至 PLC 中，PLC 与模块需要重新上电。

- b. 在下行数据中将控制字置为 1，并打开调试软件发送数据，如下图所示。

位置	端口	说明	R/W	数据类型	变量
插槽0	XB6S-C01SP-32Bytes				
	Outputs_TX 1_7001_01		W	USINT	1
	Outputs_TX 2_7001_02		W	USINT	0
	Outputs_TX 3_7001_03		W	USINT	0
	Outputs_TX 4_7001_04		W	USINT	0
	Outputs_TX 5_7001_05		W	USINT	0
	Outputs_TX 6_7001_06		W	USINT	0
	Outputs_TX 7_7001_07		W	USINT	0
	Outputs_TX 8_7001_08		W	USINT	0
	Outputs_TX 9_7001_09		W	USINT	0
	Outputs_TX 10_7001_0A		W	USINT	0
	Outputs_TX 11_7001_0B		W	USINT	0
	Outputs_TX 12_7001_0C		W	USINT	0
	Outputs_TX 13_7001_0D		W	USINT	0
	Outputs_TX 14_7001_0E		W	USINT	0
	Outputs_TX 15_7001_0F		W	USINT	0
	Outputs_TX 16_7001_10		W	USINT	0
	Outputs_TX 17_7001_11		W	USINT	0
	Outputs_TX 18_7001_12		W	USINT	0
	Outputs_TX 19_7001_13		W	USINT	0
	Outputs_TX 20_7001_14		W	USINT	0
	Outputs_TX 21_7001_15		W	USINT	0
	Outputs_TX 22_7001_16		W	USINT	0
	Outputs_TX 23_7001_17		W	USINT	0
	Outputs_TX 24_7001_18		W	USINT	0
	Outputs_TX 25_7001_19		W	USINT	0
	Outputs_TX 26_7001_1A		W	USINT	0
	Outputs_TX 27_7001_1B		W	USINT	0
	Outputs_TX 28_7001_1C		W	USINT	0
	Outputs_TX 29_7001_1D		W	USINT	0
	Outputs_TX 30_7001_1E		W	USINT	0
	Outputs_TX 31_7001_1F		W	USINT	0
	Outputs_TX 32_7001_20		W	USINT	0

- c. 在上行数据中可以看到接收到的数据，如下图所示。

位置	端口	说明	R/W	数据类型	变量
	Inputs_RX 1_6001_01		R	USINT	5
	Inputs_RX 2_6001_02		R	USINT	0
	Inputs_RX 3_6001_03		R	USINT	0
	Inputs_RX 4_6001_04		R	USINT	255
	Inputs_RX 5_6001_05		R	USINT	0
	Inputs_RX 6_6001_06		R	USINT	255
	Inputs_RX 7_6001_07		R	USINT	0
	Inputs_RX 8_6001_08		R	USINT	255
	Inputs_RX 9_6001_09		R	USINT	0
	Inputs_RX 10_6001_0A		R	USINT	255
	Inputs_RX 11_6001_0B		R	USINT	0
	Inputs_RX 12_6001_0C		R	USINT	255
	Inputs_RX 13_6001_0D		R	USINT	0
	Inputs_RX 14_6001_0E		R	USINT	255
	Inputs_RX 15_6001_0F		R	USINT	0
	Inputs_RX 16_6001_10		R	USINT	255
	Inputs_RX 17_6001_11		R	USINT	0
	Inputs_RX 18_6001_12		R	USINT	255
	Inputs_RX 19_6001_13		R	USINT	0
	Inputs_RX 20_6001_14		R	USINT	255
	Inputs_RX 21_6001_15		R	USINT	0
	Inputs_RX 22_6001_16		R	USINT	255
	Inputs_RX 23_6001_17		R	USINT	0
	Inputs_RX 24_6001_18		R	USINT	0
	Inputs_RX 25_6001_19		R	USINT	0
	Inputs_RX 26_6001_1A		R	USINT	0
	Inputs_RX 27_6001_1B		R	USINT	0
	Inputs_RX 28_6001_1C		R	USINT	0
	Inputs_RX 29_6001_1D		R	USINT	0
	Inputs_RX 30_6001_1E		R	USINT	0
	Inputs_RX 31_6001_1F		R	USINT	0
	Inputs_RX 32_6001_20		R	USINT	0

10、 Freeport_Input 功能示例

示例：通过串口调试助手等工具或设备利用 Freeport_Input 在 Level 模式下接收一个 8 字节数据。

据。

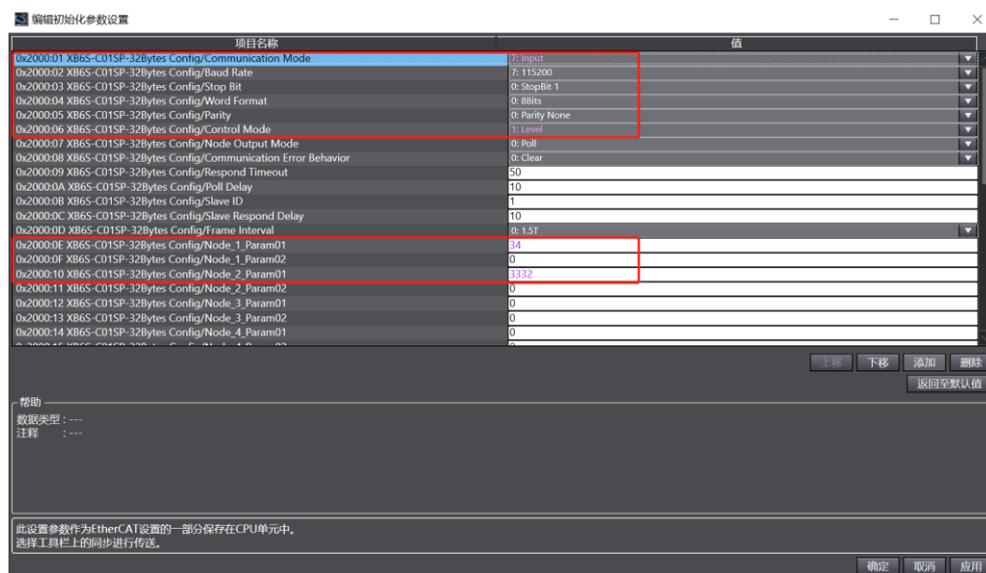
- a. 对配置参数进行配置，通讯模式选择 7 即 Input 模式，如下图所示。

Communication Mode: 选择 Input;

Control Mode: 选择 Level;

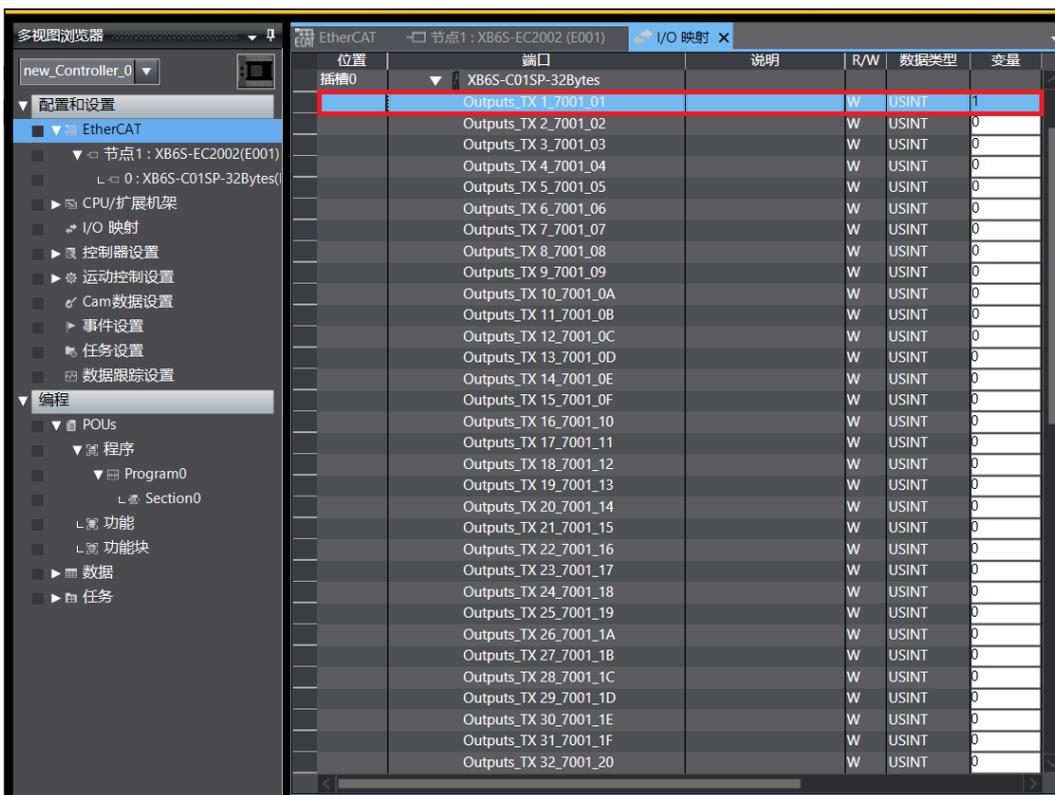
Node_1_Param_01: 配置 34 (0x00000022) , 配置参数详见 [6.2.8 控制与状态节点代码](#);

Node_2_Param_01: 配置 3332 (0x00000D04) , 配置参数详见 [6.2.6 Freeport 自由口功能](#)。

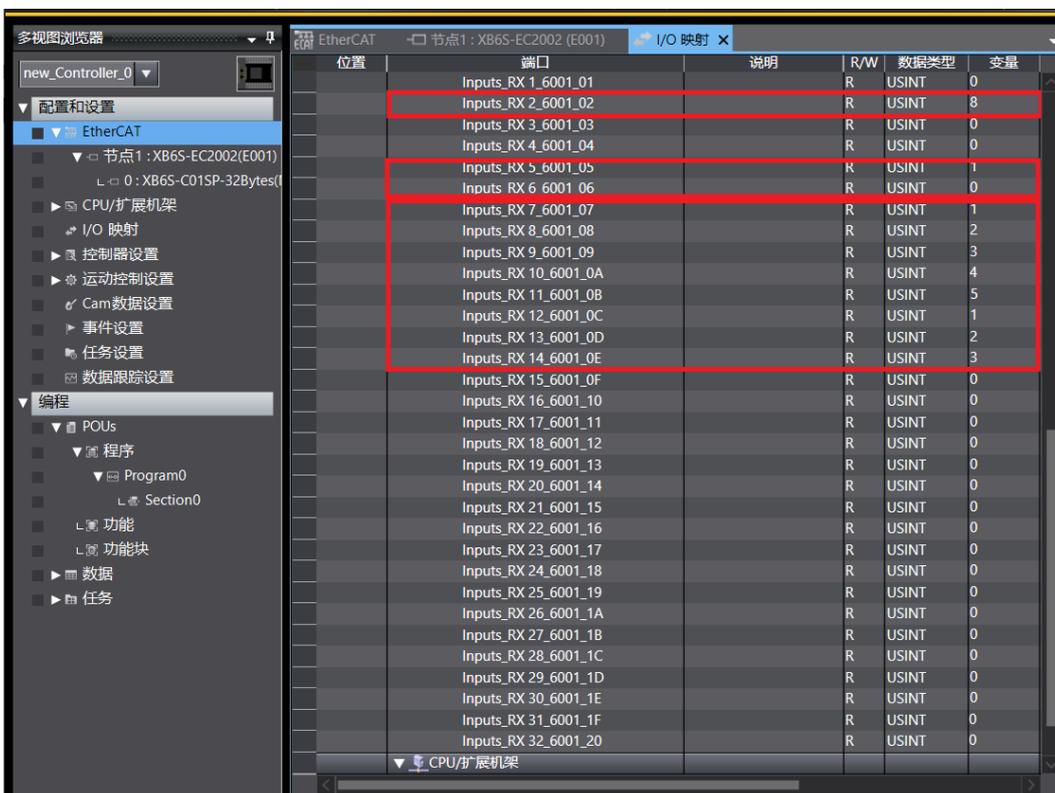


参数全部配置完成后，需重新下载程序至 PLC 中，PLC 与模块需要重新上电。

- b. 在下行数据中将控制字置为 1，并打开串口调试助手发送数据，如下图所示。



- c. 上行数据可以看到接收的数据长度为 8，接收的数据计数为 1，接收的 8 个字节的数据 (RX7~RX14)，如下图所示。



6.4.3 在 TIA Portal V17 软件环境下的应用

1、准备工作

● 硬件环境

- 模块型号 XB6S-C01SP
- PROFINET 总线耦合器模块，端盖
本说明以 XB6S-PN2002 耦合器模块为例
- 计算机一台，预装 TIA Portal V17 软件
- PROFINET 专用屏蔽电缆
- 西门子 PLC 一台，本说明以西门子 S7-1500 CPU 1511-1 PN 为例
- 开关电源一台
- 模块安装导轨及导轨固定件
- 设备配置文件

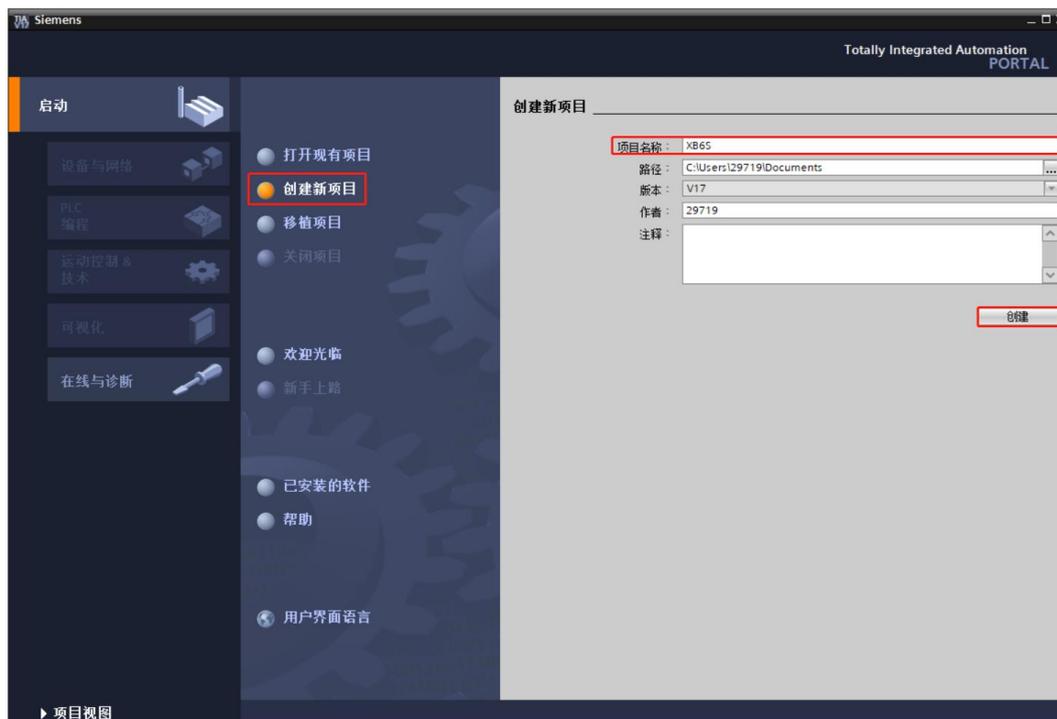
配置文件获取地址：<https://www.solidotech.com/documents/configfile>

● 硬件组态及接线

请按照“[4 安装和拆卸](#)”“[5 接线](#)”要求操作

2、新建工程

- a. 打开 TIA Portal V17 软件，单击“创建新项目”，各项信息输入完成后单击“创建”按钮，如下图所示。



- ◆ 项目名称：自定义，可保持默认。
- ◆ 路径：项目保持路径，可保持默认。
- ◆ 版本：可保持默认。
- ◆ 作者：可保持默认。
- ◆ 注释：自定义，可不填写。

3、添加 PLC 控制器

- a. 单击“组态设备”，如下图所示。

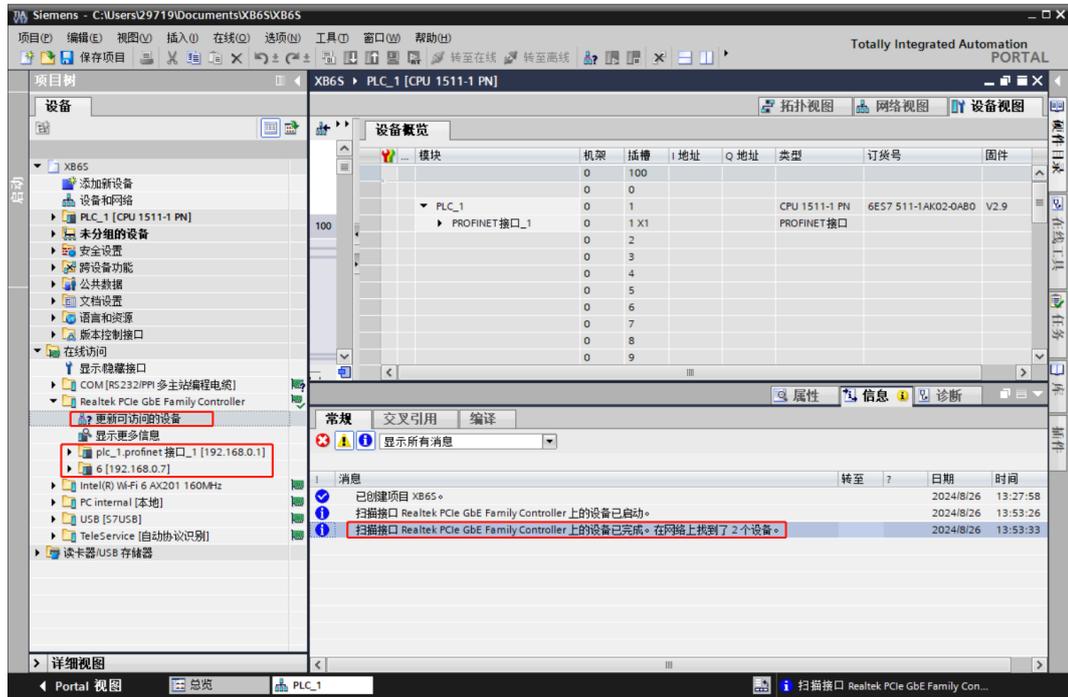


- b. 单击“添加新设备”，选择当前所使用的 PLC 型号，单击“添加”，如下图所示。添加完成后可查看到 PLC 已经添加至设备导航树中。



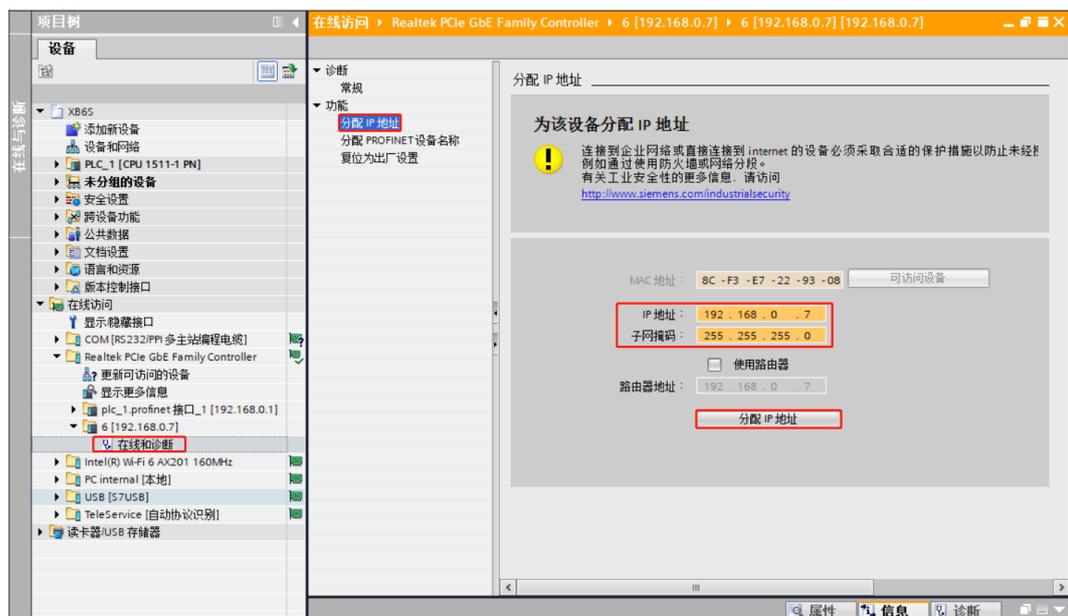
4、扫描连接设备

- a. 单击左侧导航树“在线访问 -> 更新可访问的设备”，如下图所示。更新完毕，显示连接的从站设备，如下图所示。

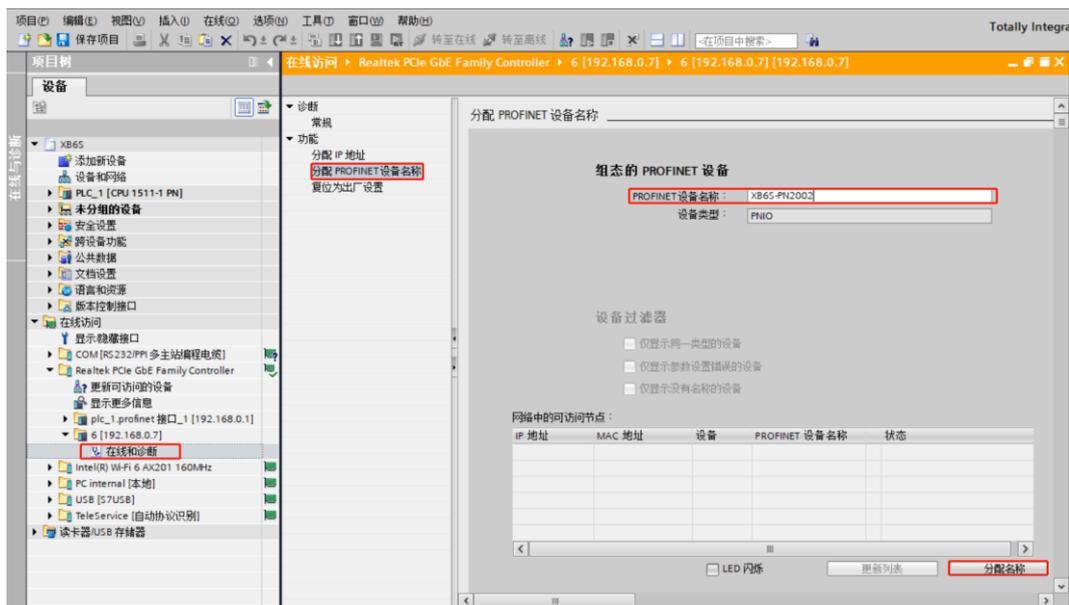


电脑的 IP 地址必须和 PLC 在同一网段，若不在同一网段，修改电脑 IP 地址后，重复上述步骤。

- b. 双击左侧导航树从站设备下的“在线和诊断”，在“功能”菜单下可以分配当前从站的 IP 地址及设备名称。单击“分配 IP 地址”，先填写“子网掩码”，再填写“IP 地址”，单击最下方的“分配 IP 地址”，如下图所示。

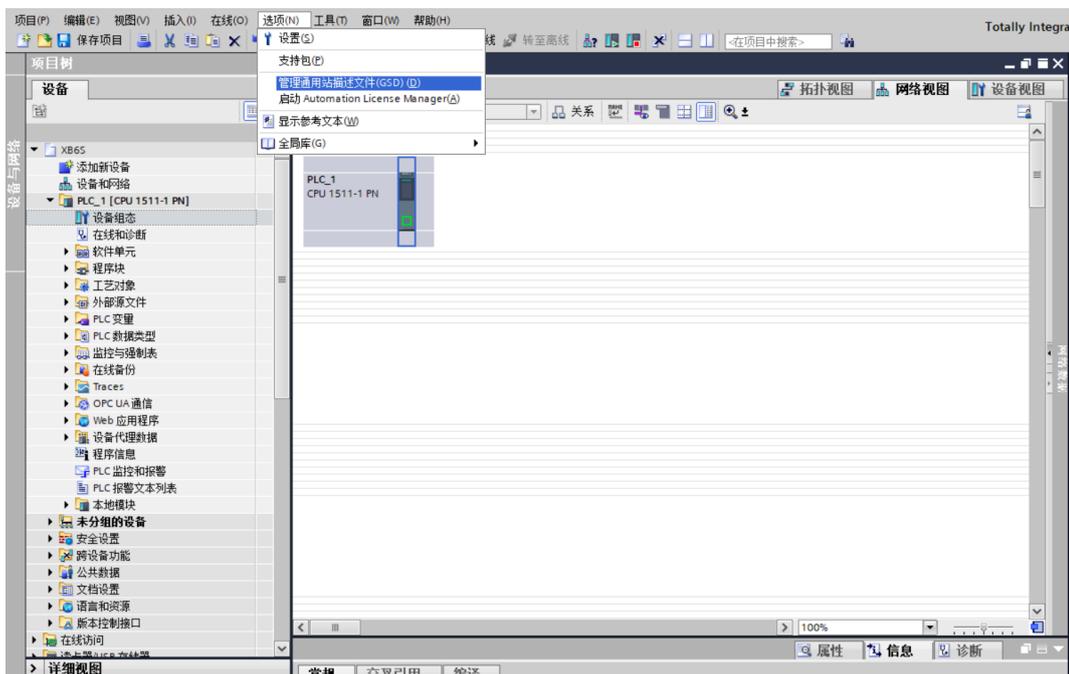


- c. 单击“分配 PROFINET 设备名称”，填写“PROFINET 设备名称”，单击“分配名称”，如下图所示。

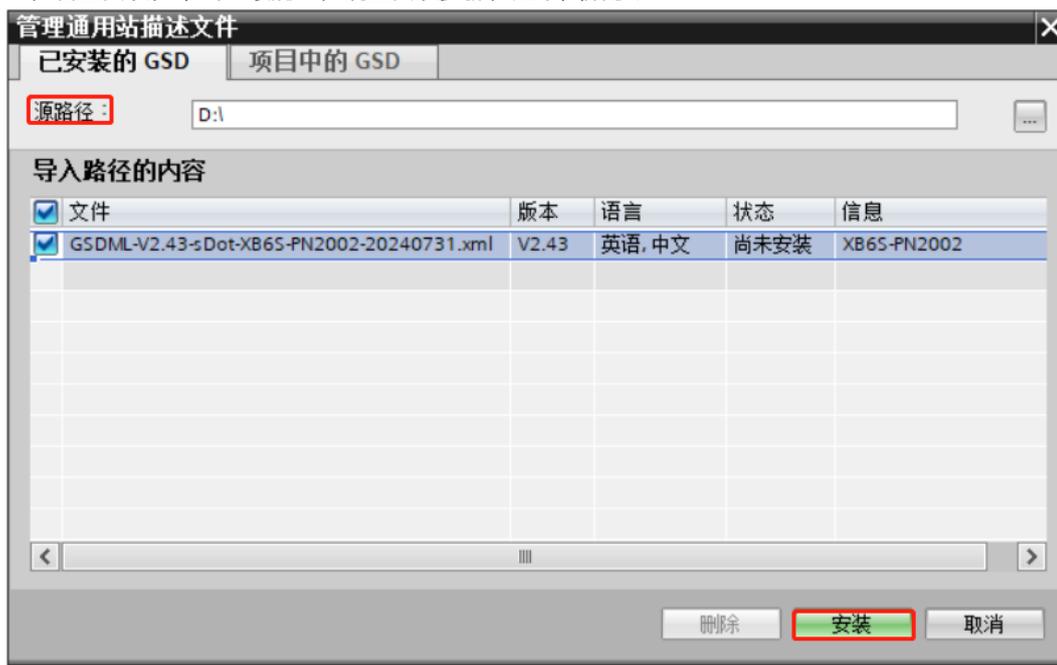


5、添加 GSD 配置文件

- a. 菜单栏中，选择“选项 -> 管理通用站描述文件(GSDML)(D)”，如下图所示。

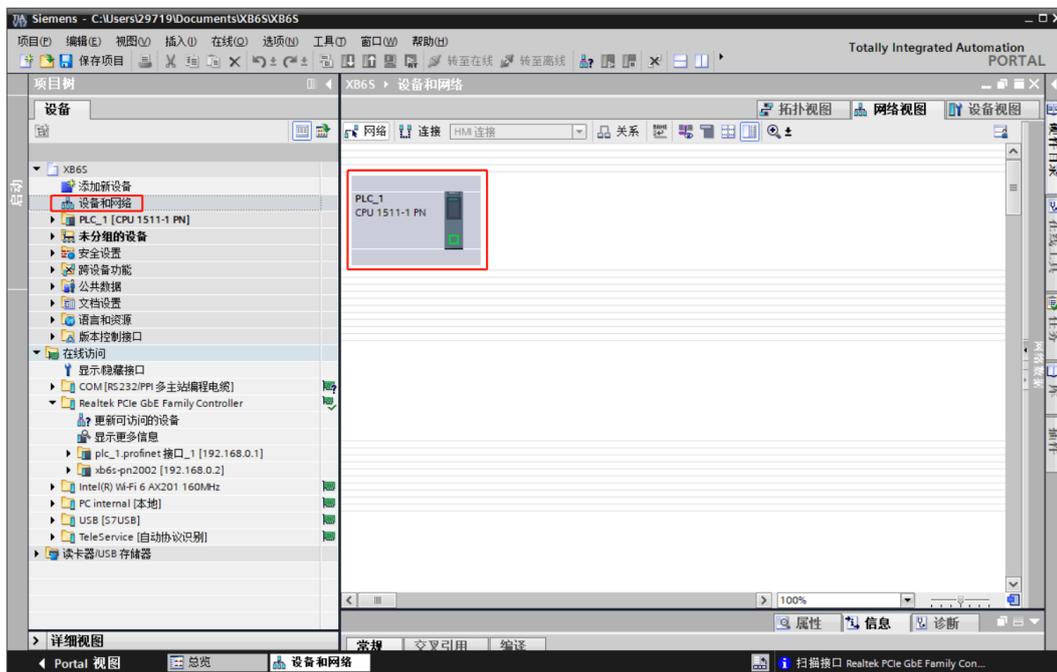


- b. 单击“源路径”选择文件夹，查看要添加的 GSD 文件的状态是否为“尚未安装”，未安装单击“安装”按钮，若已安装，单击“取消”，跳过安装步骤，如下图所示。

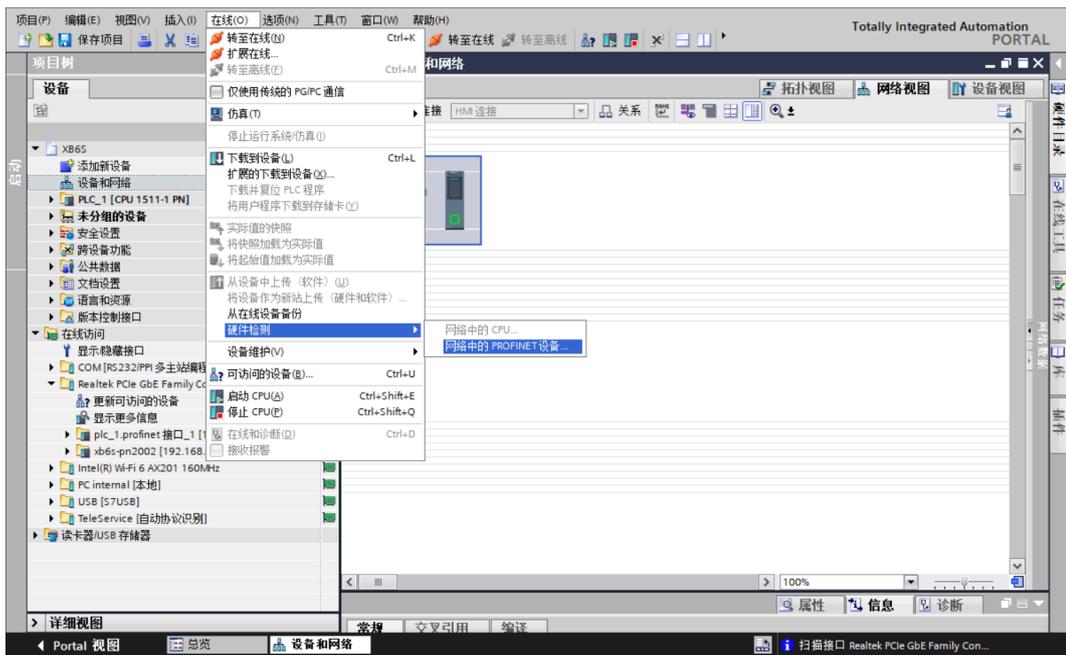


6、硬件检测添加设备

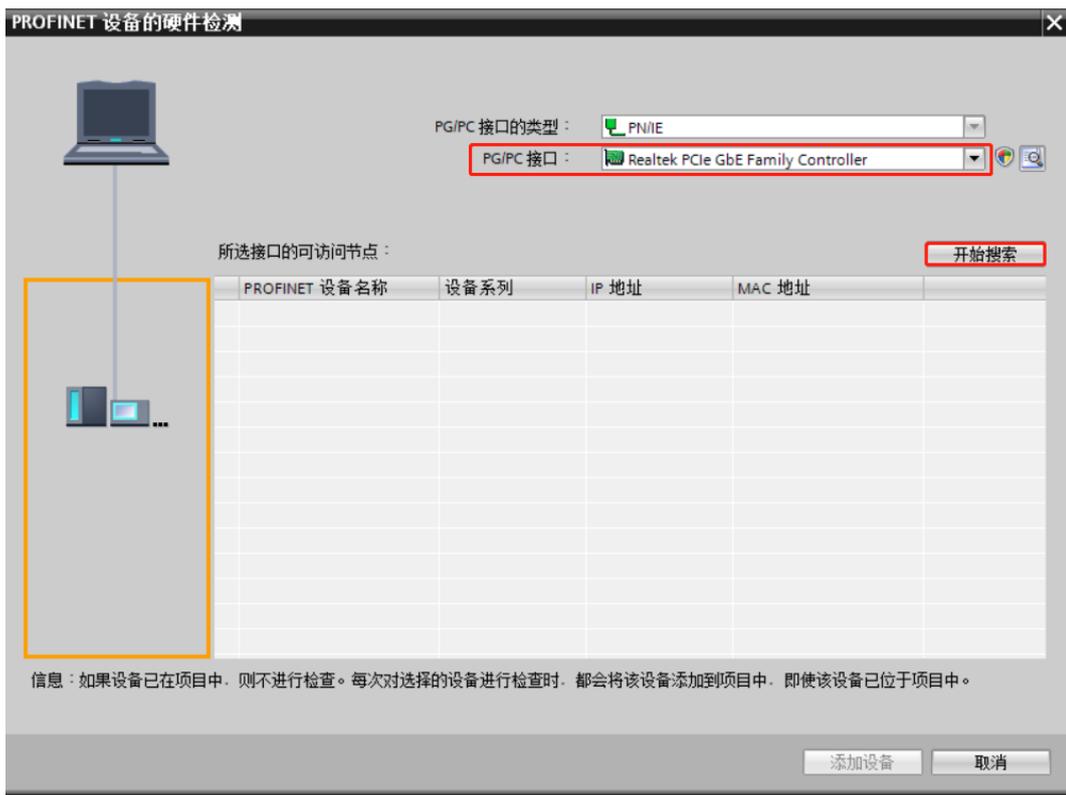
- a. 双击左侧导航树“设备和网络”，在网络视图中选中 PLC，如下图所示。



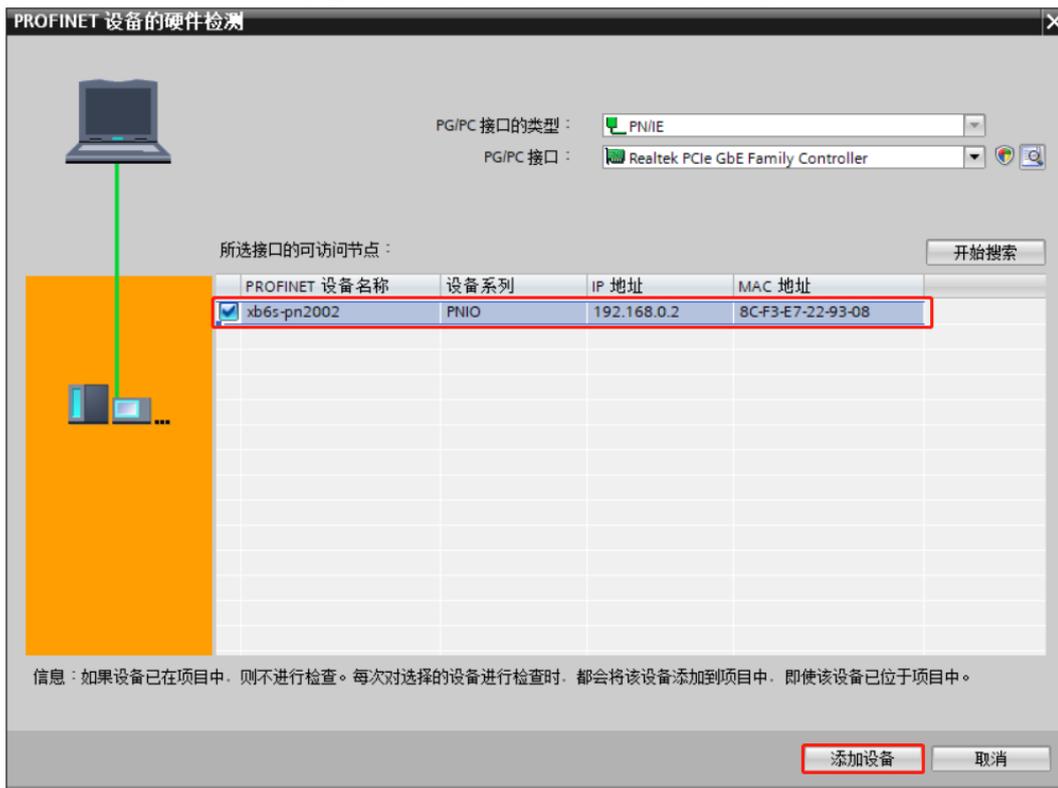
- b. 菜单栏中，选择“在线 -> 硬件检测 -> 网络中的 PROFINET 设备”，如下图所示。



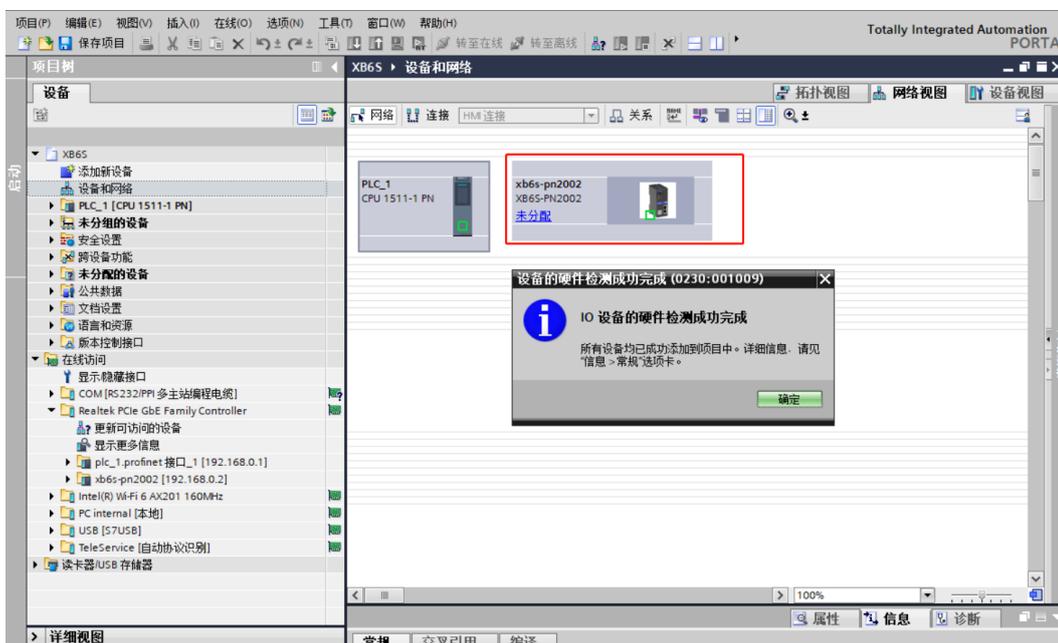
- c. 弹出硬件检测窗口，选择正确的 PG/PC 接口，单击“开始搜索”，如下图所示。



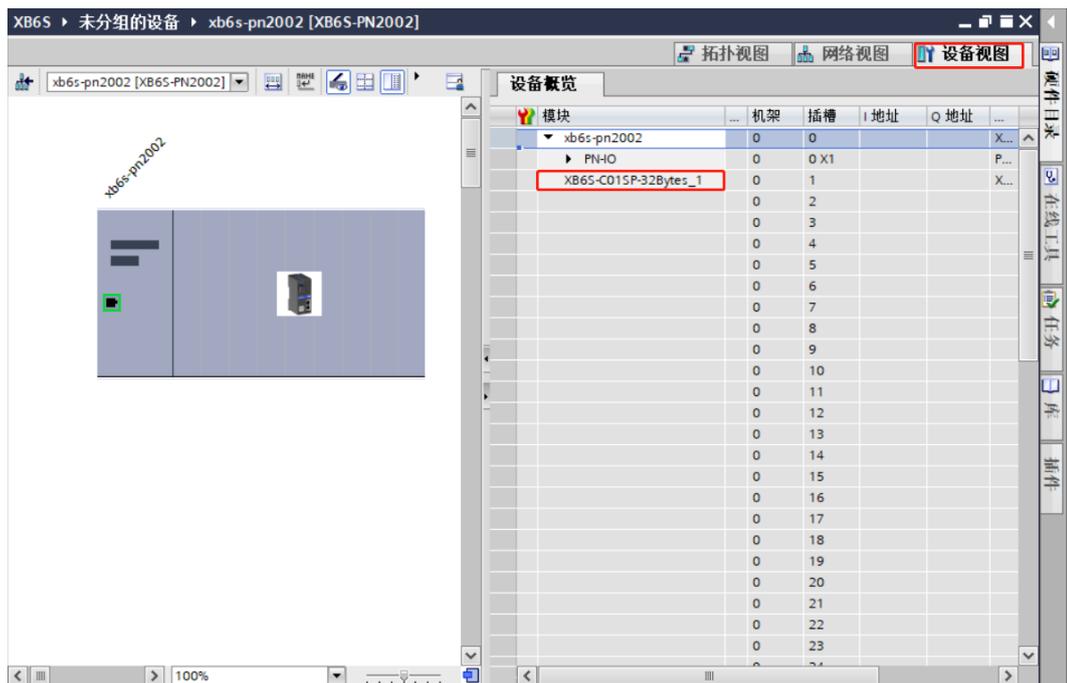
- d. 检测网络中的耦合器设备，勾选耦合器，单击“添加设备”，如下图所示。



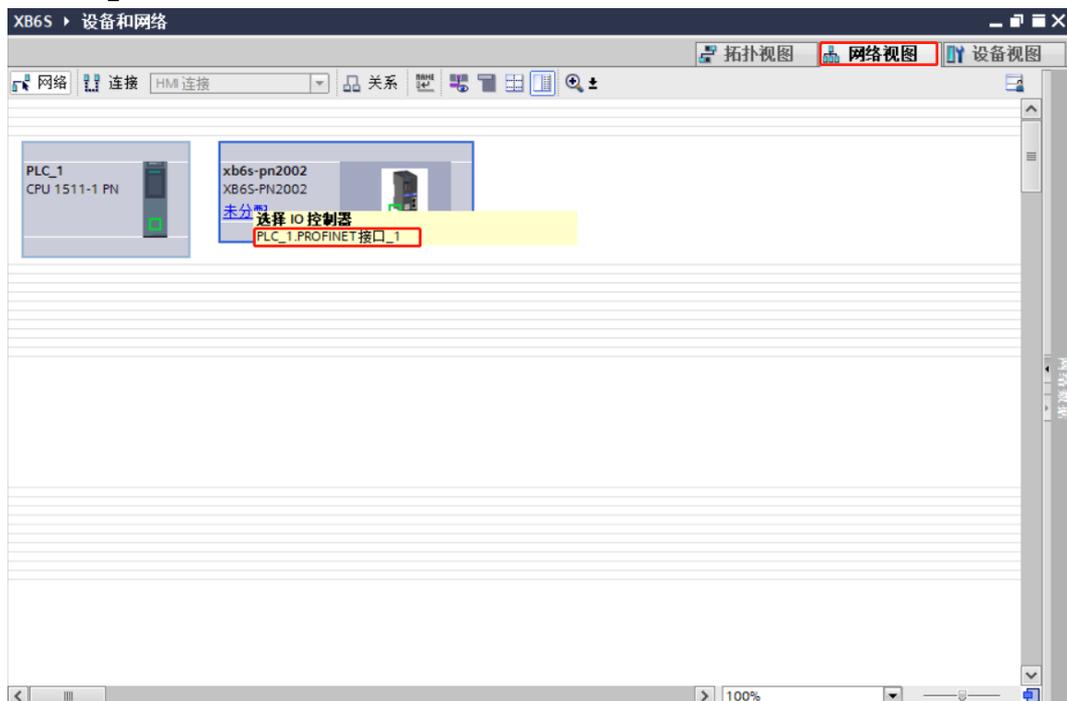
- e. 耦合器添加成功后，可以在网络视图中看到耦合器图标，如下图所示。



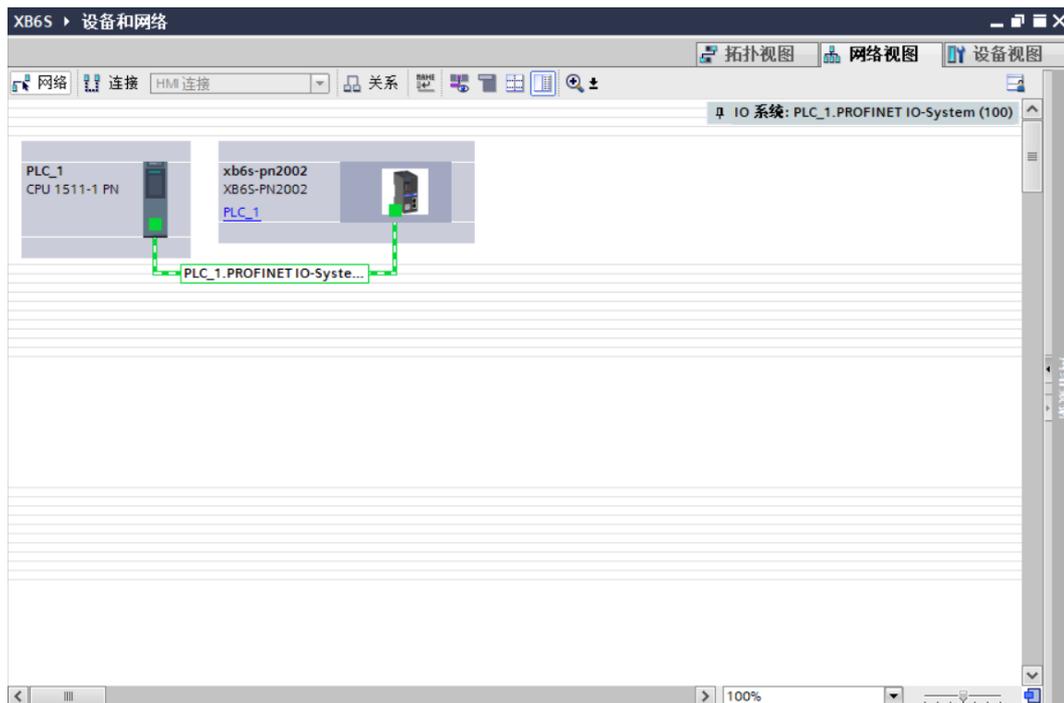
- f. 选中耦合器图标，切换到设备视图，可以看到拓扑中的 IO 设备均检测添加完成，如下图所示。



- g. 切换到网络视图，单击耦合器即从站设备上的“未分配（蓝色字体）”，选择“PLC_1.PROFINET interface_1”，如下图所示。

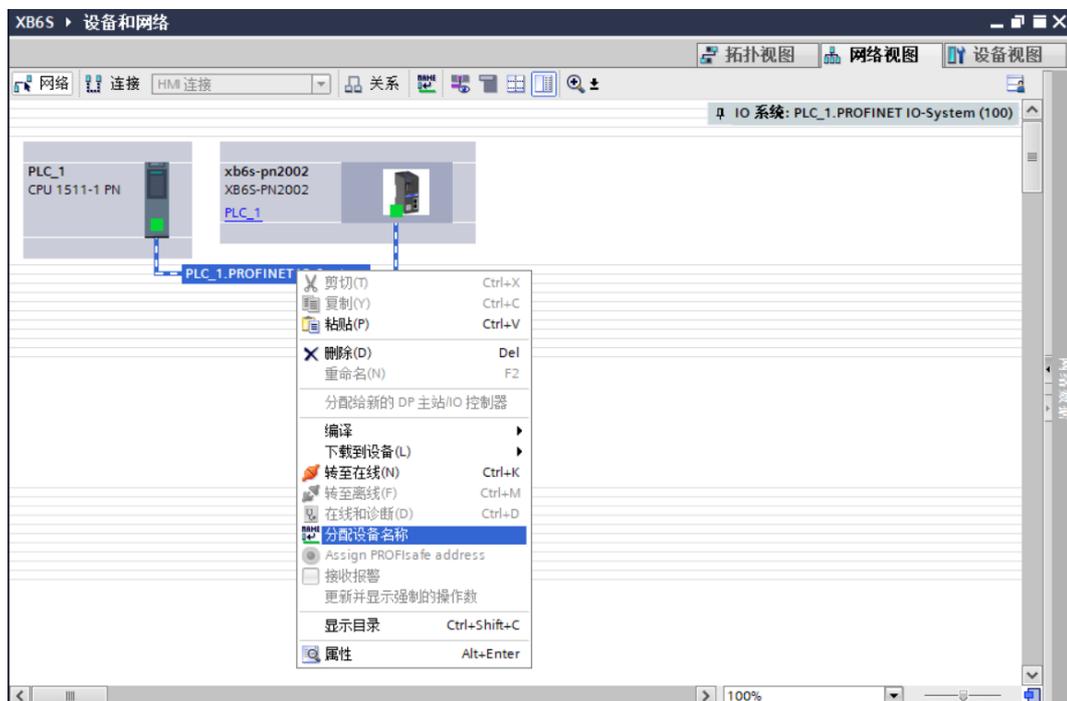


- h. 连接完成后，如下图所示。

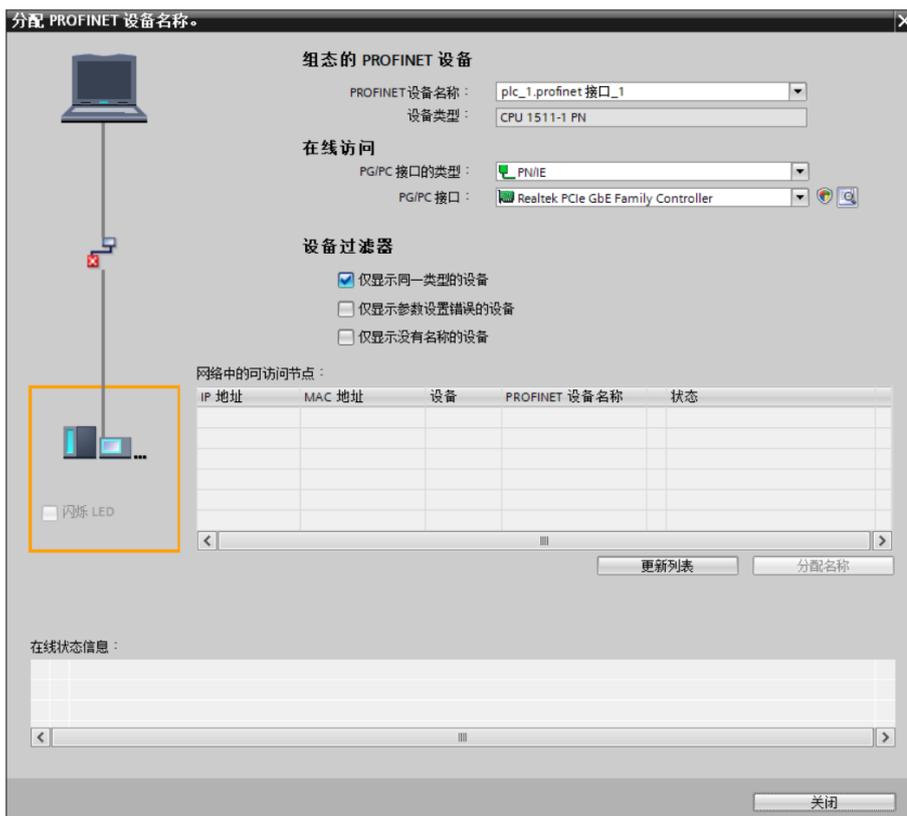


7、分配设备名称

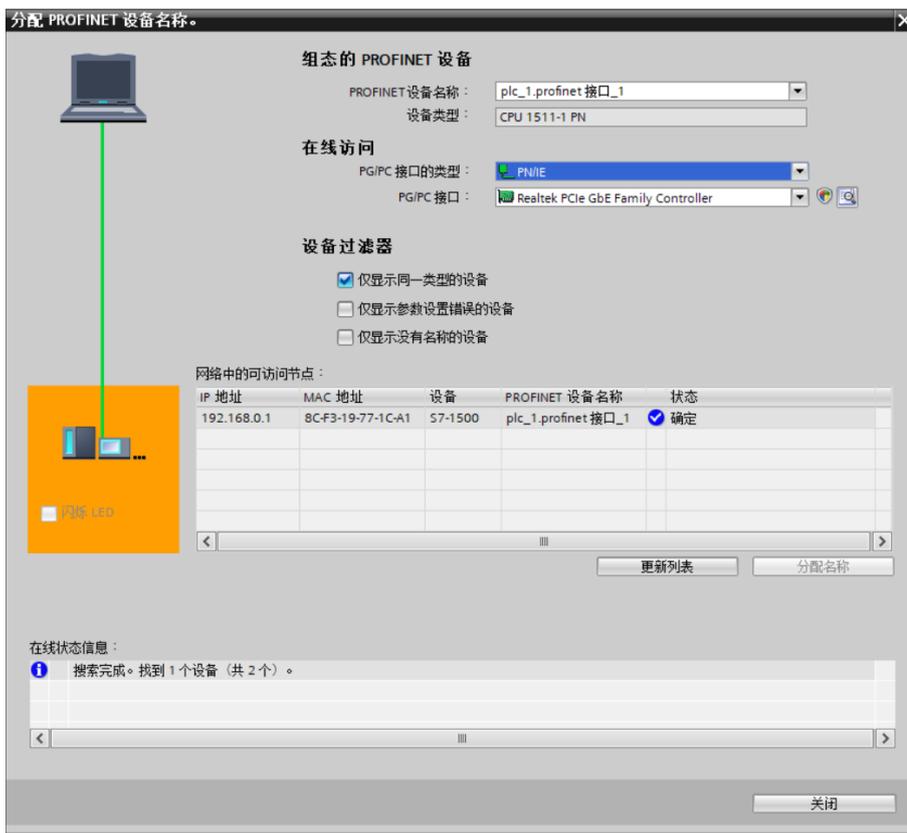
- a. 在网络视图中，右击 PLC 和耦合器的连接线，选择“分配设备名称”，如下图所示。



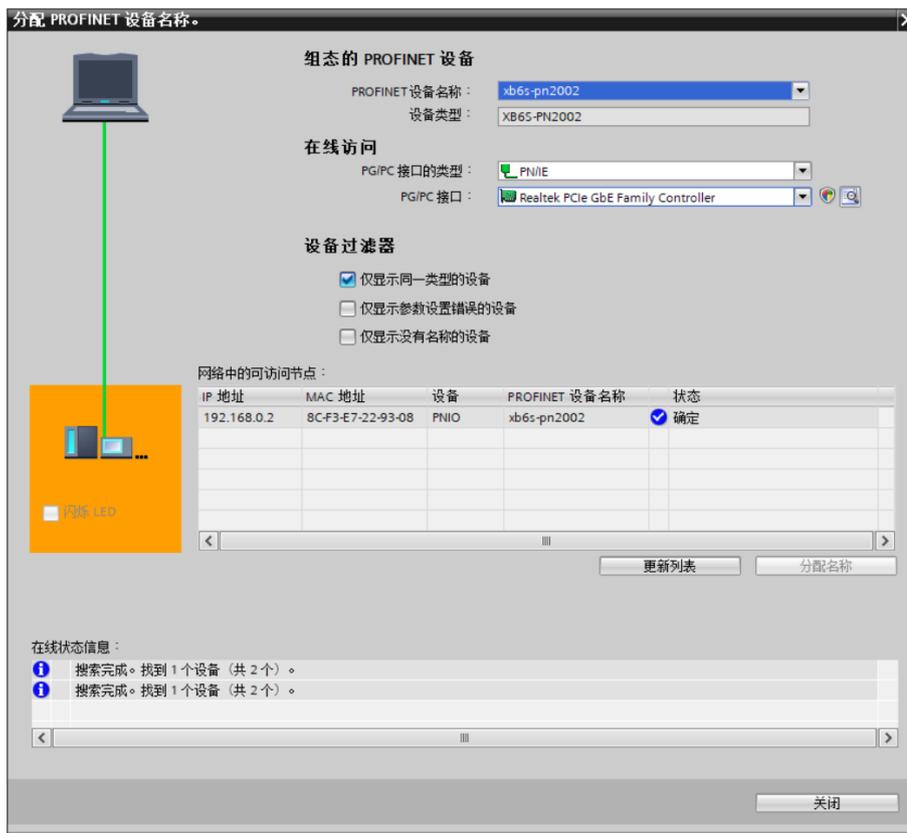
- b. 弹出“分配 PROFINET 设备名称”窗口，如下图所示。



- c. 设备名称选择 PLC，单击“更新列表”。更新完成后，查看“网络中的可访问节点”中，节点的状态是否为“确定”。若不为确定，选中设备，单击“分配名称”，如下图所示。



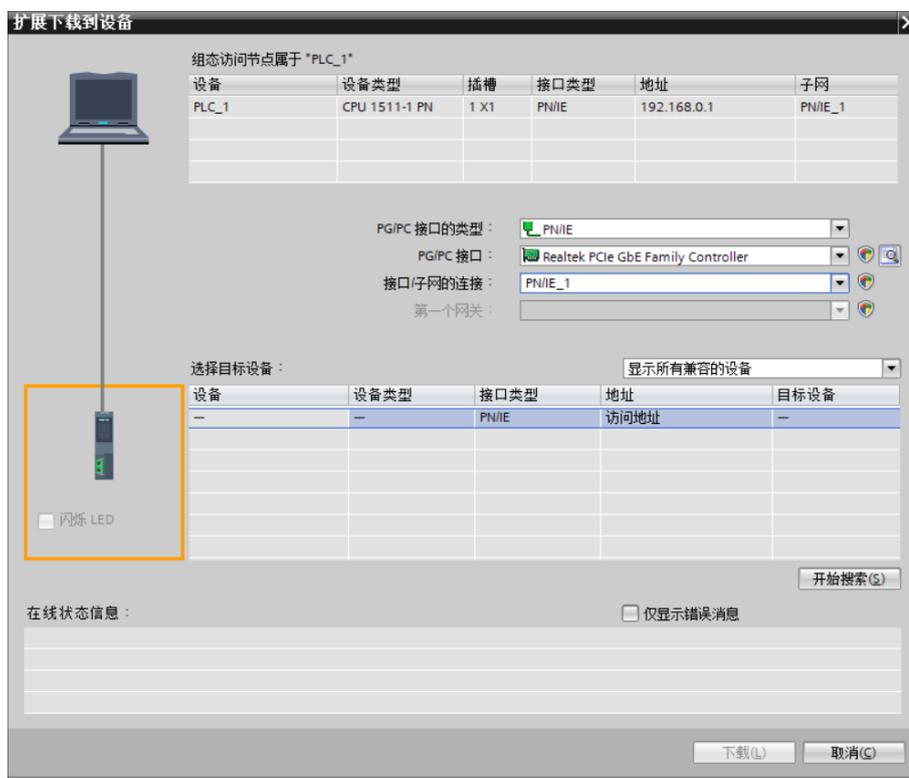
- d. 设备名称选择耦合器，单击“更新列表”，更新后用同样的方法分配名称，如下图所示。



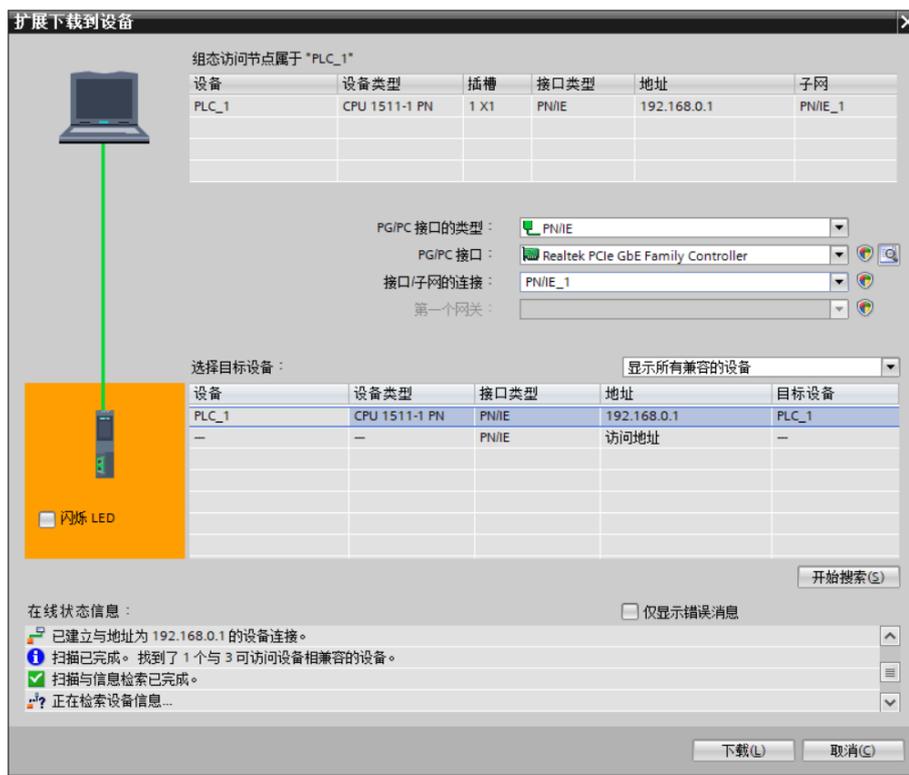
- e. 查看模块丝印上的 MAC 地址是否与所分配设备名称的 MAC 地址相同。单击“关闭”。

8、下载组态结构

- 在网络视图中，选中 PLC。先单击菜单栏中的编译按钮，再单击下载按钮，将当前组态下载到 PLC 中。
- 在弹出的“扩展下载到设备”界面，配置如下图所示。



- 单击“开始搜索”按钮，如下图所示。



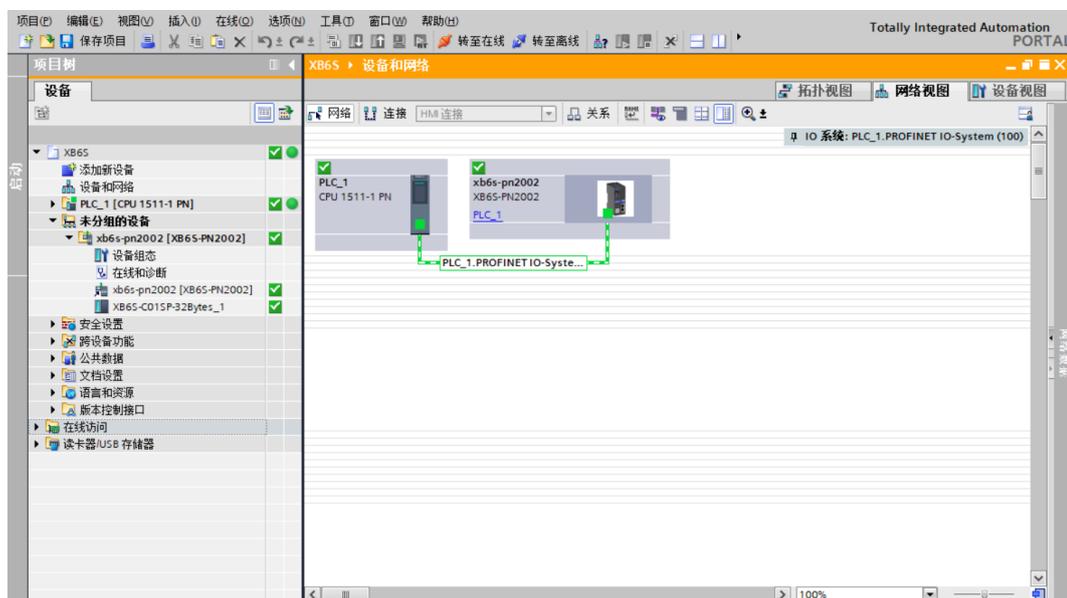
- d. 单击“下载”，弹出下载预览窗口，如下图所示。



- e. 单击“装载”。
- f. 单击“完成”。
- g. 将设备重新上电。

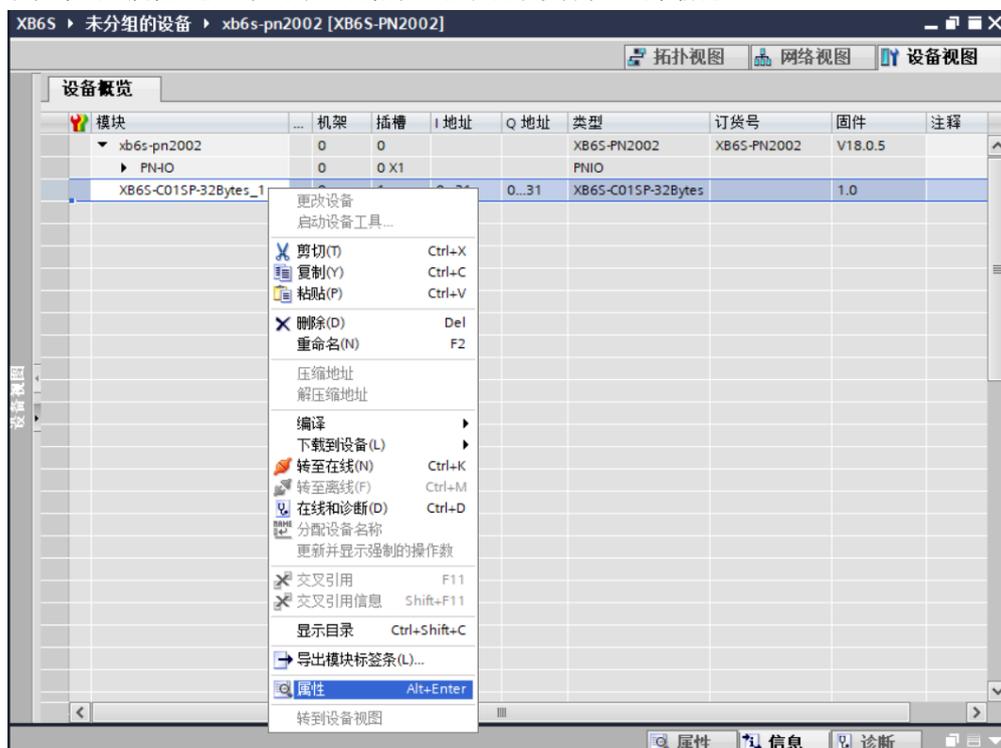
9、通讯连接

- a. 单击菜单栏中的“启动 CPU”按钮，再单击“转至在线”按钮，图标均为绿色即连接成功，如下图所示。

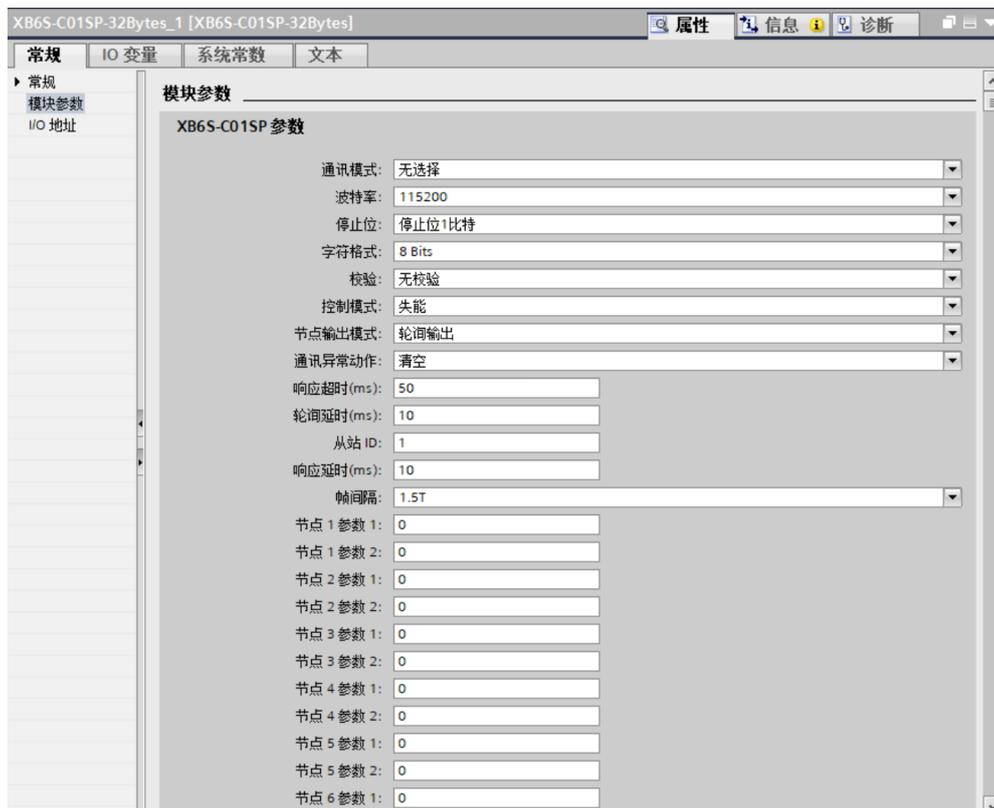


10、 参数设置

- a. 在离线状态下，打开“网络视图”，选中耦合器模块，切换到设备视图，右击 XB6S-C01SP-32Bytes 模块，单击“属性”按钮，可以查看和设置模块各项参数，如下图所示。

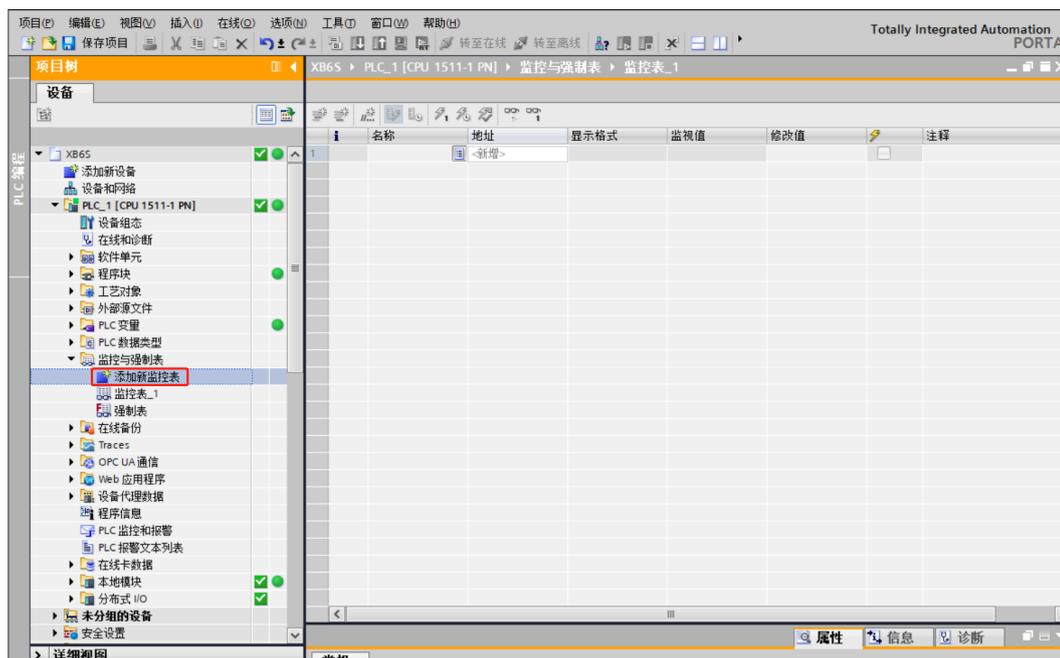


- b. 在属性页面，单击“模块参数”，如下图所示。参数可以根据实际使用需要进行配置，配置完成后，重新下载程序至 PLC 中，PLC 与模块需要重新上电。



11、 功能验证

- a. 展开左侧的项目导航，选择“监控与强制表”，双击“添加新监控表”，系统新增监控表，如下图所示。



- b. 打开“设备视图”，查看设备概览中各个模块的通道 I 地址（输入信号的通道地址）和 Q 地址（输出信号的通道地址）。

例如查看到 XB6S-C01SP-32Bytes 模块的“I 地址”为 0 至 31，“Q 地址”为 0 至 31，如下图所示。

模块	机架	插槽	I 地址	Q 地址	类型	订货号	固件	注释
xb6s-pn2002	0	0			XB6S-PN2002	XB6S-PN2002	V18.0.5	
PN-IO	0	0 X1			PNIO			
XB6S-C01SP-32Bytes_1	0	1	0...31	0...31	XB6S-C01SP-32Bytes		1.0	
	0	2						
	0	3						
	0	4						
	0	5						
	0	6						
	0	7						
	0	8						

- c. 在监控表的地址单元格填写输入输出通道地址，如写入“IB0”到“IB31”，“QB0”到“QB31”，按“回车键”，全部填写完毕后，单击  按钮，对数据进行监控。

12、 RTU 主站模式功能示例

示例 1：通过 Modbus Slave 调试软件等工具或设备验证模块 RTU 主站在 disable 模式下写 2 个保持寄存器的值。

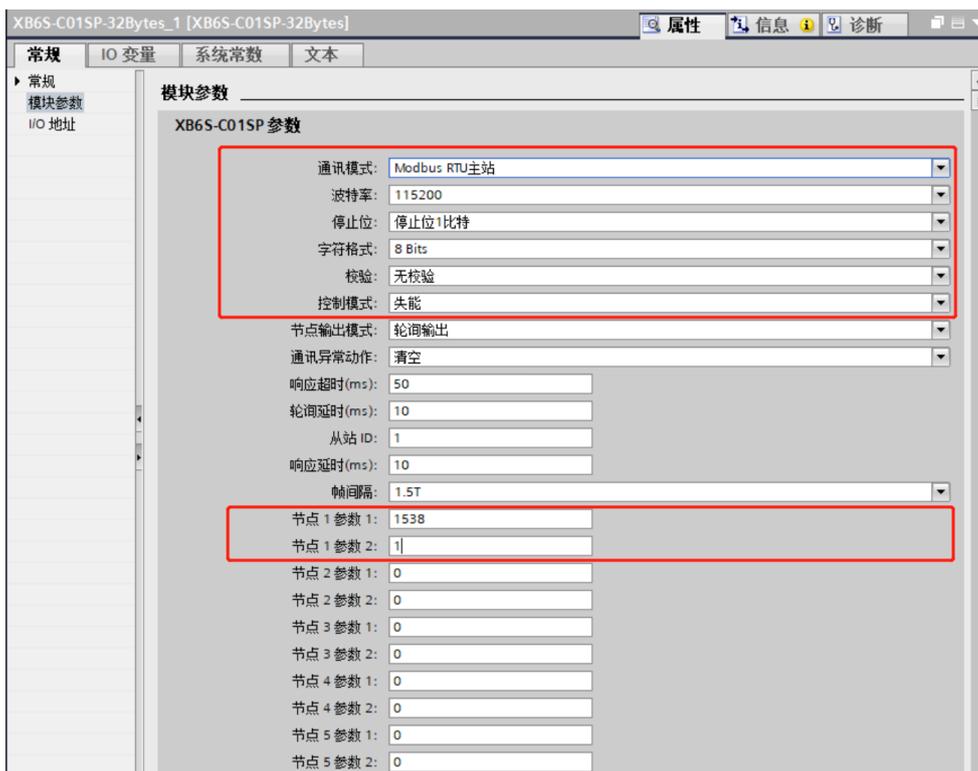
- a. 对配置参数进行配置，通讯模式选择 Modbus RTU 主站模式，如下图所示。

通讯模式：选择 Modbus RTU 主站；

控制模式：选择失能；

节点 1 参数 1：配置 1538 (0x00000602)，配置参数详见 [6.2.1 Modbus 主站功能](#)；

节点 1 参数 2：配置 1 (0x00000001)，配置参数详见 [6.2.1 Modbus 主站功能](#)。



参数全部配置完成后，需重新下载程序至 PLC 中，PLC 与模块需要重新上电。

- b. 在下行数据中，写入 2 个保持寄存器的值，如下图所示。

名称	地址	显示格式	监视值	修改值	注释
	%QB0	无符号十进制	1	1	⚠
	%QB1	无符号十进制	1	1	⚠
	%QB2	无符号十进制	1	1	⚠
	%QB3	无符号十进制	1	1	⚠
	%QB4	无符号十进制	0		
	%QB5	无符号十进制	0		
	%QB6	无符号十进制	0		
	%QB7	无符号十进制	0		
	%QB8	无符号十进制	0		
	%QB9	无符号十进制	0		
	%QB10	无符号十进制	0		
	%QB11	无符号十进制	0		
	%QB12	无符号十进制	0		
	%QB13	无符号十进制	0		
	%QB14	无符号十进制	0		
	%QB15	无符号十进制	0		
	%QB16	无符号十进制	0		
	%QB17	无符号十进制	0		
	%QB18	无符号十进制	0		
	%QB19	无符号十进制	0		
	%QB20	无符号十进制	0		
	%QB21	无符号十进制	0		
	%QB22	无符号十进制	0		
	%QB23	无符号十进制	0		
	%QB24	无符号十进制	0		
	%QB25	无符号十进制	0		
	%QB26	无符号十进制	0		
	%QB27	无符号十进制	0		

- c. 通过调试软件可以看到接收的 2 个寄存器的值，如下图所示。

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

For Help, press F1.

示例 2：通过 Modbus Slave 调试软件等工具或设备验证模块 RTU 主站在 Level 模式下读取 10 个保持寄存器。

- a. 对配置参数进行配置，通讯模式选择 Modbus RTU 主站模式，如下图所示。

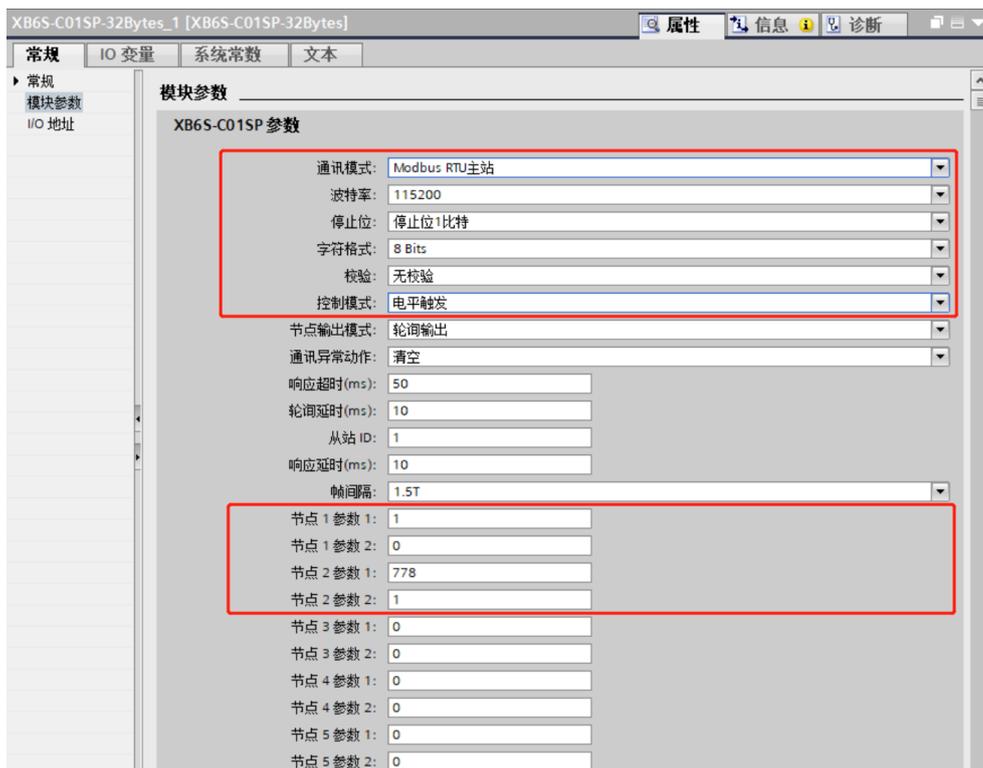
通讯模式：选择 Modbus RTU 主站；

控制模式：选择电平触发；

节点 1 参数 1：配置 1 (0x00000001)，配置参数详见 [6.2.8 控制与状态节点代码](#)；

节点 2 参数 1：配置 778 (0x0000030A)，配置参数详见 [6.2.1 Modbus 主站功能](#)；

节点 2 参数 2：配置 1 (0x00000001)，配置参数详见 [6.2.1 Modbus 主站功能](#)。



参数全部配置完成后，需重新下载程序至 PLC 中，PLC 与模块需要重新上电。

- b. 在下行数据中将控制字置为 1，并打开调试软件发送数据，如下图所示。

名称	地址	显示格式	监视值	修改值	注释
	%QB0	无符号十进制	1	1	
	%QB1	无符号十进制	0		
	%QB2	无符号十进制	0		
	%QB3	无符号十进制	0		
	%QB4	无符号十进制	0		
	%QB5	无符号十进制	0		
	%QB6	无符号十进制	0		
	%QB7	无符号十进制	0		
	%QB8	无符号十进制	0		
	%QB9	无符号十进制	0		
	%QB10	无符号十进制	0		
	%QB11	无符号十进制	0		
	%QB12	无符号十进制	0		
	%QB13	无符号十进制	0		
	%QB14	无符号十进制	0		
	%QB15	无符号十进制	0		
	%QB16	无符号十进制	0		
	%QB17	无符号十进制	0		
	%QB18	无符号十进制	0		
	%QB19	无符号十进制	0		
	%QB20	无符号十进制	0		
	%QB21	无符号十进制	0		
	%QB22	无符号十进制	0		
	%QB23	无符号十进制	0		
	%QB24	无符号十进制	0		
	%QB25	无符号十进制	0		
	%QB26	无符号十进制	0		

- c. 在上行数据中可以看到接收到的数据，如下图所示。

名称	地址	显示格式	监视值	修改值	注释
	%IB0	无符号十进制	5		
	%IB1	无符号十进制	0		
	%IB2	无符号十进制	0		
	%IB3	无符号十进制	255		
	%IB4	无符号十进制	0		
	%IB5	无符号十进制	255		
	%IB6	无符号十进制	0		
	%IB7	无符号十进制	255		
	%IB8	无符号十进制	0		
	%IB9	无符号十进制	255		
	%IB10	无符号十进制	0		
	%IB11	无符号十进制	255		
	%IB12	无符号十进制	0		
	%IB13	无符号十进制	255		
	%IB14	无符号十进制	0		
	%IB15	无符号十进制	255		
	%IB16	无符号十进制	0		
	%IB17	无符号十进制	255		
	%IB18	无符号十进制	0		
	%IB19	无符号十进制	255		
	%IB20	无符号十进制	0		
	%IB21	无符号十进制	255		
	%IB22	无符号十进制	0		
	%IB23	无符号十进制	0		
	%IB24	无符号十进制	0		
	%IB25	无符号十进制	0		
	%IB26	无符号十进制	0		
	%IB27	无符号十进制	0		

13、 Freeport_Input 功能示例

示例：通过串口调试助手等工具或设备利用 Freeport_Input 在 Level 模式下接收一个 8 字节数据。

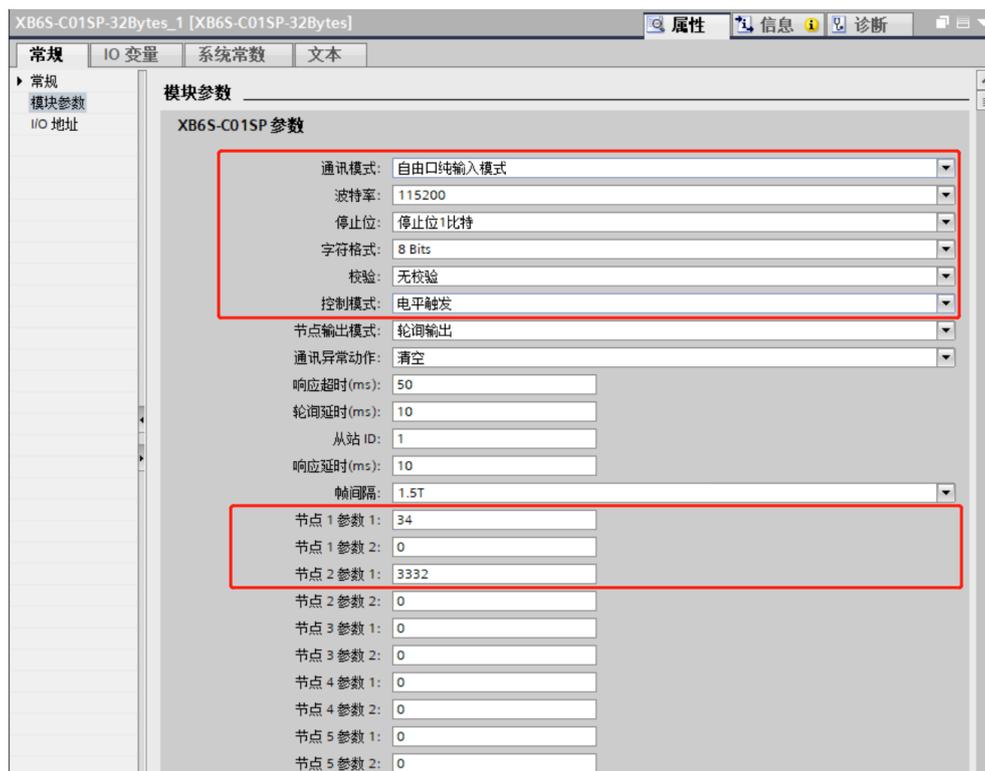
- a. 对配置参数进行配置，通讯模式选择自由口纯输入模式，如下图所示。

通讯模式：选择自由口纯输入模式；

控制模式：选择电平触发；

节点 1 参数 1：配置 34 (0x00000022)，配置参数详见 [6.2.8 控制与状态节点代码](#)；

节点 2 参数 1：配置 3332 (0x0000D04)，配置参数详见 [6.2.6 Freeport 自由口功能](#)。



参数全部配置完成后，需重新下载程序至 PLC 中，PLC 与模块需要重新上电。

- b. 在下行数据中将控制字置为 1，并打开串口调试助手发送数据，如下图所示。

名称	地址	显示格式	监视值	修改值	注释
	%QB0	无符号十进制	1	1	
	%QB1	无符号十进制	0		
	%QB2	无符号十进制	0		
	%QB3	无符号十进制	0		
	%QB4	无符号十进制	0		
	%QB5	无符号十进制	0		
	%QB6	无符号十进制	0		
	%QB7	无符号十进制	0		
	%QB8	无符号十进制	0		
	%QB9	无符号十进制	0		
	%QB10	无符号十进制	0		
	%QB11	无符号十进制	0		
	%QB12	无符号十进制	0		
	%QB13	无符号十进制	0		
	%QB14	无符号十进制	0		
	%QB15	无符号十进制	0		
	%QB16	无符号十进制	0		
	%QB17	无符号十进制	0		
	%QB18	无符号十进制	0		
	%QB19	无符号十进制	0		
	%QB20	无符号十进制	0		
	%QB21	无符号十进制	0		
	%QB22	无符号十进制	0		
	%QB23	无符号十进制	0		
	%QB24	无符号十进制	0		
	%QB25	无符号十进制	0		
	%QB26	无符号十进制	0		

- c. 上行数据可以看到接收的数据长度为 8，接收的数据计数为 1，接收的 8 个字节的数据（IB6~IB13），如下图所示。

名称	地址	显示格式	监视值	修改值	注释
	%IB0	无符号十进制	0		
	%IB1	无符号十进制	8		
	%IB2	无符号十进制	0		
	%IB3	无符号十进制	0		
	%IB4	无符号十进制	1		
	%IB5	无符号十进制	0		
	%IB6	无符号十进制	1		
	%IB7	无符号十进制	2		
	%IB8	无符号十进制	3		
	%IB9	无符号十进制	4		
	%IB10	无符号十进制	5		
	%IB11	无符号十进制	1		
	%IB12	无符号十进制	2		
	%IB13	无符号十进制	3		
	%IB14	无符号十进制	0		
	%IB15	无符号十进制	0		
	%IB16	无符号十进制	0		
	%IB17	无符号十进制	0		
	%IB18	无符号十进制	0		
	%IB19	无符号十进制	0		
	%IB20	无符号十进制	0		
	%IB21	无符号十进制	0		
	%IB22	无符号十进制	0		
	%IB23	无符号十进制	0		
	%IB24	无符号十进制	0		
	%IB25	无符号十进制	0		
	%IB26	无符号十进制	0		
	%IB27	无符号十进制	0		