

EtherCAT 转 CANopen/CAN 网关 SG-ECAT-COE/CAN

(产品手册 v1.11)



版本信息

日期	版本号	修改内容	备注
2024/10/11	v1.0	建立	
2024/10/14	v1.1	更改使用场景拓扑图	
2024/10/15	v1.11	更改使用场景拓扑图	

目录

版本信息	2
目录	3
一、功能概述	5
1.1 设备简介	5
1.2 硬件参数	6
1.3 软件参数	7
二、硬件说明	8
2.1 电源接口	8
2.2 指示灯定义	8
2.3 恢复出厂设置按键	9
2.4 配置口	9
2.5 EtherCAT 口	9
2.6 CAN 接口	9
三、ECAT-CANOPEN 软件说明	10
3.1 ECAT-CANOPEN 工作方式	10
3.2 ECAT-CANOPEN 配置软件参数	10
3.2.1 配置软件参数概述	11
3.2.2 “网关参数”	11
3.2.3 添加 CANOPEN 从站	13
3.2.4 给 CANOPEN 从站添加 PDO	14
3.2.5 给 CANOPEN 从站添加快速 SDO	15
3.3 ECAT-CANOPEN 配置软件使用说明	15
3.4 状态/数据映射表说明	16
3.4.1 “CANOPEN 从站状态映射表”	17
3.4.2 “PDO 映射表”	17
3.4.3 “SDO 映射表”	18
3.5 ECAT-CANOPEN ECAT ESI 文件	18
3.5.1 下载安装 ESI 文件	18
3.5.2 IO 数据映射	18
四、ECAT-CAN(自由口)软件说明	19
4.1 ECAT-CAN(自由口)工作方式	19
4.2 ECAT-CAN(自由口)配置软件参数	20
4.2.1 配置软件参数概述	20
4.2.2 常规参数	20
4.2.3 按 ID 收发 CAN 帧	21
4.2.4 通用收发 CAN 帧	22
4.3 ECAT-CAN(自由口)配置软件使用说明	23
4.4 ECAT-CAN(自由口) IO 数据映射偏移/长度说明	24
4.4.1 “按 ID 收发 CAN 帧” 映射偏移/长度	24
4.4.2 “通用收发 CAN 帧” 映射偏移/长度	25
4.5 ECAT-CAN(自由口) ECAT ESI 文件	26
4.5.1 下载安装 ESI 文件	26

4.5.2 IO 数据映射	26
五、ECAT-CANOPEN 应用实例	26
5.1 实例预操作	26
5.5.1 实例拓扑	26
5.5.2 配置 SV660C	26
5.5.3 配置网关	29
5.2 TwinCAT 下实例	31
5.3 CODESYS 例程	37
5.4 欧姆龙 Sysmac Studio 例程	47
5.5 汇川 H5U-A8 例程	59
六、ECAT-CAN(自由口)应用实例	69
6.1 实例预操作	69
6.2 TwinCAT 实例	70
6.3 CODESYS 例程	75
6.4 欧姆龙 Sysmac Studio 例程	86
6.5 汇川 H5U-A8 例程	97
七、产品尺寸	104
附录:	106
CANopen 通信协议简介	106
2.1 CANopen 报文结构	106
2.2 CANopen 从站设备的状态机	107
2.3 CANopen 子协议	108
2.3.1 NMT 协议	109
2.3.2 Node guard 协议	109
2.3.3 Heartbeat 协议	110
2.3.4 Bootup 协议	110
2.3.5 SDO 协议	110
2.3.6 PDO 协议	112
2.3.7 SYNC 协议	112

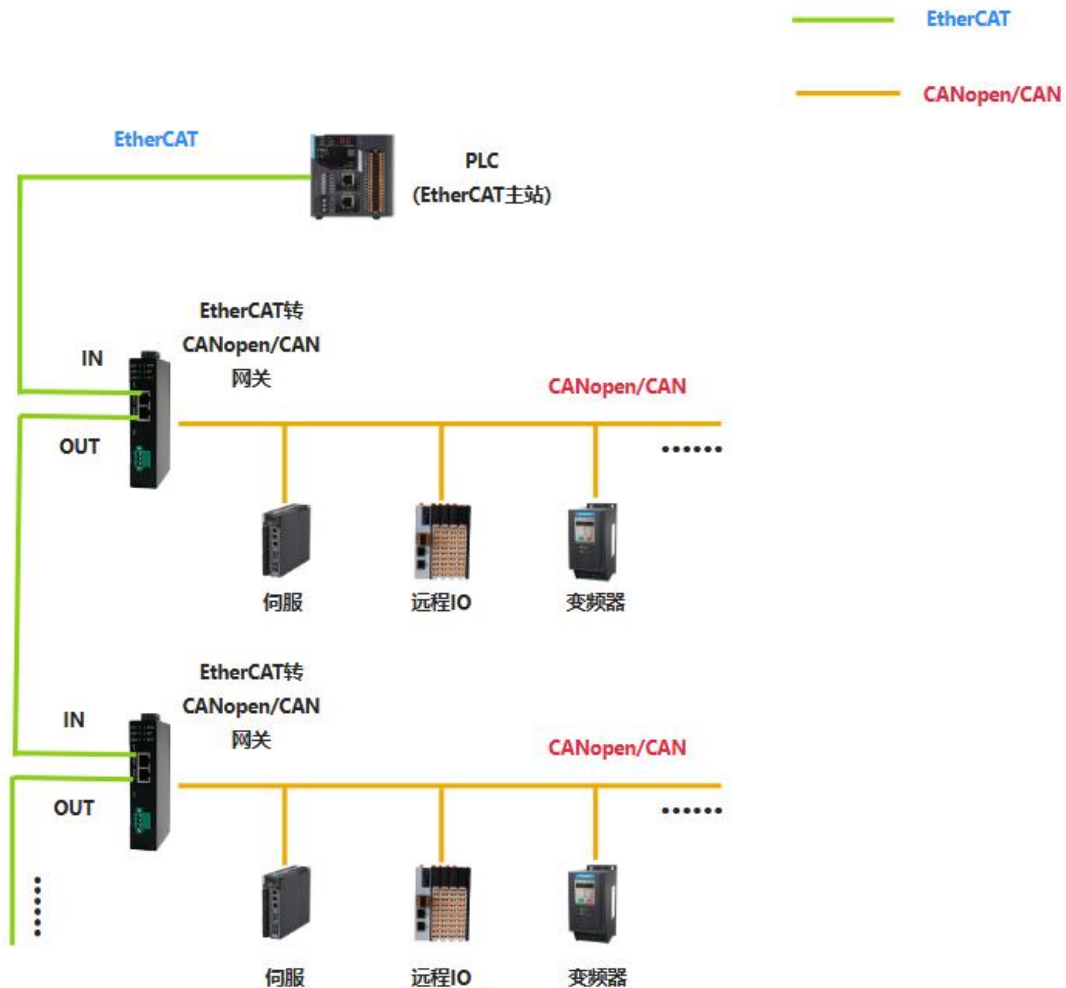
一、功能概述

1.1 设备简介

本产品是 ECAT(EtherCAT)和 CANOPEN(CAN OPEN)/CAN(自由口)网关，使用数据映射方式工作。

本产品在 ECAT 侧作为从站，接 ECAT 主站，如 TwinCAT、CODESYS、PLC 等；在 CANOPEN 侧作为 CANOPEN 主站，接 CANOPEN 设备，如伺服、变频器、电机等；在 CAN(自由口)模式下 CAN 口为自由口，用户可以自由收发 CAN 报文。

使用场景：ECAT 接口 PLC 控制 CANOPEN 接口设备。



EtherCAT 支持最多 1024 个输入字节 Input Bytes[0..1023]和 1024 个输出字节 Output Bytes[0..1023]，分为 64 个 TPDO 和 64 个 RPDO，每个 PDO 映射 16 个字

节对象。在 TwinCAT3 下显示如下：

The image displays two screenshots of the TwinCAT3 software interface, specifically the 'Process Data' tab. Both screenshots show the 'Sync Manager' on the left, the 'PDO List' in the center, and 'PDO Assignment' and 'PDO Content' on the right.

Top Screenshot (SM 2):

- Sync Manager:** SM 2 is selected, with a size of 16 and type 'Output Bytes'.
- PDO List:** A table showing PDOs for SM 2. The first eight rows (0x1600 to 0x1607) are highlighted with a red box. These are all 'Output Bytes' with a size of 16.0 and flags set to 'F'.
- PDO Assignment (0x1C12):** Shows checkboxes for 0x1600, 0x1601, and 0x1602.
- PDO Content (0x1A00):** Shows a table with one entry: Index 0x6000..., Size 16.0, Offs 0.0, Name 'Input Bytes[0..15]', Type 'ARRAY [0....', and Default (h...).

Bottom Screenshot (SM 3):

- Sync Manager:** SM 3 is selected, with a size of 16 and type 'Input Bytes'.
- PDO List:** A table showing PDOs for SM 3. The first eight rows (0x1A00 to 0x1A07) are highlighted with a red box. These are all 'Input Bytes' with a size of 16.0 and flags set to 'F'.
- PDO Assignment (0x1C12):** Shows checkboxes for 0x1600, 0x1601, and 0x1602.
- PDO Content (0x1A00):** Shows a table with one entry: Index 0x6000..., Size 16.0, Offs 0.0, Name 'Input Bytes[0..15]', Type 'ARRAY [0....', and Default (h...).

CANOPEN 侧支持 CANOPEN DS301；CAN(自由口)下支持 CAN2.0B(包含 CAN2.0A)。

1.2 硬件参数

硬件参数	参数说明
电源	9-36V(典型值 12V/120mA)，支持双电源冗余供电，带过压、过流保护
工作温度	-30~75℃
工作湿度	5%~95%无冷凝
EtherCAT 口	2 个 RJ45 以太网接口，支持 100BASE-TX/RX，MDI/MDIX 自检测，用来组成链式网络。 <u>EtherCAT 的两个网口严格区分输入输出，不是交换</u>

	<u>机!!!接线时必须输入接上一个的输出,第一个 ECAT 从站的输入接主站。</u>
CAN 口	内置 120R 终端电阻, 通过内部跳线帽使能, 带过压、过流保护, 最多支持 128 个 CAN 设备

1.3 软件参数

软件参数	参数说明
EtherCAT	采用 EtherCAT 从站芯片 AX58100。 工作在 FreeRUN 模式。 支持 COE。 支持最多 1024 个输入字节 Input Bytes[0..1023] 和 1024 个输出字节 Output Bytes[0..1023], 分为 64 个 TPDO 和 64 个 RPDO, 每个 PDO 映射 16 个字节对象。
CANOPEN	波特率 5K-1000K 支持 CAN2.0A 接收缓存: 100; 发送 FIFO: 3 最大支持从站数量: 20 最大 PDO 数量: 200 最大快速 SDO 数量: 100 支持 ECAT 主站手动 NMT 管理 CANOPEN 从站 支持接收紧急报文
CAN(自由口)	波特率 5K-1000K 支持 CAN2.0B(包含 CAN2.0A) 接收缓存: 100; 发送 FIFO: 3 最大支持按 ID 收发数量: 200 最大支持通用收发数量: 200 按 ID 收发+通用收发<=200
最大映射数据量	输入 1024 字节; 输出 1024 字节

工作参数配置	CANOPEN/CAN 端通过软件配置。 EtherCAT 端通过 ESI 文件（XML 文件）配置。
--------	--

二、硬件说明

2.1 电源接口

本网关电源接口如下图所示，支持压线端子接法和圆头电源接法，支持双电源冗余供电，支持 9-36V 输入：



接口符号	参数说明
圆头 V1	接圆头电源 9-36V（和端子 V1 不能同时接）
端子 V1/V2	接直流 9-36V 正
G	接直流 9-36V 负
PE	接大地

2.2 指示灯定义

本网关六个指示灯如下图所示：



指示灯	指示灯说明
PWR	电源指示灯，常亮说明电源正常
SYS	系统灯，常亮说明系统正常启动
OP	ECAT 状态指示灯，常亮说明网关进入 OP 状态
NC	没用到

CAN	CAN 通道数据收发闪烁
-----	--------------

2.3 恢复出厂设置按键

当配置错误导致网关工作异常可以按住之后给网关重上电，网关会清除当前所有配置。用户应该等到 SYS 闪烁再松开按键，网关自动重启恢复正常。



2.4 配置口

配置口如下：



配置口用来连接电脑，配置网关。

2.5 EtherCAT 口

EtherCAT 口如下：



EtherCAT 口	接口说明
PORT1	EtherCAT 输入口
PORT2	EtherCAT 输出口

EtherCAT 的两个网口严格区分输入输出，不是交换机！！！接线时必须输入接上一个的输出，第一个 ECAT 从站的输入接主站。

2.6 CAN 接口

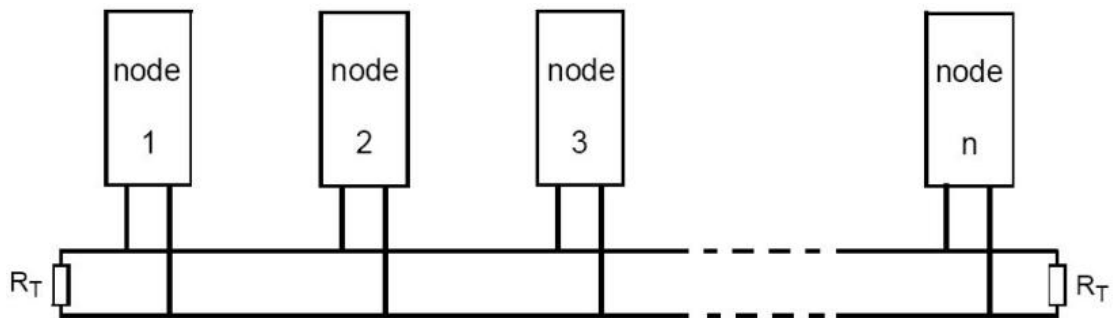
CAN 口如下：



CAN 口默认使能 120R 终端电阻，如果不使用 120R 终端电阻可以拆开壳子去掉跳线帽即可。

CAN 接口	参数说明
H	接 CAN 线 H
L	接 CAN 线 L
G	不接

CAN 网络拓扑，CAN 总线两端需要接 120R 终端电阻，中间的不需要。



三、ECAT-CANOPEN 软件说明

3.1 ECAT-CANOPEN 工作方式

网关在 ECAT 侧作为从站，在 CANOPEN 侧作为主站。即网关的 ECAT 口接 ECAT 主站，如 TwinCAT、CODESYS、PLC；网关的 CANOPEN 口接 CANOPEN 从站，例如伺服驱动器。

ECAT 主站与 CANOPEN 从站采用数据映射的方式通信，即 PLC 通过数据映射的方式控制 CANOPEN 从站，读写 CANOPEN 从站设备。

CANOPEN 需要使用软件配置 CANOPEN 工作参数，ECAT 需要根据配置软件计算出来的数据长度使能相应个数的 PDO 即可，每个 PDO 映射 16 个字节。

3.2 ECAT-CANOPEN 配置软件参数

配置软件页面如下图所示：



3.2.1 配置软件参数概述

软件页面上有 5 个页：配置网关、COE 从站状态映射表、PDO 映射表、SDO 映射表、配置 CANOPEN 从站。

其中“配置网关”页面用户用来选择网关工作参数，添加 CANOPEN 从站、在从站下面添加 PDO、在从站下面添加 SDO。

“COE 从站状态映射表”、“PDO 映射表”、“SDO 映射表”是根据用户配置的 CANOPEN 从站内容计算出来的相对于 ECAT 端的数据映射偏移地址。

“配置 CANOPEN 从站”暂时没用到。

3.2.2 “网关参数”

“网关参数”是网关工作的基本参数

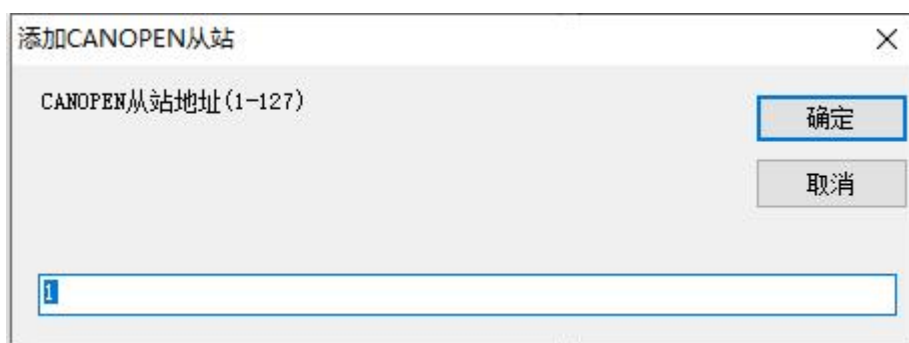
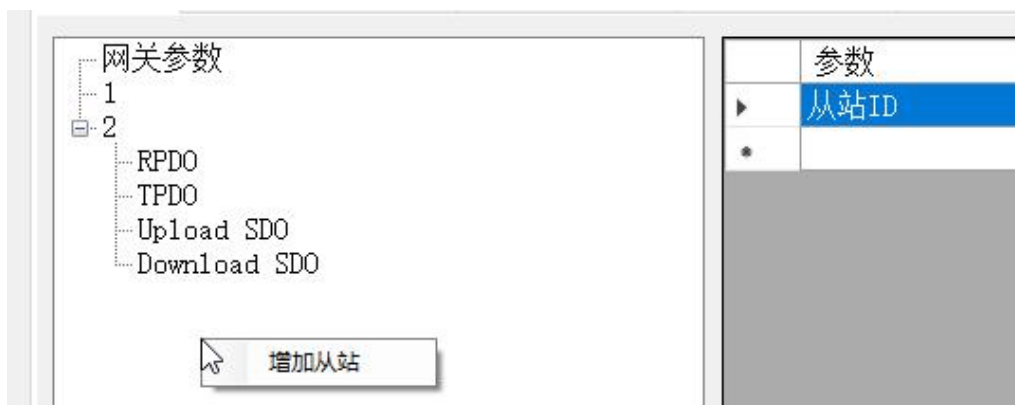
参数名称	参数内容
CAN 波特率	5000-1000000
网关 CANOPEN 站号	没意义，填 0 就行
大小端转换	不要使能大小端转换，没有意义
CANOPEN 从站状态监控	设置网关用来监控 CANOPEN 从站的方式，通过从站的心跳或节点守护
从站心跳/节点守护	设置网关多长时间收不到 CANOPEN 从站心跳或节点守护

超时时间	应答就把从站状态设为未知 0												
手动 NMT 管理从站	启用手动 NMT 管理则 CANOPEN 从站状态切换全由 ECAT 主站控制，网关不再控制 CANOPEN 从站状态切换；不启用 NMT 管理则由网关全程控制 CANOPEN 从站状态，让从站一直处于操作状态。 启用之后占用 ECAT 的 3 个字节输出 O 和一个字节输入 I。3 个 O 字节分别表示“指令序号” a、“从站 ID” b、“从站状态切换命令” c。 1 个 I 字节表示已“完成指令序号” d。 <table border="1"> <thead> <tr> <th>状态切换命令</th><th>命令含义</th></tr> </thead> <tbody> <tr> <td>0x01 启动命令</td><td>让节点进入操作状态</td></tr> <tr> <td>0x02 停止命令</td><td>让节点进入停止状态</td></tr> <tr> <td>0x80 进入预操作状态</td><td>让节点进入预操作状态</td></tr> <tr> <td>0x81 复位节点应用层</td><td>让节点的应用复位</td></tr> <tr> <td>0x82 复位节点通讯层</td><td>让节点的 CAN 和 CANopen 通讯重新初始化，一般用于总线收到干扰，导致节点总线错误被动，或者总线关闭时</td></tr> </tbody> </table> 当 ECAT 主站要切换某个 CANOPEN 从站状态时流程： <pre> if(d == a) //说明上一条指令已经完成 { b = 从站 id; c = 状态切换命令; a += 1; } </pre>	状态切换命令	命令含义	0x01 启动命令	让节点进入操作状态	0x02 停止命令	让节点进入停止状态	0x80 进入预操作状态	让节点进入预操作状态	0x81 复位节点应用层	让节点的应用复位	0x82 复位节点通讯层	让节点的 CAN 和 CANopen 通讯重新初始化，一般用于总线收到干扰，导致节点总线错误被动，或者总线关闭时
状态切换命令	命令含义												
0x01 启动命令	让节点进入操作状态												
0x02 停止命令	让节点进入停止状态												
0x80 进入预操作状态	让节点进入预操作状态												
0x81 复位节点应用层	让节点的应用复位												
0x82 复位节点通讯层	让节点的 CAN 和 CANopen 通讯重新初始化，一般用于总线收到干扰，导致节点总线错误被动，或者总线关闭时												
同步报文周期	同步报文周期												
接收 CANOPEN 从站紧急报文	接收紧急报文网关会把接收到的 CANOPEN 从站紧急报文传输到 ECAT 主站，占用 ECAT 的 11 个字节输入 I 和 1 个												

	字节输出 O。 11 个字节 I 分别表示“紧急报文序号” a 一个字节、“紧急报文帧 ID” b 两个字节、“紧急报文内容” c 八个字节。 1 个字节输出 O 表示“紧急报文已读取序号” d 当 ECAT 主站要读取有无紧急报文时： <pre>if(d != a) //有新的紧急报文 { 紧急报文 id = b; 紧急报文内容 = c; d = a; }</pre>
RPDO 发送周期	网关发送 RPDO 报文的周期，0 表示数据改变时发
主站断线 RPDO 输出	网关检测到 ECAT 不处于 OP 状态或 100ms 没有收到主站数据时 RPDO 是输出 0 还是保持之前的输出
TPDO 接收超时清零	网关超时没有收到某条 TPDO 数据时清零该 TPDO 已读数据，0 表示不启用。
SDO 应答超时时间	SDO 应答超时时间
快速 upload SDO 周期	快速 upload SDO 周期
快速 upload SDO 错误/应答超时	当 CANOPEN 从站响应某条 upload SDO 错误或超时时是否清零该 SDO 已读数据
快速 download SDO 周期	网关发送快速 download SDO 的周期，0 表示数据改变时发
快速 download SDO 重传次数	仅当上者为 0 时有效，当快速 download SDO 响应错误或超时最大重试次数
主站断线 快速 download SDO 输出	网关检测到 ECAT 不处于 OP 状态或 100ms 没有收到主站数据时快速 download SDO 是输出 0 还是保持之前的输出

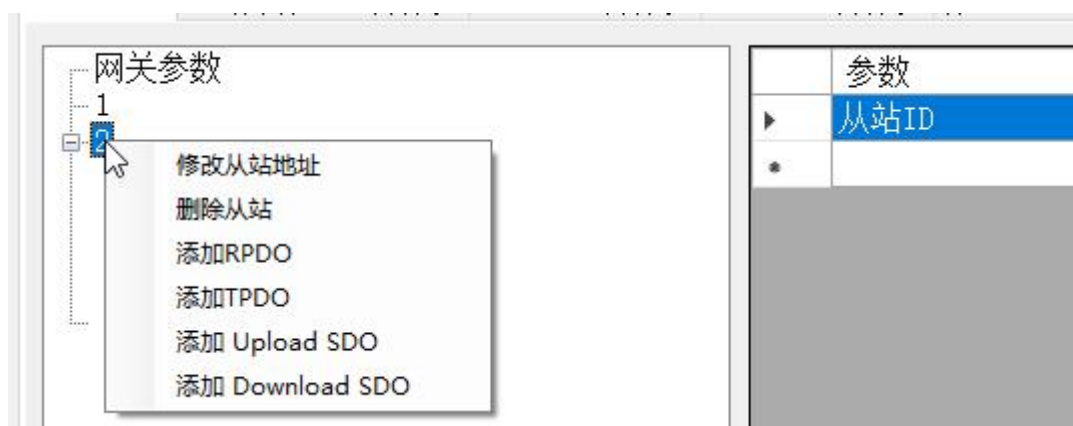
3.2.3 添加 CANOPEN 从站

如下图空白位置右击可添加从站



在弹出的对话框输入从站 ID，1-127，最多可以添加 20 个从站。

右击添加好的从站 ID 可以删除、修改 ID、添加 PDO、添加 SDO。



3.2.4 给 CANOPEN 从站添加 PDO

右击从站 ID 即可添加 PDO, RPDO 和 TPDO 是相对于 CANOPEN 从站来说的，即 RPDO 对应网关 ECAT 的 Output Bytes，TPDO 对应网关 ECAT 的 Input Bytes。

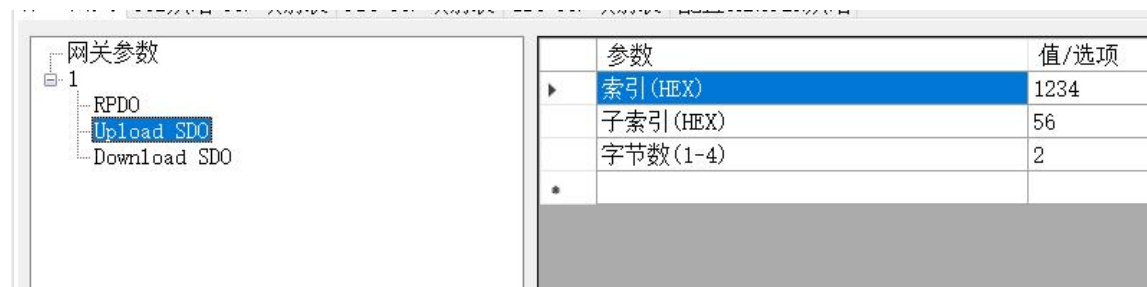


RPDO/TPDO 都需要设置 COB-ID 和映射参数。COB-ID 跟 CANOPEN 设备相关，一般在设备的说明书或 ESD 文件。如果说明书或 EDS 文件有准确的映射参数就填写实际映射参数，如果说明书或 ESD 没有映射参数（一般不会没有）就根据 PDO 的数据部分字节长度填写，填写规则如下：

0x00000008 表示一个字节，0x00000010 表示两个字节，0x00000020 表示四个字节。根据数据长度填写，比如 PDO 数据长度有 6 个字节就在映射参数 1 写 00000020，映射参数 2 写 00000010，映射参数 3-8 都写 0。

3.2.5 给 CANOPEN 从站添加快速 SDO

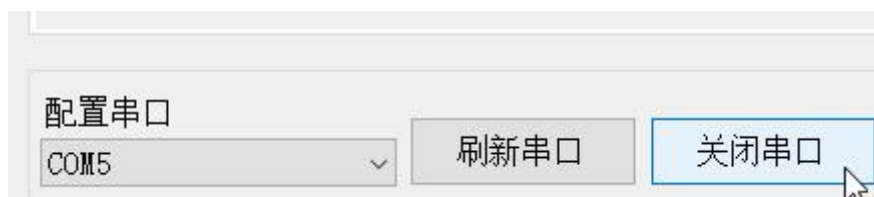
右击从站 ID 即可添加快速 SDO，支持 upload SDO 和 download SDO。



3.3 ECAT-CANOPEN 配置软件使用说明

配置软件用来配置本网关，流程如下：

- ①打开软件
- ②选择和网关连接的 USB 串口并打开



- ③在页面上设置要配置的参数，如果配置参数很多建议在软件上填好参数后

点击一下 保存配置，这样再次开启软件可以通过 读取配置 来加载刚保存的参数。

④点击 配置网关，弹出配置网关成功就说明配置完成了，配置内容比较多的时候配置时间会比较长，需要多等一下直到弹出配置成功或失败提示。



3.4 状态/数据映射表说明

ECAT 和 CANOPEN 是通过数据映射的方式交换数据的。当用户在软件页面上填好参数后点击一下 计算 IO 数据映射，这时软件会根据软件页面上的参数自动计算映射地址。映射地址分为三个表：“CANOPEN 从站状态映射表”、“PDO 映射表”、“SDO 映射表”。

映射地址是网关 ECAT 的 Input Bytes[0..1023]和 Output Bytes[0..1023]的地址。

General EtherCAT DC Process Data Plc Startup CoE - Online Online

Sync Manager:

SM	Size	Type	Flags
0	256	Mbx...	
1	256	MbxIn	
2	16	Outp...	
3	16	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1600	16.0	Output Bytes[0..15]	F	2	0
0x1601	16.0	Output Bytes[16..31]	F		0
0x1602	16.0	Output Bytes[32..47]	F		0
0x1603	16.0	Output Bytes[48..63]	F		0
0x1604	16.0	Output Bytes[64..79]	F		0
0x1605	16.0	Output Bytes[80..95]	F		0
0x1606	16.0	Output Bytes[96..111]	F		0
0x1607	16.0	Output Bytes[112..127]	F		0

PDO Assignment (0x1C12):

☒ 0x1600
☐ 0x1601
☐ 0x1602

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h...
0x6000...	16.0	0.0	Input Bytes[0..15]	ARRAY [0...	

General EtherCAT DC Process Data Plc Startup CoE - Online Online

Sync Manager:

SM	Size	Type	Flags
0	256	Mbx...	
1	256	MbxIn	
2	16	Outp...	
3	16	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	16.0	Input Bytes[0..15]	F	3	0
0x1A01	16.0	Input Bytes[16..31]	F		0
0x1A02	16.0	Input Bytes[32..47]	F		0
0x1A03	16.0	Input Bytes[48..63]	F		0
0x1A04	16.0	Input Bytes[64..79]	F		0
0x1A05	16.0	Input Bytes[80..95]	F		0
0x1A06	16.0	Input Bytes[96..111]	F		0
0x1A07	16.0	Input Bytes[112..127]	F		0

PDO Assignment (0x1C13):

☒ 0x1A00
☐ 0x1A01
☐ 0x1A02

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h...
0x6000...	16.0	0.0	Input Bytes[0..15]	ARRAY [0...	

见后面案例。

3.4.1 “CANOPEN 从站状态映射表”

“CANOPEN 从站状态映射表”如下：在配置页面添加了两个从站，则第一个从站的状态（1 个字节）放在 Input Bytes[0]，第二个从站的状态（1 个字节）放在 Input Bytes[1]。

ECAT-CANOPEN v3.0

配置网关 COE从站状态映射表 PDO映射表 SDO映射表 配置CANOPEN从站

参数	输入映射地址/字节长度	输出映射地址/字节长度
COE 从站 1 状态	0 / 1	
COE 从站 2 状态	1 / 1	
*		

CANOPEN 从站状态码	状态
0	CANOPEN 从站掉线
4	CANOPEN 从站处于 停止 状态
5	CANOPEN 从站处于 操作 状态
127	CANOPEN 从站处于 预操作 状态

ECAT 主站可以根据 CANOPEN 从站状态确定通信情况。

3.4.2 “PDO 映射表”

“PDO 映射表”如下：在配置页面添加了 4 个 PDO。自动计算出来 4 个 PDO 的共 8 个映射参数的数据映射到网关 ECAT 的 Input Bytes 和 Output Bytes 的地址。

ECAT-CANOPEN v3.0

配置网关							COE从站状态映射表		PDO映射表		SDO映射表		配置CANOPEN从站	
	所属从站	PDO类型	COB-ID	映射参数(HEX)	输入映射地址/字节长度	输出映射地址/字节长度								
▶	1	TPDO	181	00000010	2 / 2									
	1	TPDO	181	00000008	4 / 1									
	1	RPDO	201	00000010								0 / 2		
	1	RPDO	201	00000010								2 / 2		
	2	RPDO	202	00000008								4 / 1		
	2	RPDO	202	00000008								5 / 1		
	2	TPDO	182	00000020	5 / 4									
	2	TPDO	182	00000010	9 / 2									
*														

3.4.3 “SDO 映射表”

“SDO 映射表”如下：在配置页面添加了两个 SDO。自动计算出来两个 SDO 的数据映射到网关 ECAT 的 Input Bytes 和 Output Bytes 的地址。

配置网关

COE从站状态映射表

PDO映射表

SDO映射表

配置CANOPEN从站

	所属从站	SDO类型	索引(HEX)	子索引(HEX)	字节长度	输入映射地址/字节长度	输出映射地址
▶	1	Upload SDO	1234	56	2	11 / 2	
	1	Download SDO	2345	67	3		6 / 3
*							

3.5 ECAT-CANOPEN ECAT ESI 文件

3.5.1 下载安装 ESI 文件

在我司网站下载 ECAT ESI 文件，安装或导入到 ECAT 主站编程软件。

3.5.2 IO 数据映射

软件计算的顺序是先计算从站状态，再计算 PDO，最后计算 SDO，所以如果有 SDO 则最后的地址在“SDO 映射表”，没有 SDO 最后的地址在“PDO 映射表”，没有 PDO 最后的地址在“CANOPEN 从站状态映射表”。

例如：在软件页面上填写完参数之后，点击计算IO数据映射，之后来到 SDO 映射表页面，看到“输入映射地址/字节长度”最后一条是“6/2”说明 ECAT 端上需要的输入数据长度为 $6+2=8$ ；“输出映射地址/字节长度”最后一条是“8/3”，说明 ECAT 端需要的输出数据长度是 $8+3=11$ 。则 ECAT 端只需要使能 0x1A00、0x1600 两个 PDO 即可，0x1A00 映射 Input Bytes[0..15]，0x1600 映射 Output Bytes[0..15]。

General EtherCAT DC Process Data Plc Startup CoE - Online Online

Sync Manager:

SM	Size	Type	Flags
0	256	Mbx...	
1	256	MbxIn	
2	16	Outp...	
3	16	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1600	16.0	Output Bytes[0..15]	F	2	0
0x1601	16.0	Output Bytes[16..31]	F		0
0x1602	16.0	Output Bytes[32..47]	F		0
0x1603	16.0	Output Bytes[48..63]	F		0
0x1604	16.0	Output Bytes[64..79]	F		0
0x1605	16.0	Output Bytes[80..95]	F		0
0x1606	16.0	Output Bytes[96..111]	F		0
0x1607	16.0	Output Bytes[112..127]	F		0

PDO Assignment (0x1C12):

☒ 0x1600
☐ 0x1601
☐ 0x1602

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h...
0x6000...	16.0	0.0	Input Bytes[0..15]	ARRAY [0...	

General EtherCAT DC Process Data Plc Startup CoE - Online Online

Sync Manager:

SM	Size	Type	Flags
0	256	Mbx...	
1	256	MbxIn	
2	16	Outp...	
3	16	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	16.0	Input Bytes[0..15]	F	3	0
0x1A01	16.0	Input Bytes[16..31]	F		0
0x1A02	16.0	Input Bytes[32..47]	F		0
0x1A03	16.0	Input Bytes[48..63]	F		0
0x1A04	16.0	Input Bytes[64..79]	F		0
0x1A05	16.0	Input Bytes[80..95]	F		0
0x1A06	16.0	Input Bytes[96..111]	F		0
0x1A07	16.0	Input Bytes[112..127]	F		0

PDO Assignment (0x1C13):

☒ 0x1A00
☐ 0x1A01
☐ 0x1A02

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h...
0x6000...	16.0	0.0	Input Bytes[0..15]	ARRAY [0...	

配置软件上的“输入映射地址和长度”就是 Input Bytes 的起始和长度，“输出映射地址和长度”就是 Output Bytes 的起始和长度。

例如“输入映射地址和长度”的值是“6/2”，说明对应到 Input Bytes[6..7]；“输出映射地址/字节长度”的值是“8/3”，说明对应到 Output Bytes[8..10]。

四、ECAT-CAN(自由口)软件说明

4.1 ECAT-CAN(自由口)工作方式

网关在 ECAT 侧做从站，在 CAN(自由口)侧做 CAN 自由口。即网关的 EtherCAT 口接 EtherCAT 主站，如 TwinCAT、CODESYS、PLC；网关的 CAN 口接 CAN 设备。

ECAT 主站与 CAN 采用数据映射的方式通信，即 ECAT 主站通过数据映射的方式收发 CAN 数据包。

CAN(自由口)需要使用软件配置工作参数，ECAT 端需要根据配置软件计算出来的数据长度使能相应个数的 PDO 即可，每个 PDO 映射 16 个字节。

4.2 ECAT-CAN(自由口)配置软件参数

配置软件页面如下图所示：

ECAT-CAN v1.1

CAN波特率: 1000000 (5000-1000000) 数据帧发送周期(按ID收发CAN帧): 0 (0-65535)ms

远程帧发送周期(按ID收发CAN帧): 1000 (1-65535)ms 数据帧接收超时(按ID收发CAN帧): 0 (0-65535)ms

按ID收发CAN帧 通用收发CAN帧

ID收发CAN数量: 10 (1-200)

序号	发送/接收	帧ID类型	数据/远程帧	数据长度	帧ID (HEX)	数据偏移/长度
1	None	标准帧	数据帧	0	0	
2	None	标准帧	数据帧	0	0	
3	按ID发送	标准帧	数据帧	0	0	
4	按ID接收	标准帧	数据帧	0	0	
5	None	标准帧	数据帧	0	0	
6	None	标准帧	数据帧	0	0	
7	None	标准帧	数据帧	0	0	
8	None	标准帧	数据帧	0	0	
9	None	标准帧	数据帧	0	0	
10	None	标准帧	数据帧	0	0	

计算IO数据映射 保存配置 读取配置

配置串口: COM1 刷新串口 打开串口 获取网关参数 配置网关 重启设备 恢复出厂设置

4.2.1 配置软件参数概述

软件页面上分为四个部分：**常规参数**、**按 ID 收发 CAN 帧**、**通用收发 CAN 帧**及**配置串口**部分部分。

“常规参数”部分用户用来设置 CAN 波特率、CAN 收发周期等。

“按 ID 收发 CAN 帧”部分用来设置要收发的 CAN 帧 ID、类型、数据长度。

“通用收发 CAN 帧”部分用来设置要使用的通用 CAN 收发数量。

“网关列表”部分用来配置网关。

4.2.2 常规参数

CAN波特率: 1000000 (5000-1000000) 数据帧发送周期(按ID收发CAN帧): 0 (0-65535)ms

远程帧发送周期(按ID收发CAN帧): 1000 (1-65535)ms 数据帧接收超时(按ID收发CAN帧): 0 (0-65535)ms

“常规参数”是网关工作的基本参数

参数名称	参数内容
CAN 波特率	5000-1000000bps
数据帧发送周期	0-65535ms, 0 代表数据改变了才发送

远程帧发送周期	1-65535ms，远程帧不携带数据，所以只能周期发送
数据帧接收超时	0-65535ms，当多久接收不到对应 ID 的数据帧时把对应数据清零，0 代表不启用

4.2.3 按 ID 收发 CAN 帧

ECAT-CAN v1.1

CAN波特率 (5000-1000000) 数据帧发送周期(按ID收发CAN帧)

远程帧发送周期(按ID收发CAN帧) (1-65535)ms 数据帧接收超时(按ID收发CAN帧)

按ID收发CAN帧 通用收发CAN帧

ID收发CAN数量 (1-200)

序号	发送/接收	帧ID类型	数据/远程帧	数据长度	帧ID(HEX)	数据偏移/长度
1	按ID发送	标准帧	数据帧	1	12	输出+0/1
2	按ID发送	扩展帧	远程帧	0	12345678	---
3	按ID接收	扩展帧	数据帧	8	11223345	输入+0/8
4	None	标准帧	数据帧	0	0	
5	None	标准帧	数据帧	0	0	

参数名称	参数内容
发送/接受	配置发送 CAN 帧或接收 CAN 帧
帧 ID 类型	标准帧和扩展帧
数据/远程帧	数据帧和远程帧，接收模式下不支持远程帧，因为远程帧不携带数据，没意义
数据长度	CAN 帧数据部分长度 0-8，远程帧固定为 0，因为远程帧不携带数据
帧 ID(HEX)	CAN 帧 ID，十六进制表示。标准帧 0-7FF，扩展帧 0-1FFFFFFF。
数据偏移/长度	自动计算出来的

在“按 ID 收发 CAN 帧”标签下的“ID 收发 CAN 数量”填写所需要的按 ID 收发 CAN 帧的数量，可以多写，用不到的为 None 即可。

如上图：网关每 1000ms 发送一次标准数据帧，ID 为 0x012，数据长度为 1，数据为 Output Bytes[0]；每 1000ms 发送一次扩展远程帧，ID 为 0x12345678；当网关收到扩展数据帧，ID 为 0x11223344 且数据长度为 8 的 CAN 帧时会把数据放到 Input Bytes[0..7]。

4.2.4 通用收发 CAN 帧

按ID收发CAN帧 通用收发CAN帧							
通用收发CAN数量		10 (1-200)					
序号	发送/接收	PLC序号偏移/长度	网关序号偏移/长度	帧标志偏移/长度	帧ID偏移/长度	数据偏移/长度	
1	通用发送	输出+1/1	输入+0/1	输出+2/1	输出+3/4	输出+7/8	
2	通用发送	输出+15/1	输入+1/1	输出+16/1	输出+17/4	输出+21/8	
3	通用接收	输出+29/1	输入+2/1	输入+3/1	输入+4/4	输入+8/8	
4	None						
5	None						

参数名称	参数内容
发送/接受	配置发送 CAN 帧或接收 CAN 帧
PLC 序号偏移/长度	占 ECAT Output Bytes 一个字节
网关序号偏移/长度	占 ECAT Input Bytes 一个字节
帧标志偏移/长度	占 ECAT Output/Input Bytes 一个字节
帧 ID 偏移/长度	占 ECAT Output/Input Bytes 四个字节
数据偏移/长度	占 ECAT Output/Input Bytes 八个字节

在“通用收发 CAN 帧”标签下的“通用收发 CAN 数量”填写所需要的通用收发 CAN 帧的数量，可以多写，用不到的为 None 即可。

“通用收发 CAN 帧”相较于“按 ID 收发 CAN 帧”在使用起来更加灵活。“按 ID 收发 CAN 帧”只能发送和接受固定的 CAN 报文，使用场景有限制。“通用收发 CAN 帧”用户可以在 ECAT 主站自定义收发 CAN 帧的类型、ID、数据等，在使用上相对复杂些，但更灵活。

用户使用“通用发送”发送一包 CAN 流程如下：

①用户先找到一个网关序号等于 PLC 序号的“通用发送”(用户可以添加多个通用发送)

②在相应的帧标志、ID、数据对应的 Output Bytes 填写好内容

③把 PLC 序号+1

④网关检测到 PLC 序号改变，代表有一条 CAN 帧要发送，之后解析 CAN 标志、帧 ID、帧数据并发送 CAN 帧，当发送 CAN 帧成功之后把网关序号=PLC 序号。PLC 检测到网关序号等于了 PLC 序号，代表 CAN 帧发送成功，准备发送下一条。

用户在 ECAT 主站使用“通用接收”接受一包 CAN 流程如下：

①网关收到 CAN 帧在“按 ID 收发”里面寻找有没有匹配的接收，如果没

有则准备放到“通用接收”的接收。

②网关先找到一个 PLC 序号等于网关序号的“通用接收”(用户可以添加多个通用接收)。

③网关把 CAN 帧的标志、ID、数据放到对应的 Input Bytes。

④网关把网关序号+1

⑤PLC 检测到网关序号改变，代表“通用接收”有一条新的 CAN 帧。PLC 去处理完之后把 PLC 序号=网关序号。网关检测到 PLC 序号等于了网关序号说明 PLC 已经读走了这条 CAN 帧，准备填入下一条。

帧标志：最高位（第 7 位）0 代表标准帧，1 代表扩展帧；第 6 位 0 代表数据帧，1 代表远程帧；第 3-0 位代表数据长度（0-8）。

帧 ID：CAN 帧 ID，标准帧低 11 位有效，扩展帧低 29 位有效。

帧数据：数据部分 0-8 个字节。

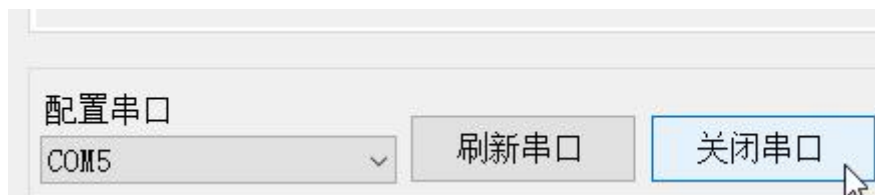
这一段不容易理解，见后面案例。

4.3 ECAT-CAN(自由口)配置软件使用说明

配置用来配置本网关，流程如下：

①打开软件

②选择和网关连接的 USB 串口并打开



③在页面上设置要配置的参数，如果配置参数很多建议在软件上填好参数后点击一下 保存配置，这样再次开启软件可以通过 读取配置来加载刚保存的参数。

④点击 配置网关，弹出配置网关成功就说明配置完成了，配置内容比较多的时候配置时间会比较长，需要多等一下直到弹出配置成功或失败提示。



4.4 ECAT-CAN(自由口) IO 数据映射偏移/长度说明

ECAT 主站和 CAN(自由口)是通过数据映射的方式交换数据的。当用户在软件页面上填好参数后点击一下计算IO 数据映射，这时软件会根据软件页面上的参数自动计算映射地址。“按 ID 收发 CAN 帧”和“通用收发 CAN 帧”都有对应的偏移地址和长度。

映射地址是网关 ECAT 的 Input Bytes[0..1023]和 Output Bytes[0..1023]的地址。

4.4.1 “按 ID 收发 CAN 帧” 映射偏移/长度

“按 ID 收发 CAN 帧”映射表如下例：

按ID收发CAN帧

通用收发CAN帧

ID收发CAN数量

10

(1-200)

	序号	发送/接收	帧ID类型	数据/远程帧	数据长度	帧ID(HEX)	数据偏移/长度
	1	按ID发送	标准帧	数据帧	1	00000012	Q+0/1
	2	按ID发送	扩展帧	远程帧	0	12345678	Q+1/0
	3	按ID接收	扩展帧	数据帧	8	11223344	I+0/8
	4	None	标准帧	数据帧	0	0	

“按 ID 收发 CAN 帧”页面添加了三条 CAN，两条“按 ID 发送”和一条“按 ID 接收”。

General	EtherCAT	DC	Process Data	Plc	Startup	CoE - Online	Online
Sync Manager:							
SM	Size	Type	Flags	PDO List:			
0	256	Mbx...		Index	Size	Name	Flags SM SU
1	256	MbxIn		0x1600	16.0	Output Bytes[0..15]	F 2 0
2	16	Outp...		0x1601	16.0	Output Bytes[16..31]	F 0
3	16	Inputs		0x1602	16.0	Output Bytes[32..47]	F 0
				0x1603	16.0	Output Bytes[48..63]	F 0
				0x1604	16.0	Output Bytes[64..79]	F 0
				0x1605	16.0	Output Bytes[80..95]	F 0
				0x1606	16.0	Output Bytes[96..111]	F 0
				0x1607	16.0	Output Bytes[112..127]	F 0
PDO Assignment (0x1C12):				PDO Content (0x1A00):			
☒ 0x1600				Index	Size	Offs	Name Type Default (h...
☐ 0x1601				0x6000...	16.0	0.0	Input Bytes[0..15] ARRAY [0....
☐ 0x1602							

General EtherCAT DC Process Data Plc Startup CoE - Online Online

Sync Manager:

SM	Size	Type	Flags
0	256	Mbx...	
1	256	MbxIn	
2	16	Outp...	
3	16	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	16.0	Input Bytes[0..15]	F	3	0
0x1A01	16.0	Input Bytes[16..31]	F		0
0x1A02	16.0	Input Bytes[32..47]	F		0
0x1A03	16.0	Input Bytes[48..63]	F		0
0x1A04	16.0	Input Bytes[64..79]	F		0
0x1A05	16.0	Input Bytes[80..95]	F		0
0x1A06	16.0	Input Bytes[96..111]	F		0
0x1A07	16.0	Input Bytes[112..127]	F		0

PDO Assignment (0x1C13):

☒ 0x1A00
☐ 0x1A01
☐ 0x1A02

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (h...
0x6000...	16.0	0.0	Input Bytes[0..15]	ARRAY [0...	

序号 1 的“按 ID 发送”是标准数据帧，ID 为 012H，数据长度是一个字节，要发送的数据内容放在 Output Bytes[0]。

序号 2 的“按 ID 发送”是扩展远程帧，ID 是 12345678H，**远程帧不携带数据**。

序号 3 的“按 ID 接收”是扩展数据帧，ID 为 11223344H，数据长度是八个字节，网关把接收的此 ID 的数据放到 Input Bytes[0..7]。

4.4.2 “通用收发 CAN 帧” 映射偏移/长度

“通用收发 CAN 帧”映射表如下例：

按 ID 收发 CAN 帧		通用收发 CAN 帧					
通用收发 CAN 数量		10 (1-200)					
序号	发送/接收	PLC 序号偏移/长度	网关序号偏移/长度	帧标志偏移/长度	帧 ID 偏移/长度	数据偏移/长度	
1	通用发送	输出+1/1	输入+8/1	输出+2/1	输出+3/4	输出+7/8	
2	通用发送	输出+15/1	输入+9/1	输出+16/1	输出+17/4	输出+21/8	
3	通用接收	输出+29/1	输入+10/1	输入+11/1	输入+12/4	输入+16/8	
4	None						
5	None						

“通用收发 CAN 帧”页面添加了三条 CAN 收发，两条“通用发送”和一条“通用接收”。

序号 1 的“通用发送”的 PLC 序号在 Output Bytes[1]，网关序号在 Input Bytes[8]，帧标志在 Output Bytes[2]，帧 ID 在 Output Bytes[3..6]，帧数据在 Output Bytes[7..14]。

序号 2 的“通用发送”参考上者。

序号 3 的“通用接收”的 PLC 序号在 Output Bytes[29]，网关序号在 Input Bytes[10]，帧标志在 Input Bytes[11]，帧 ID 在 Input Bytes[12..15]，帧数据在 Input Bytes[16..23]。

4.5 ECAT-CAN(自由口) ECAT ESI 文件

4.5.1 下载安装 ESI 文件

在我司网站下载 ECAT ESI 文件，安装或导入到 ECAT 主站编程软件。

4.5.2 IO 数据映射

软件计算的顺序是先计算“按 ID 收发 CAN 帧”，再计算“通用收发 CAN 帧”。

例如：在软件页面上填写完参数之后，点击计算 IO 数据映射，之后来到“通用收发 CAN 帧”页面，计算出来最大需要的输出字节数量和输入字节数量。

按 ID 收发 CAN 帧		通用收发 CAN 帧					
通用收发 CAN 数量		10	(1-200)				
序号	发送/接收	PLC 序号偏移/长度	网关序号偏移/长度	帧标志偏移/长度	帧 ID 偏移/长度	数据偏移/长度	
1	通用发送	输出+1/1	输入+8/1	输出+2/1	输出+3/4	输出+7/8	
2	通用发送	输出+15/1	输入+9/1	输出+16/1	输出+17/4	输出+21/8	
3	通用接收	输出+29/1	输入+10/1	输入+11/1	输入+12/4	输入+16/8	
4	None						
5	None						

配置软件上的输入+x/y 对应网关 ECAT 的 Input Bytes[]，输出+x/y 对应网关 ECAT 的 Output Bytes[]。

例如“输入+16/8”对应到 Input Bytes[16..23]；“输出+29/1”对应到 Output Bytes[29]。

五、ECAT-CANOPEN 应用实例

5.1 实例预操作

5.5.1 实例拓扑

实例拓扑如下图，ECAT 主站控制 CANOPEN 伺服走轮廓速度模式。



ECAT 主站分别使用 TwinCAT3、CODESYS、欧姆龙 NX1P2 和汇川 H5U-A8，CANOPEN 伺服器驱动器使用汇川 SV660C。

5.5.2 配置 SV660C

SV660C 说明书中轮廓速度模式推荐配置如下：

2.6.3 推荐配置

轮廓速度模式(pv)，基本配置如下：

RPDO	TPDO	备注
6040h: 控制字control word	6041h: 状态字 status word	必须
60FFh: 目标速度target Velocity	-	必须
-	6064h: 位置反馈 position actual value	可选
-	606Ch: 实际速度velocity actual value	可选
6083h: 轮廓加速度 profile acceleration	-	可选
6084h: 轮廓减速度profile deceleration	-	可选
6060h: 模式选择modes of operation	6061h: 运行模式显示 modes of operation display	可选

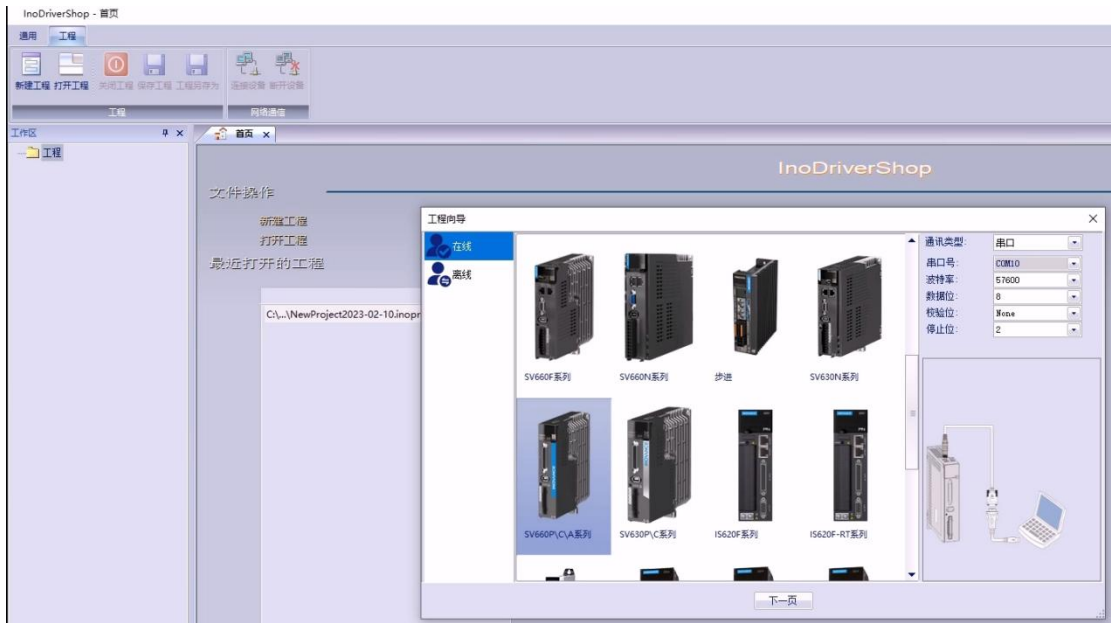
后面实例如下配置：

汇川 SV660C		
CAN 波特率	500kbps	
模式	轮廓速度模式(PV)	
DownLoad SDO	索引 0x6060，子索引 0x00，长度 1 字节	
Upload SDO	索引 0x6061，子索引 0x00，长度 1 字节	
RPDO	0x201	控制字索引是 0x6040，子索引是 0x00，长度是两个字节； 目标速度索引是 0x60FF，子索引是 0x00，长度是 4 个字节
TPDO	0x181	状态字索引是 0x6041，子索引是 0x00，长度是两个字节； 实际速度索引是 0x606C，子索引是 0x00，长度是 4 个字节

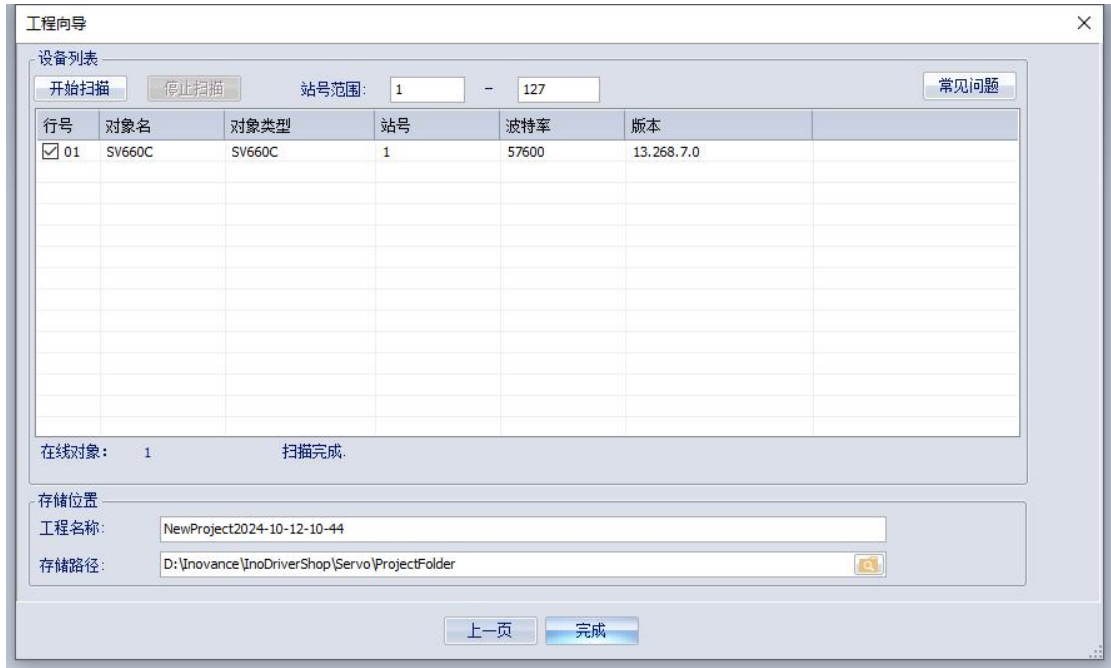
实例开始之前需要使用汇川伺服器驱动器配套软件配置一下伺服驱动器所使能的 PDO。一般伺服驱动器厂家都会提供配套软件。

①按 SV660C 驱动器手册接线，并将 232 口连接到电脑

②安装并打开汇川 InoDriverShop 软件，新建工程，选择驱动器类型和串口号



③扫描伺服驱动器



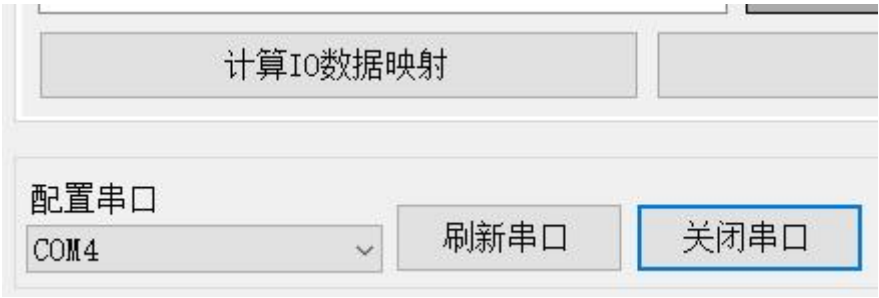
④设置 RPDO 和 TPDO 参数

轴号	功能码ID	描述	设定值	当前值
轴1	H2D.00	同步报文C08-ID (0x1005h)	—	0x00000080
轴1	H2D.02	同步循环周期 (0x1006h)	—	0x00000000
轴1	H2D.04	节点守护时间 (0x100Ch)	—	0
轴1	H2D.05	寿命因子 (0x100bh)	—	0
轴1	H2D.06	紧急报文C08-ID (0x1014h)	—	0x00000000
轴1	H2D.08	消费者心跳时间1 (0x10...	—	0x00000000
轴1	H2D.10	消费者心跳时间2 (0x10...	—	0x00000000
轴1	H2D.12	消费者心跳时间3 (0x10...	—	0x00000000
轴1	H2D.14	消费者心跳时间4 (0x10...	—	0x00000000
轴1	H2D.16	消费者心跳时间5 (0x10...	—	0x00000000
轴1	H2D.18	生产者心跳时间 (0x1017h)	—	0
轴1	H2D.19	错误行为对象-通信错...	—	0
轴1	H2D.20	RPD01的C08-ID (0x1400...	0x00000200	0x00000200
轴1	H2D.22	RPD01的传输类型 (0x14...	—	254
轴1	H2D.23	RPD02的C08-ID (0x1401...	—	0x80000300
轴1	H2D.25	RPD02的传输类型 (0x14...	—	255
轴1	H2D.26	RPD03的C08-ID (0x1402...	—	0x80000400
轴1	H2D.28	RPD03的传输类型 (0x14...	—	255
轴1	H2D.29	RPD04的C08-ID (0x1403...	—	0x80000500
轴1	H2D.31	RPD04的传输类型 (0x14...	—	255
轴1	H2D.32	RPD01有效映射对象个...	—	2
轴1	H2D.33	RPD01映射对象1 (0x160...	—	0x60400010
轴1	H2D.35	RPD01映射对象2 (0x160...	—	0x60FF0020
轴1	H2D.37	RPD01映射对象3 (0x160...	—	0x00000000
轴1	H2D.39	RPD01映射对象4 (0x160...	—	0x00000000
轴1	H2D.41	RPD01映射对象5 (0x160...	—	0x00000000
轴1	H2D.43	RPD01映射对象6 (0x160...	—	0x00000000
轴1	H2D.45	RPD01映射对象7 (0x160...	—	0x00000000
轴1	H2D.47	RPD01映射对象8 (0x160...	—	0x00000000

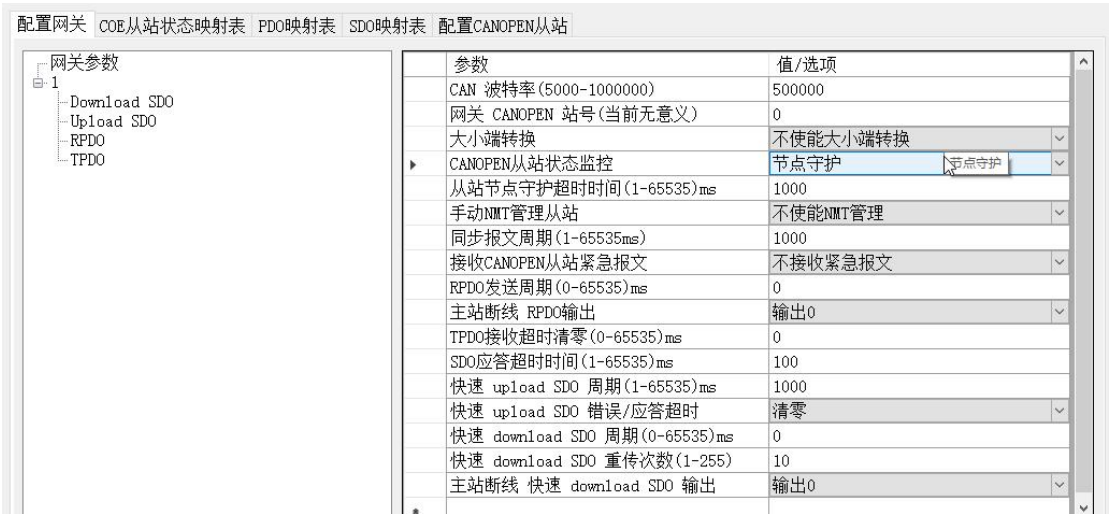
轴号	功能码ID	描述	设定值	当前值
轴1	H2E.00	TPD01的C08-ID (0x1800...	0x00000180	0x00000180
轴1	H2E.02	TPD01的传输类型 (0x18...	—	254
轴1	H2E.03	TPD01的禁止时间 (0x18...	—	0x0064
轴1	H2E.04	TPD01的事件计时器 (0x...	—	0x0000
轴1	H2E.05	TPD02的C08-ID (0x1801...	—	0x00000280
轴1	H2E.07	TPD02的传输类型 (0x18...	—	255
轴1	H2E.08	TPD02的禁止时间 (0x18...	—	0x0064
轴1	H2E.09	TPD02的事件计时器 (0x...	—	0x0000
轴1	H2E.10	TPD03的C08-ID (0x1802...	—	0x00000380
轴1	H2E.12	TPD03的传输类型 (0x18...	—	255
轴1	H2E.13	TPD03的禁止时间 (0x18...	—	0x0064
轴1	H2E.14	TPD03的事件计时器 (0x...	—	0x0000
轴1	H2E.15	TPD04的C08-ID (0x1803...	—	0x00000480
轴1	H2E.17	TPD04的传输类型 (0x18...	—	255
轴1	H2E.18	TPD04的禁止时间 (0x18...	—	0x0064
轴1	H2E.19	TPD04的事件计时器 (0x...	—	0x0000
轴1	H2E.20	TPD01有效映射对象个...	—	2
轴1	H2E.21	TPD01映射对象1 (0x1A0...	—	0x60410010
轴1	H2E.23	TPD01映射对象2 (0x1A0...	—	0x606C0020
轴1	H2E.25	TPD01映射对象3 (0x1A0...	—	0x00000000
轴1	H2E.27	TPD01映射对象4 (0x1A0...	—	0x00000000
轴1	H2E.29	TPD01映射对象5 (0x1A0...	—	0x00000000
轴1	H2E.31	TPD01映射对象6 (0x1A0...	—	0x00000000
轴1	H2E.33	TPD01映射对象7 (0x1A0...	—	0x00000000
轴1	H2E.35	TPD01映射对象8 (0x1A0...	—	0x00000000
轴1	H2E.37	TPD02有效映射对象个...	—	2
轴1	H2E.38	TPD02映射对象1 (0x1A0...	—	0x60410010
轴1	H2E.40	TPD02映射对象2 (0x1A0...	—	0x60610008
轴1	H2E.42	TPD02映射对象3 (0x1A0...	—	0x00000000

5.5.3 配置网关

- 1、使用 USB 线连接网关配置口和电脑，给网关上电。
- 2、选择并打开配置串口，如果没有找到 USB 线连接网关的串口请安装 CH340 驱动。



3、在软件上填写网关参数，添加一个 CANOPEN 从站并在从站下添加一个 download SDO、一个 upload SDO、一个 TPDO、一个 RPDO。如下图所示：



配置网关			COE从站状态映射表	PDO映射表	SDO映射表	配置CANOPEN从站
网关参数						
1						
Download SDO						
Upload SDO						
RPDO						
TPDO						
			参数	值/选项		
			COB-ID (HEX)	181		
			映射参数1 (HEX)	60410010		
			映射参数2 (HEX)	606C0020		
			映射参数3 (HEX)	00000000		
			映射参数4 (HEX)	00000000		

4、配置网关，弹出网关配置成功。

在“CANOPEN 从站状态映射表”、“PDO 映射表”、“SDO 映射参数”就可以看到数据对应 ECAT 端 Input Bytes[] Output Bytes[]的位置。

配置网关				COE从站状态映射表	PDO映射表	SDO映射表	配置CANOPEN从站
				参数	输入映射地址/字节长度		输出映射地址/字节长度
				COE 从站 1 状态	0 / 1		
				*			

配置网关

COE从站状态映射表

PDO映射表

SDO映射表

配置CANOPEN从站

	所属从站	PDO类型	COB-ID	映射参数(HEX)	输入映射地址/字节长度	输出映射地址/字节长度
▶	1	RPDO	201	60400010		0 / 2
	1	RPDO	201	60FF0020		2 / 4
	1	TPDO	181	60410010	1 / 2	
	1	TPDO	181	606C0020	3 / 4	

配置网关								COE从站状态映射表	PDO映射表	SDO映射表	配置CANOPEN从站
	所属从站	SDO类型	索引(HEX)	子索引(HEX)	字节长度	输入映射地址/字节长度	输出映射地址				
▶	1	Download SDO	6060	00	1		6 / 1				
	1	Upload SDO	6061	00	1	7 / 1					
•											

即数据映射表如下：

从站 1 状态	Input Bytes[0]
RPDO 0x201 映射的 0x6040 对象	Output Bytes[0..1]
RPDO 0x201 映射的 0x60FF 对象	Output Bytes[2..5]
TPDO 0x101 映射的 0x6041 对象	Input Bytes[1..2]
TPDO 0x101 映射的 0x606C 对象	Input Bytes[3..6]
Download SDO 0x6060 对象	Output Bytes[6]
Upload SDO 0x6061 对象	Input Bytes[7]

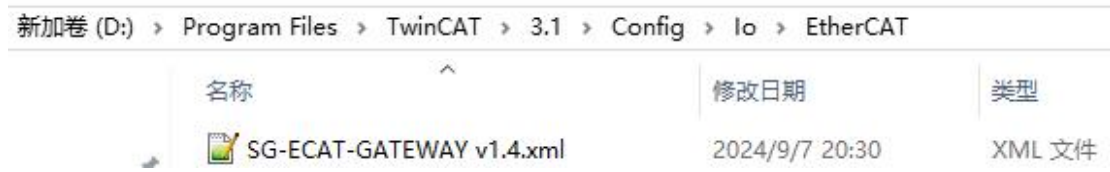
5.2 TwinCAT 下实例

PS: TwinCAT 对网卡型号有要求，要确保所使用的网卡支持 TwinCAT。

先按 5.1 设置伺服驱动器和网关。

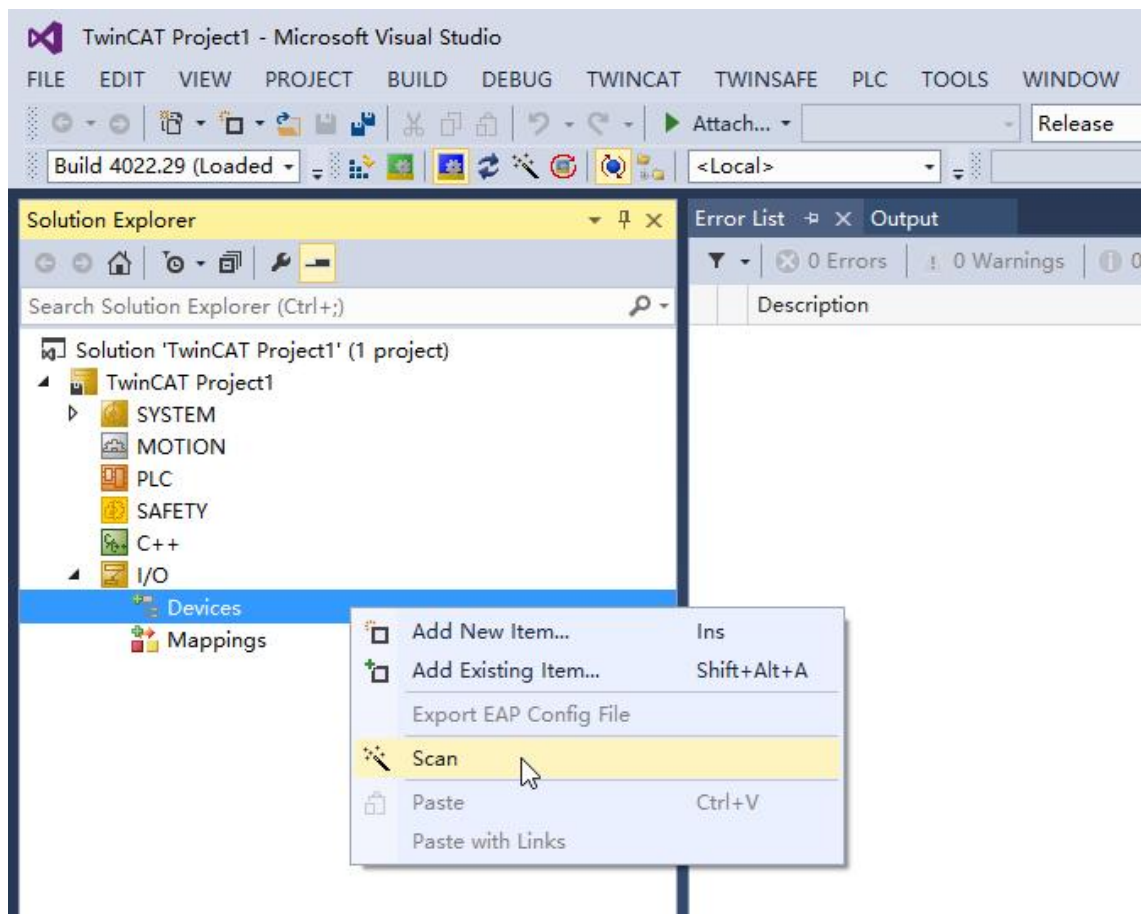
①去我司网站下载网关的 ESI 文件。

复制 ESI 文件到 TwinCAT 安装目录的 EtherCAT 目录下，如：D:\Program Files\TwinCAT 3\3.1\Config\Io\EtherCAT。

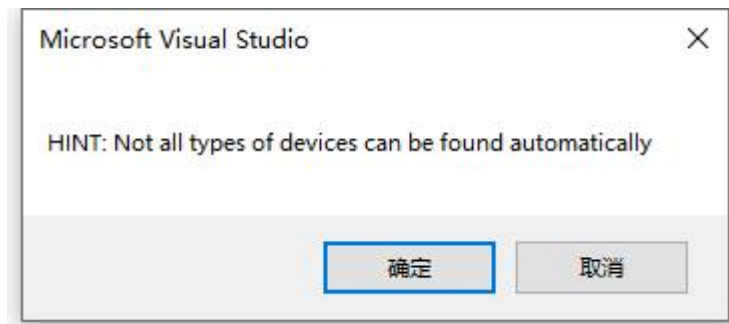


②连接电脑和网关，注意电脑连接网关的 Port1（ECAT 两个网口区分输入输出，别接反了）。

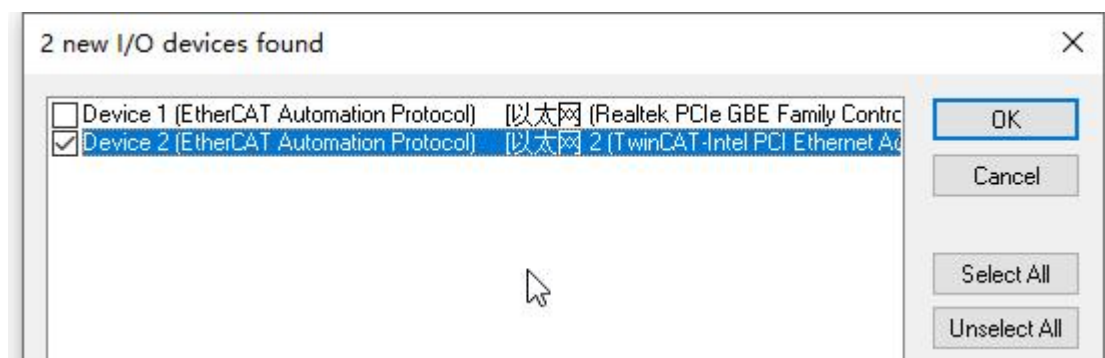
建立 TwinCAT 工程，右键 Device 并点 Scan。



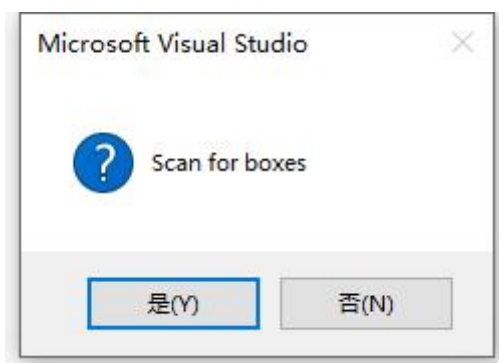
在下面这个对话框选择“确定”



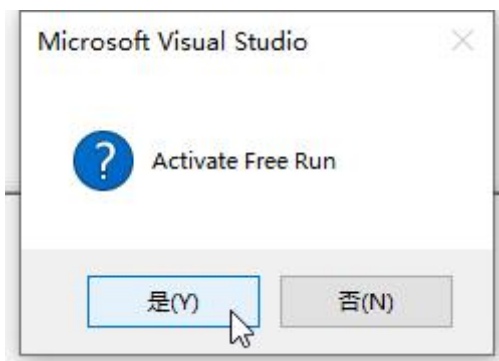
在下面这个对话框选择 TwinCAT 使用的网卡并选择“OK”。



在下面这个对话框选择“是”

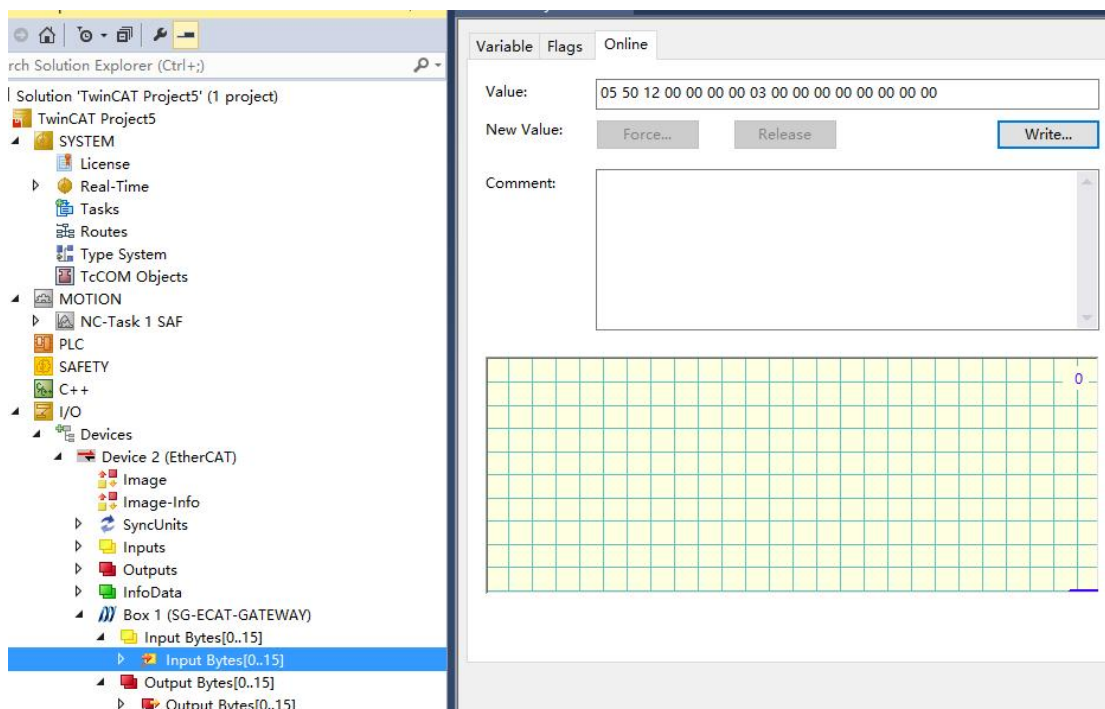


在下面这个对话框选择“是”

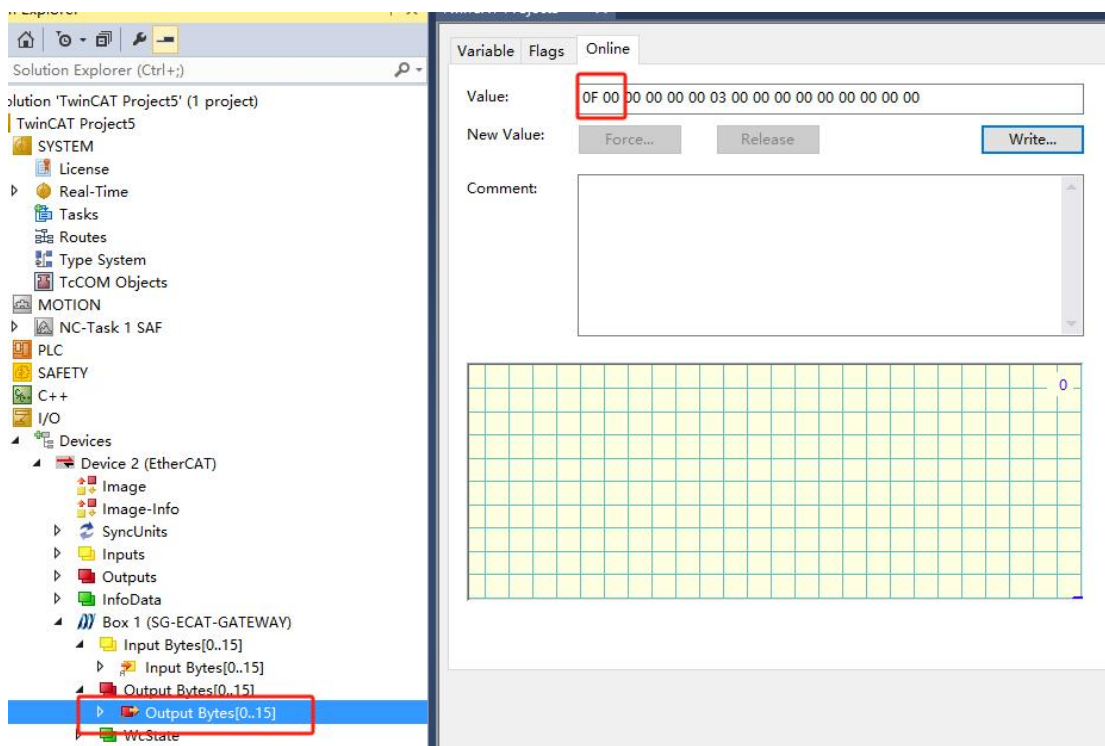


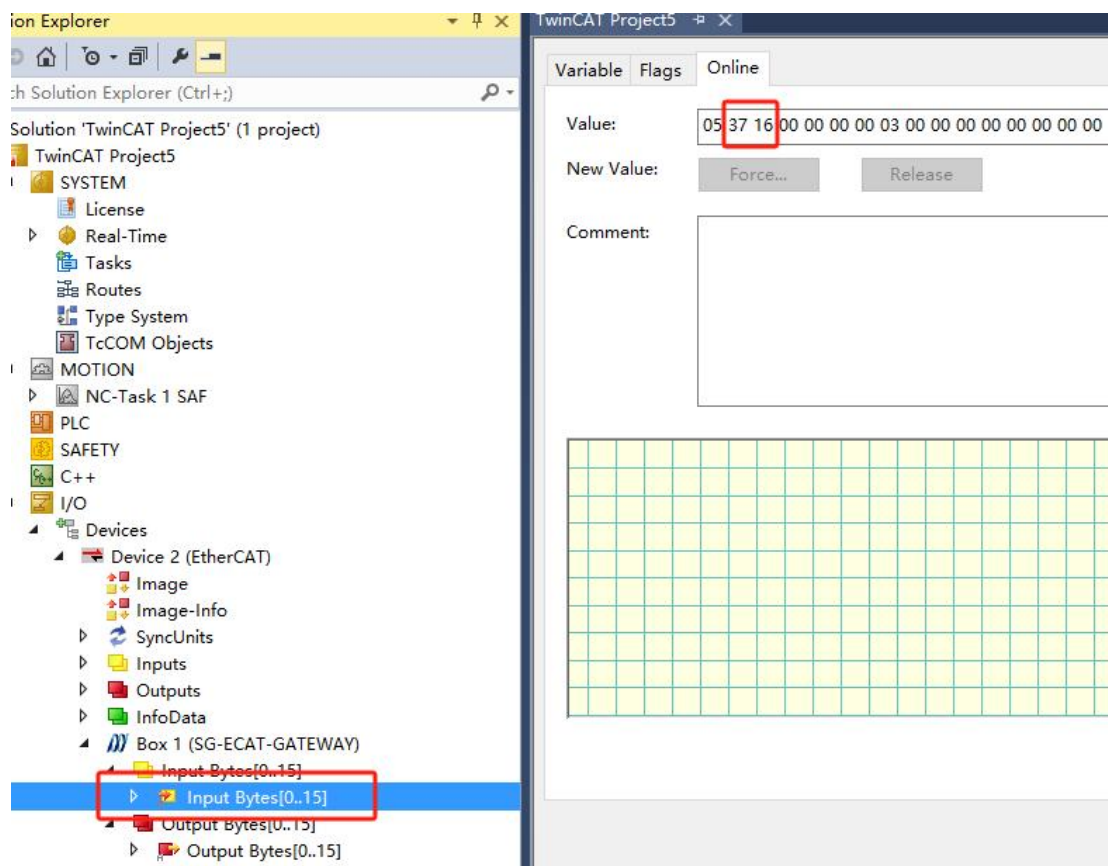
③在 Input Bytes、Output Bytes 可以查看、写入数据

首先设置模式为轮廓速度模式，在 Output Bytes[6]写 3，之后 Input Bytes[7]显示 3。

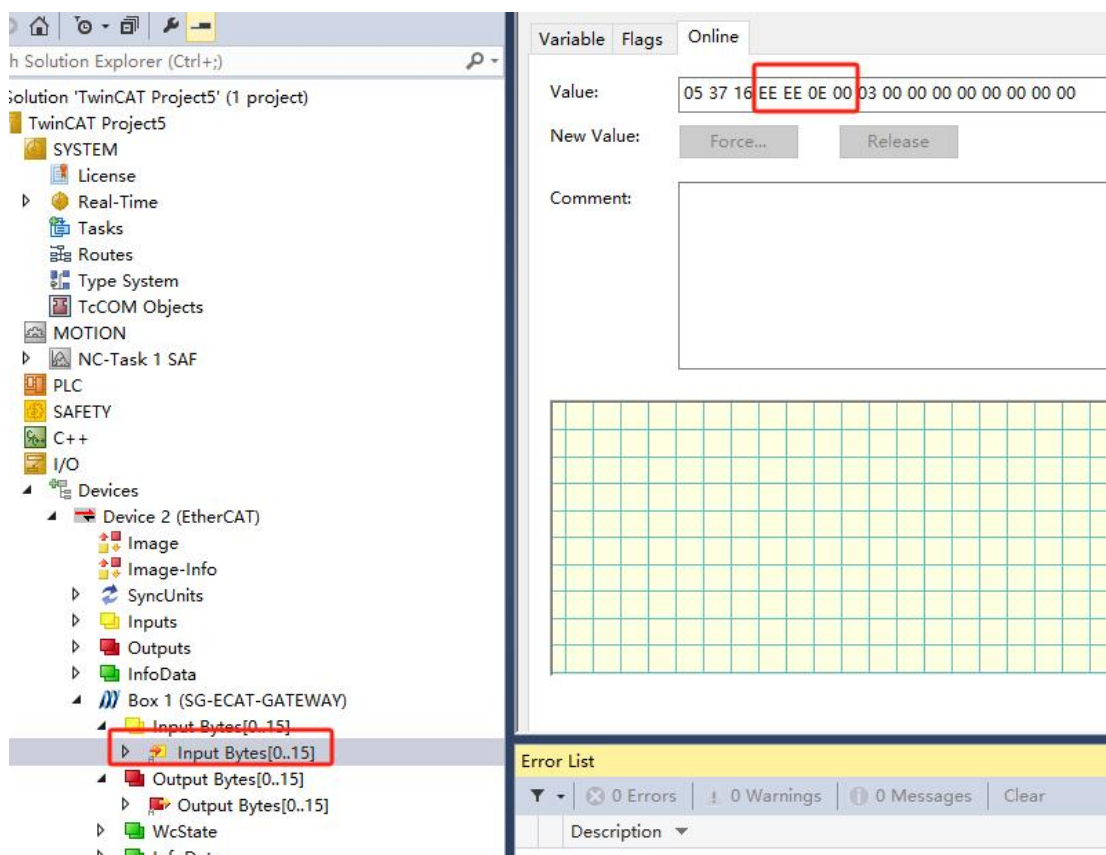
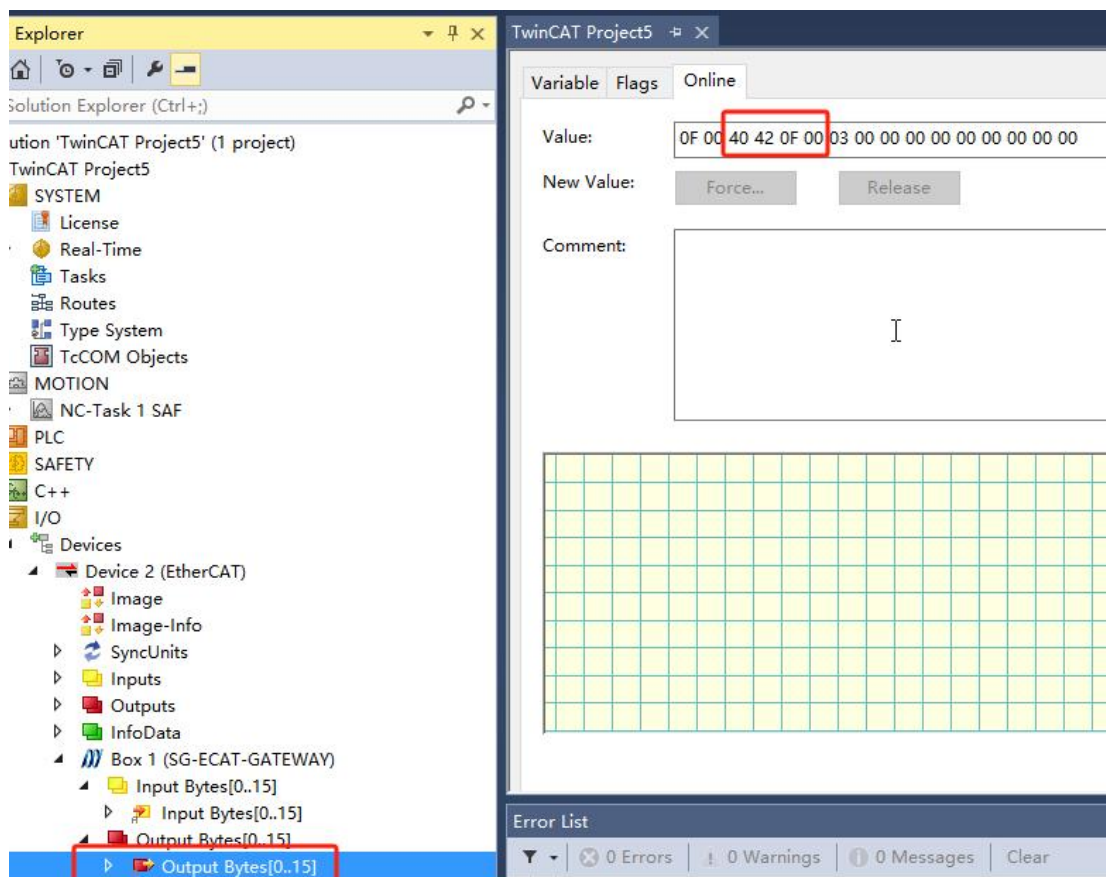


之后按 CIA402 规约引导驱动器运行，给控制字依次写入 6、7、F。注意，CANOPEN 是小端的，所以给控制字 Output Bytes[0..1]写 6、7、F 应该是 Bytes[0]写 6、7、F





之后给目标速度一个值，比如 1000000(0x000F4240)，注意 CANOPEN 是小端的，所以给控制字 Output Bytes[2..5]写 0x000F4240 是 Bytes[2..5]为 0x40、0xF2、0x0F、0x00。



之后电机缓慢转起来，位置反馈 Input Bytes[3..6]显示实际速度，实际速度和

设置有很小偏差。

5.3 CODESYS 例程

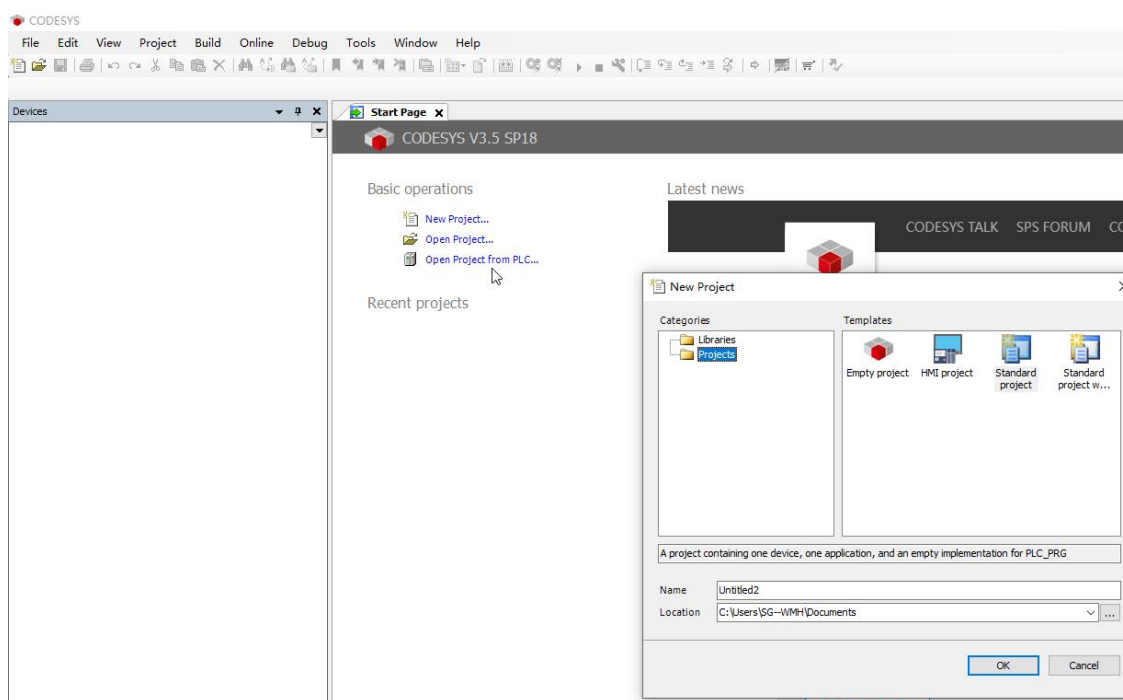
PS: CODESYS 对网卡型号有要求，要确保所使用的网卡支持 CODESYS。

先按 5.1 设置伺服驱动器和网关。

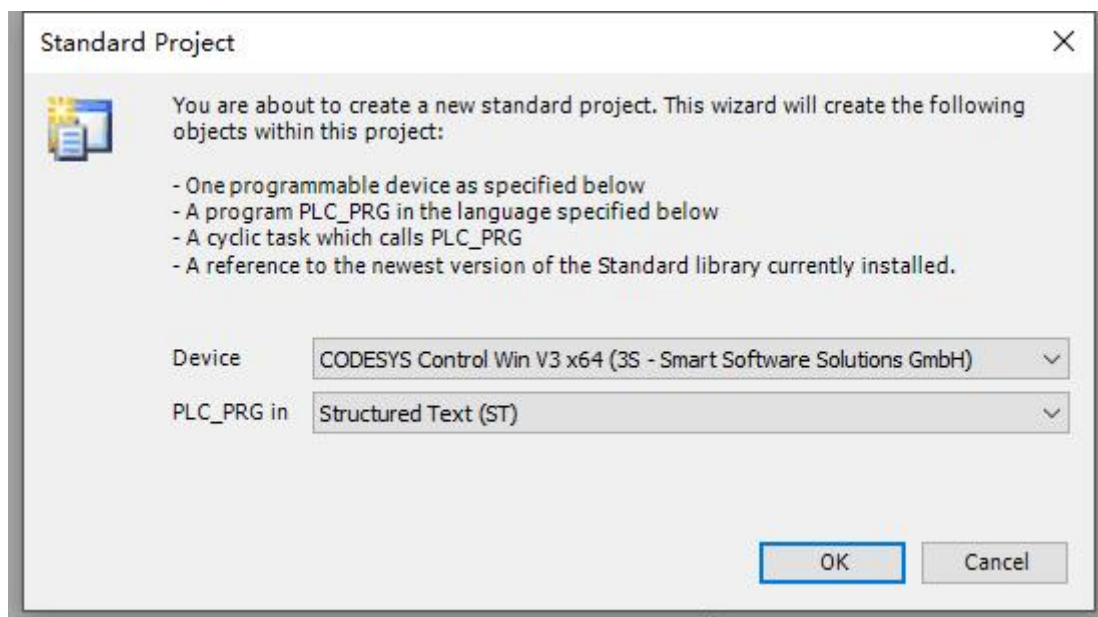
①去我司网站下载设备的 ESI 文件。

连接电脑和网关，注意电脑连接网关的 Port1（ECAT 两个网口区分输入输出，别接反了）。

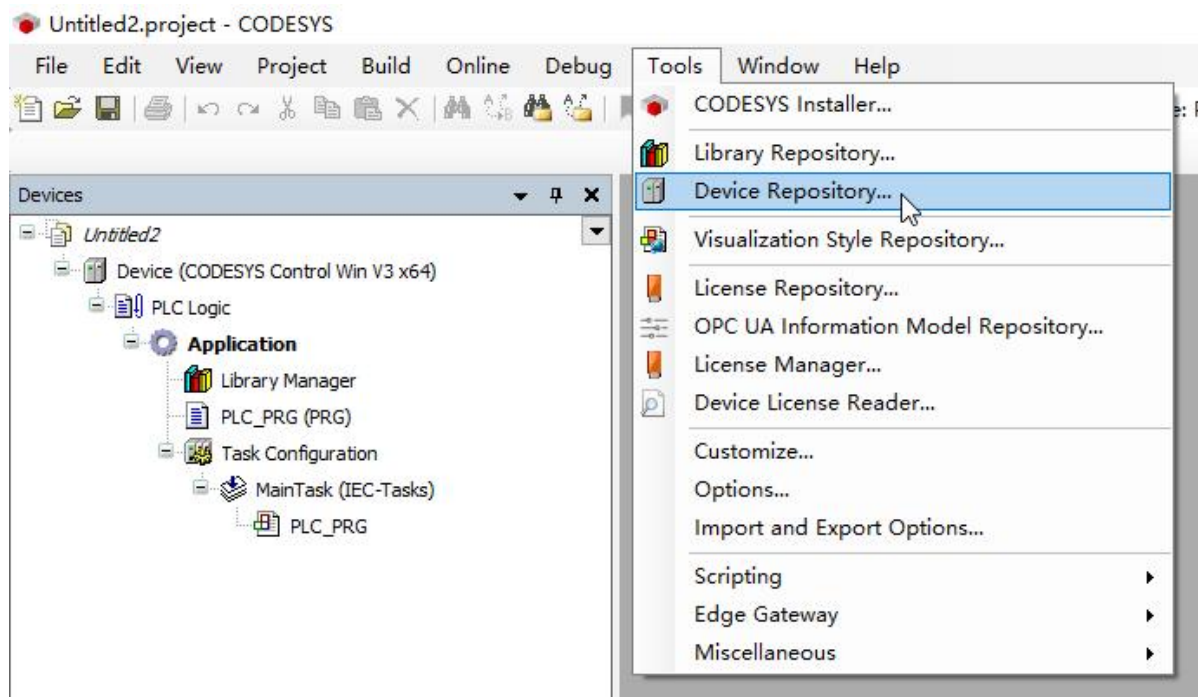
②打开 CODESYS，创建工程

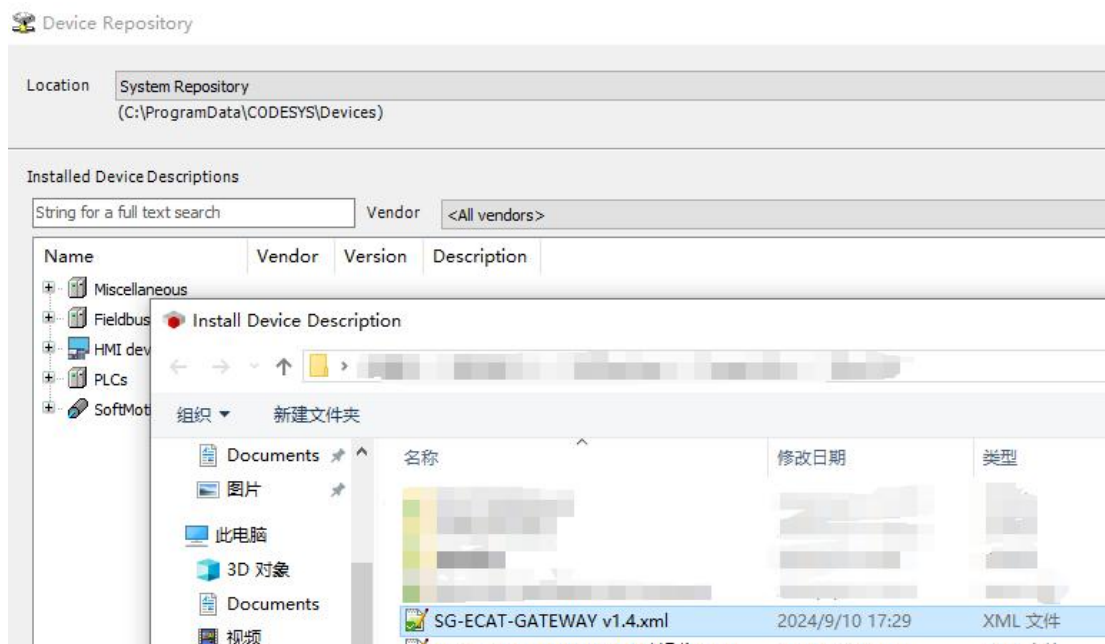


在下面这个对话框选择“OK”

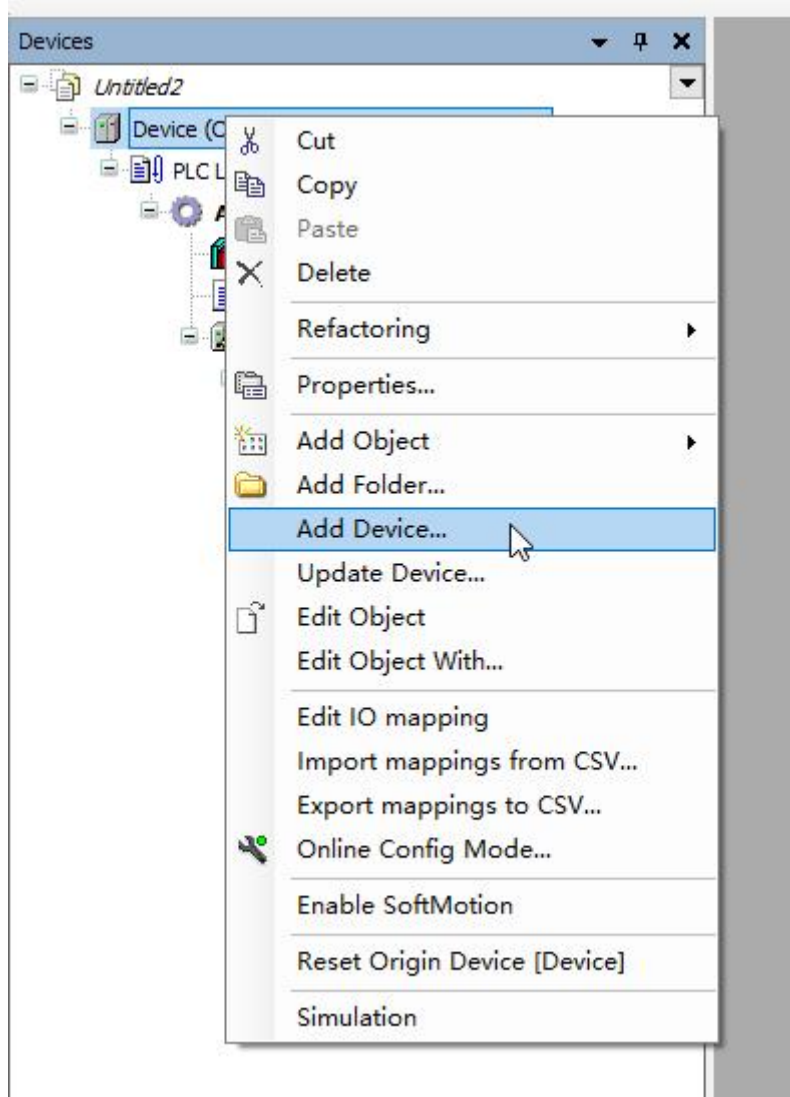


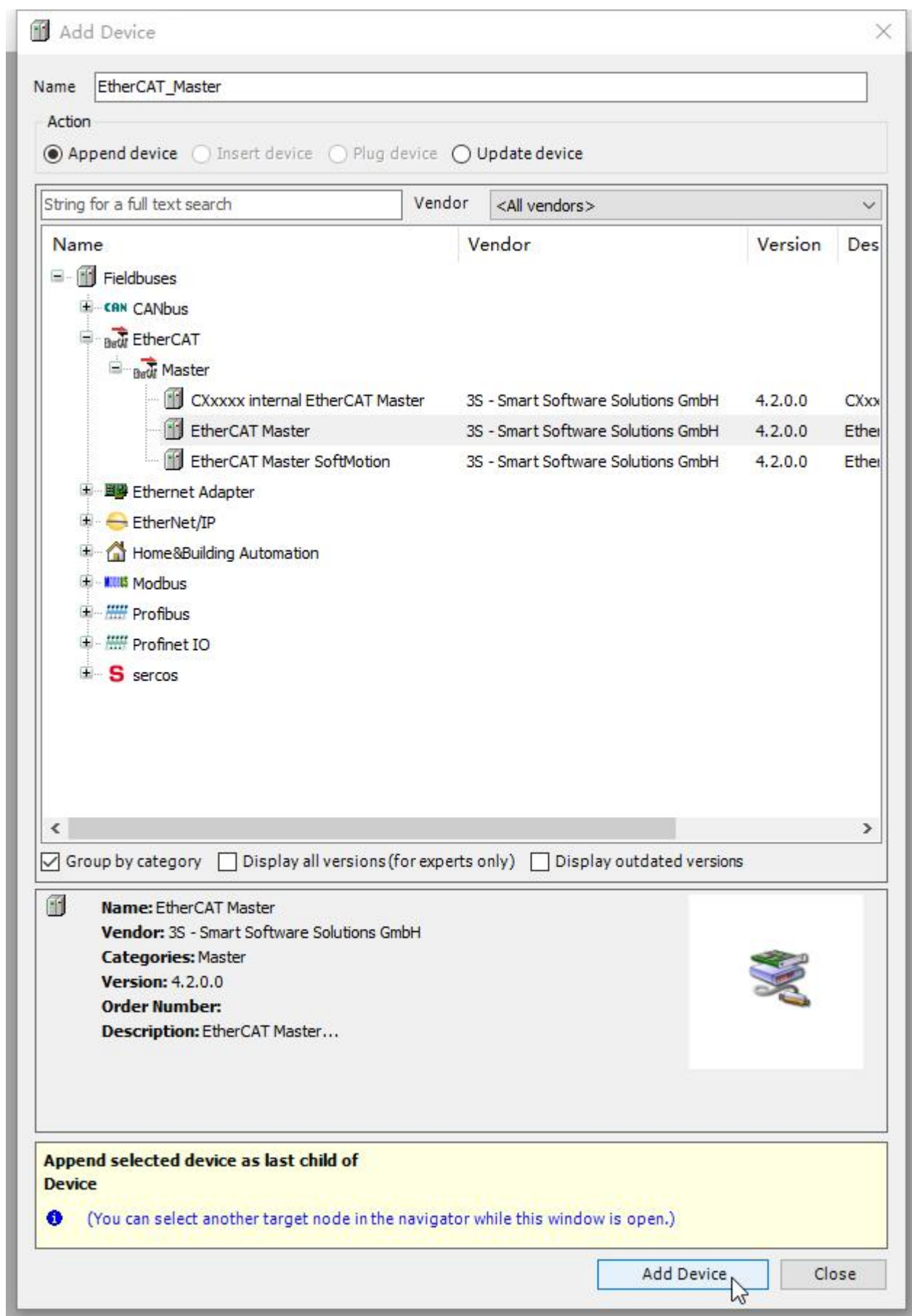
安装 ESI 文件，点击 “Tools” ->Device Repository,点击 “Install” 选择 ESI 文件并安装。





③工程目录右击“Device”选择“Add Device”,添加一个 EtherCAT 主站。



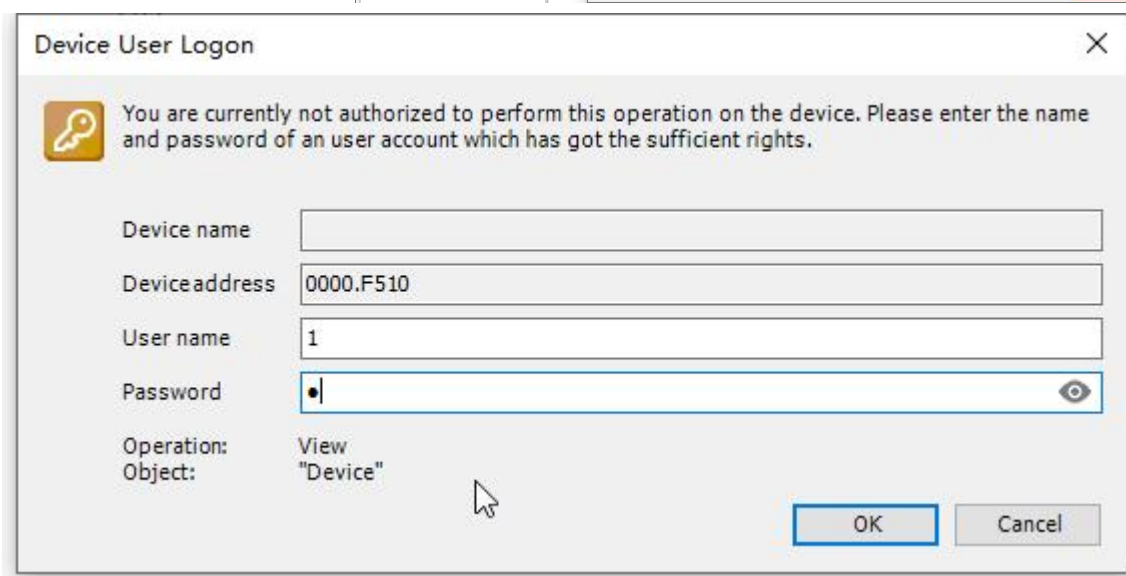
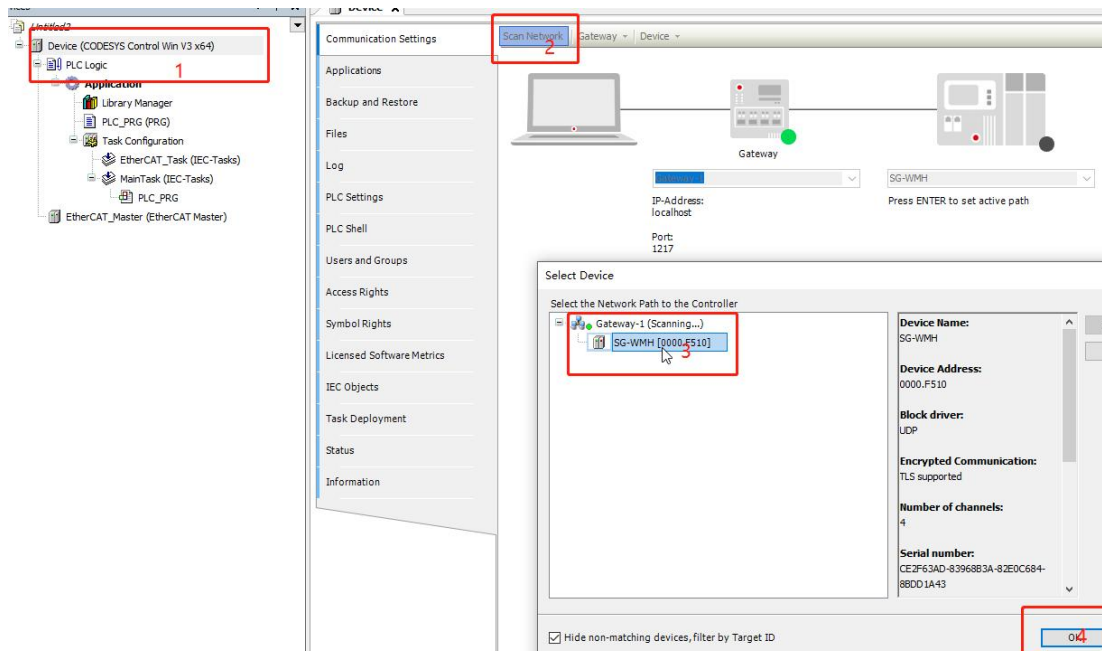


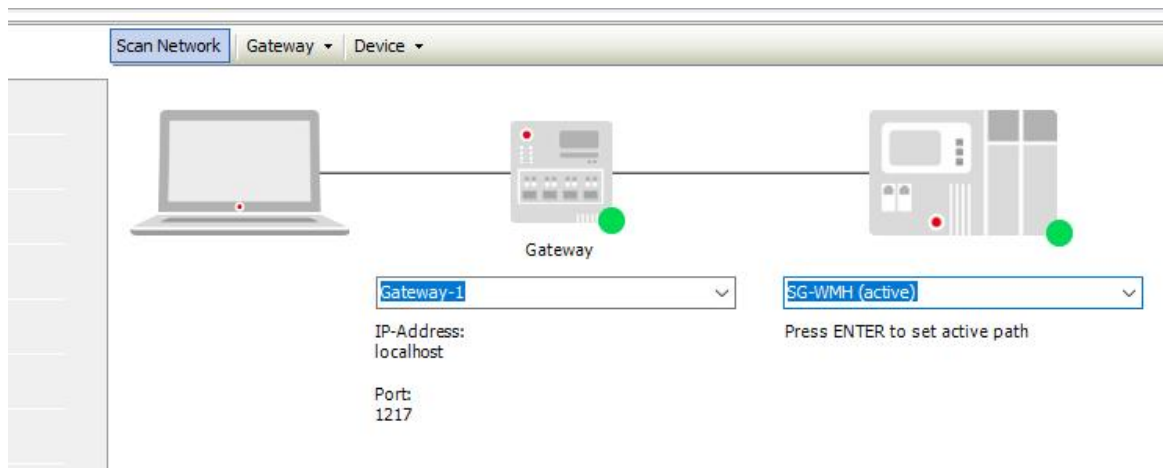
④启动 CODESYS 自带的 Gateway 和 PLC

确保任务栏的这两个图标处于运行状态。



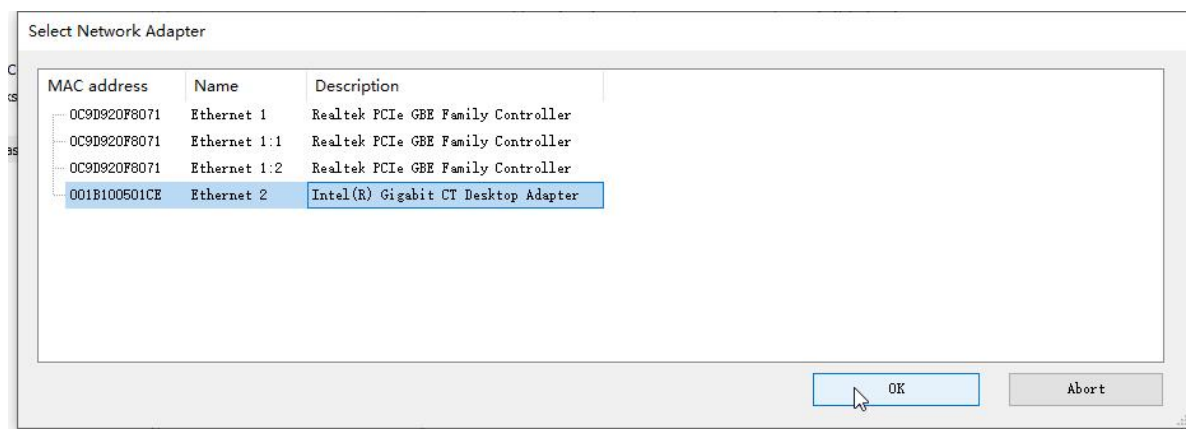
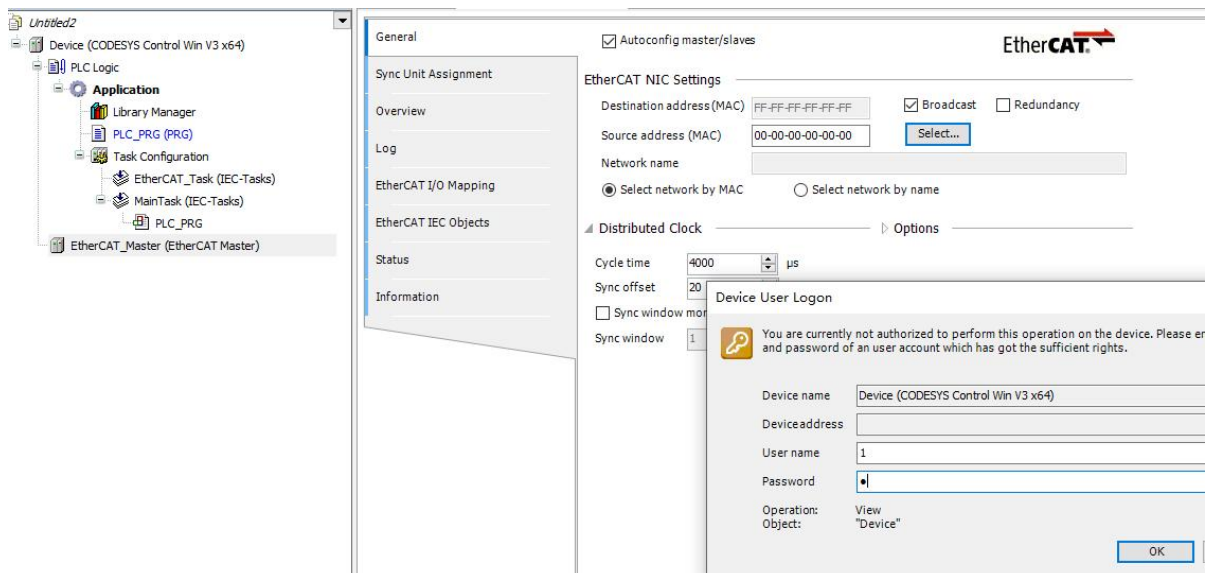
右击“Device”连接网关和 PLC，如下图所示



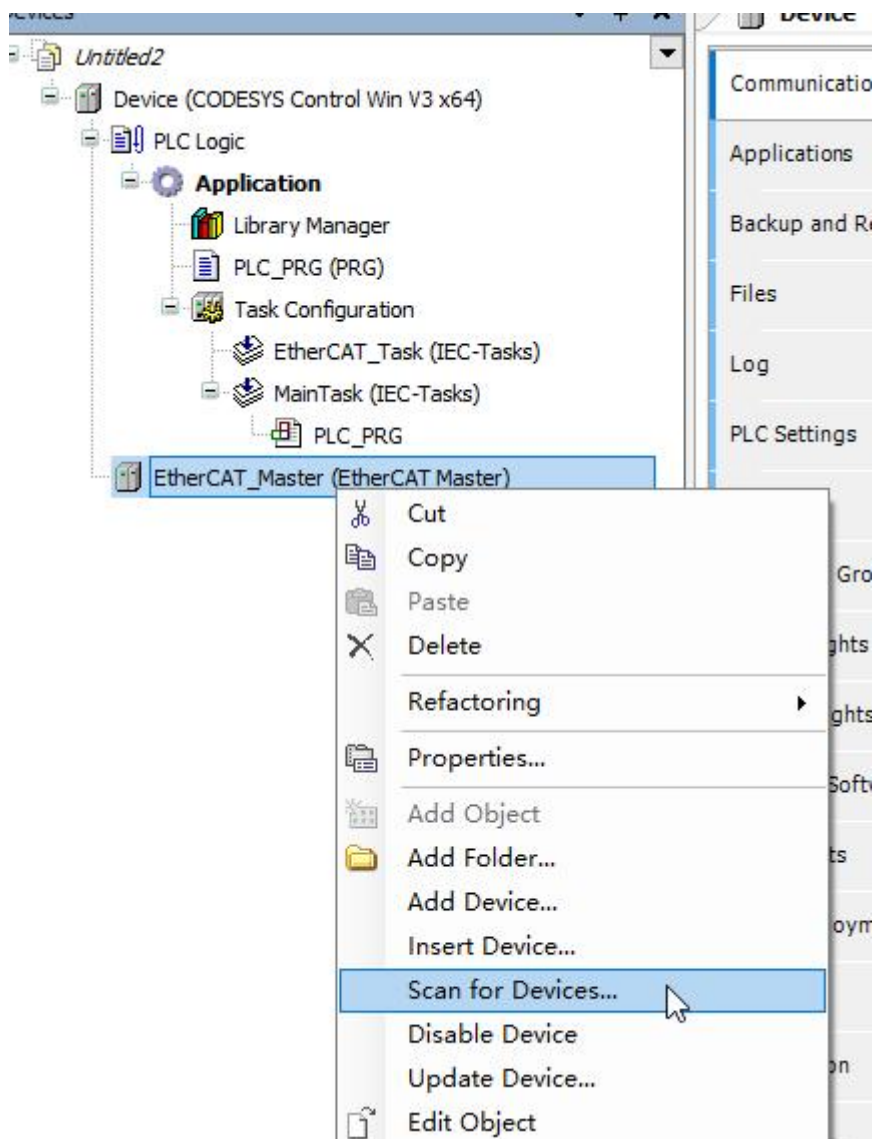


连接成功如上图所示。

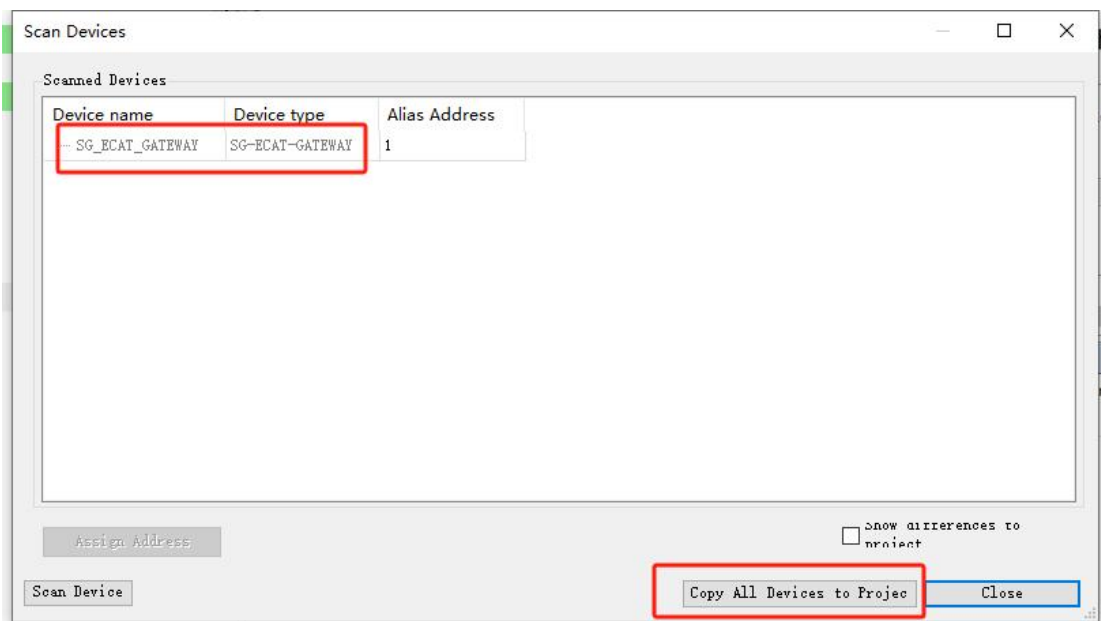
⑤给 EtherCAT 主站选择网卡，双击 EtherCAT 主站，在 General 页面选择网卡。



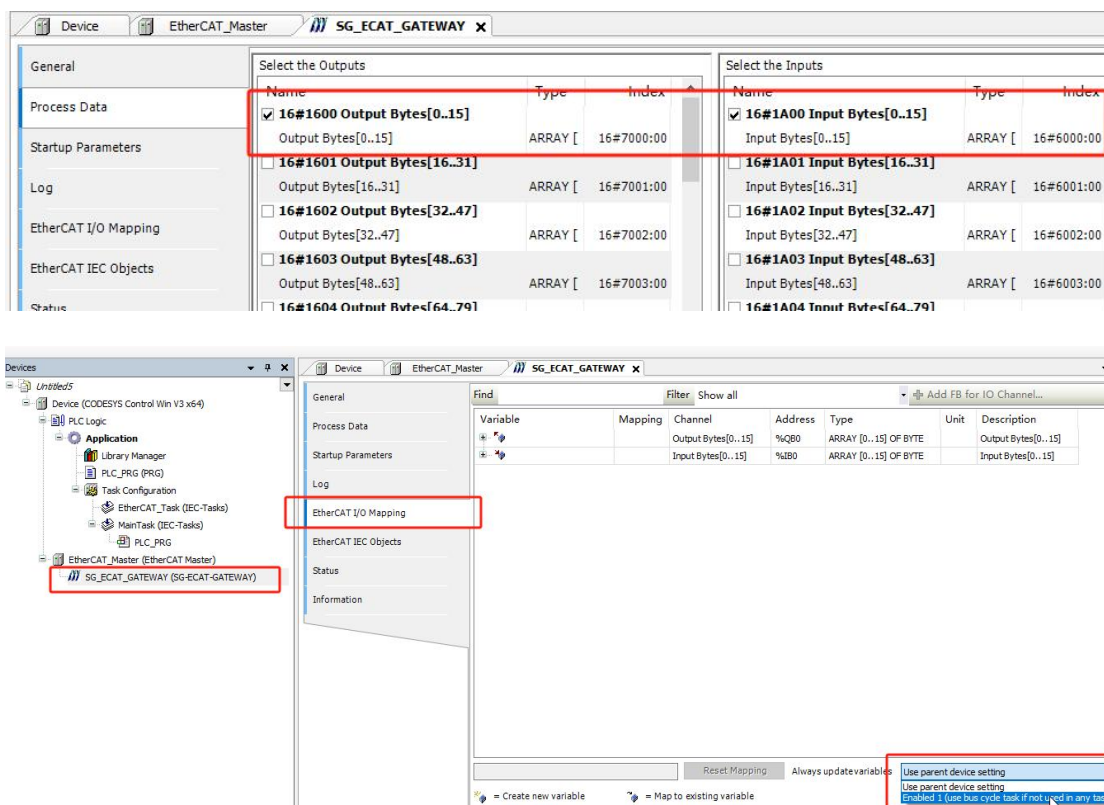
⑥右击工程目录主站点击“Scan for Devices”



弹出如下页面



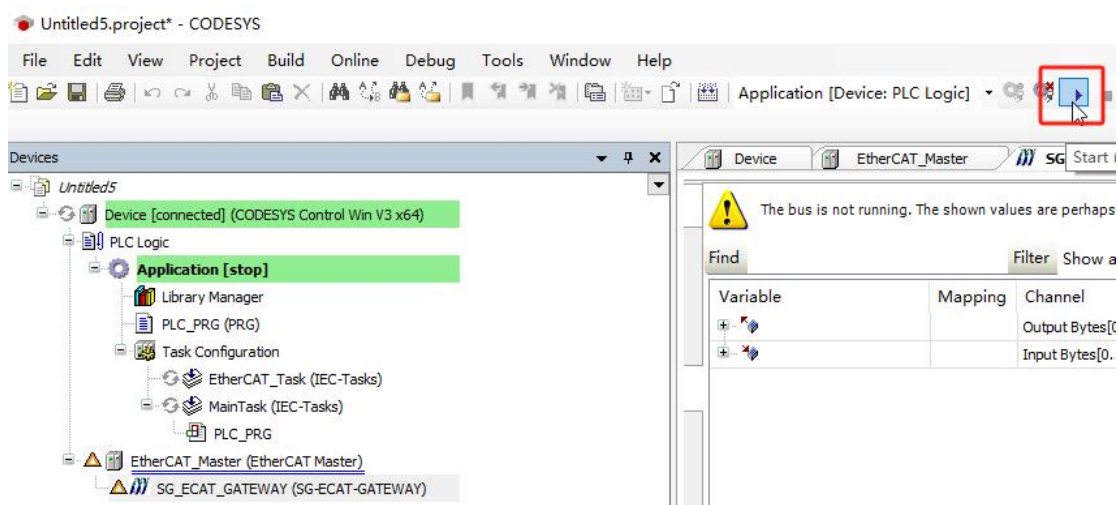
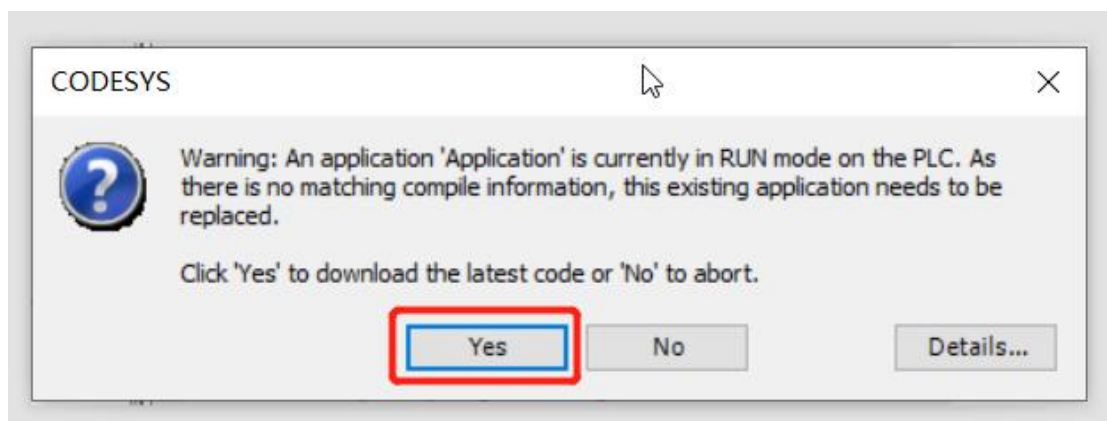
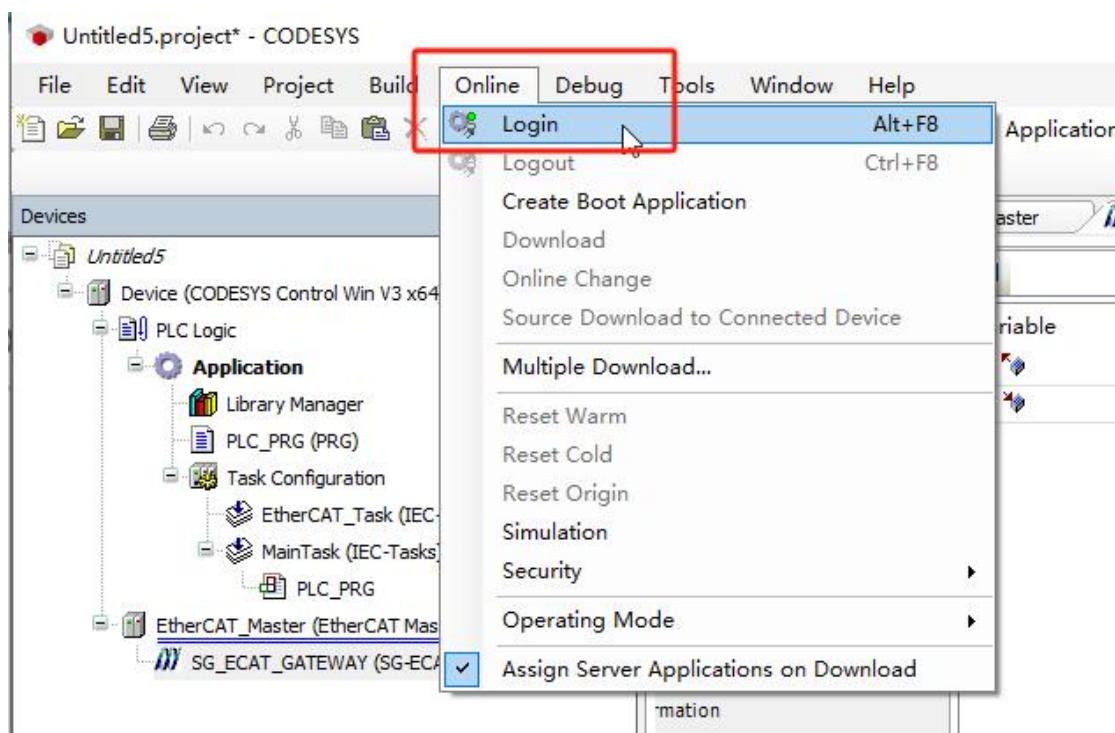
⑦在 Process Data 选择 PDO



编译一下



⑧登录、下载、运行



⑨在 EtherCAT IO Mapping 查看写入数据

首先设置模式为轮廓速度模式，在 Output Bytes[6]写 3，之后 Input Bytes[7]显示 3。

Variable	Mapping	Channel	Address	Type	Current Value	Prepared Value	Unit	Description
Output Bytes[0..15]			%QB0	ARRAY [0..15] OF BYTE	Only subelements up...			Output Bytes[0..15]
Output Bytes[0..15][0]			%QB0	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][1]			%QB1	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][2]			%QB2	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][3]			%QB3	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][4]			%QB4	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][5]			%QB5	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][6]			%QB6	BYTE	16#03			Output Bytes[0..15]
Output Bytes[0..15][7]			%QB7	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][8]			%QB8	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][9]			%QB9	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][10]			%QB10	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][11]			%QB11	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][12]			%QB12	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][13]			%QB13	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][14]			%QB14	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][15]			%QB15	BYTE	16#00			Output Bytes[0..15]
Input Bytes[0..15]			%IB0	ARRAY [0..15] OF BYTE	Only subelements up...			Input Bytes[0..15]
Input Bytes[0..15][0]			%IB0	BYTE	16#05			Input Bytes[0..15]
Input Bytes[0..15][1]			%IB1	BYTE	16#50			Input Bytes[0..15]
Input Bytes[0..15][2]			%IB2	BYTE	16#12			Input Bytes[0..15]
Input Bytes[0..15][3]			%IB3	BYTE	16#00			Input Bytes[0..15]
Input Bytes[0..15][4]			%IB4	BYTE	16#00			Input Bytes[0..15]
Input Bytes[0..15][5]			%IB5	BYTE	16#00			Input Bytes[0..15]
Input Bytes[0..15][6]			%IB6	BYTE	16#00			Input Bytes[0..15]
Input Bytes[0..15][7]			%IB7	BYTE	16#03			Input Bytes[0..15]
Input Bytes[0..15][8]			%IB8	BYTE	16#00			Input Bytes[0..15]

之后按 CIA402 规约引导驱动器运行，给控制字依次写入 6、7、F。注意，CANOPEN 是小端的，所以给控制字 Output Bytes[0..1]写 6、7、F 应该是 Bytes[0]写 6、7、F

Variable	Mapping	Channel	Address	Type	Current Value	Prepared Value	Unit	Description
Output Bytes[0..15]			%QB0	ARRAY [0..15] OF BYTE	Only subelements up...			Output Bytes[0..15]
Output Bytes[0..15][0]			%QB0	BYTE	16#0F			Output Bytes[0..15]
Output Bytes[0..15][1]			%QB1	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][2]			%QB2	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][3]			%QB3	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][4]			%QB4	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][5]			%QB5	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][6]			%QB6	BYTE	16#03			Output Bytes[0..15]
Output Bytes[0..15][7]			%QB7	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][8]			%QB8	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][9]			%QB9	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][10]			%QB10	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][11]			%QB11	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][12]			%QB12	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][13]			%QB13	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][14]			%QB14	BYTE	16#00			Output Bytes[0..15]
Output Bytes[0..15][15]			%QB15	BYTE	16#00			Output Bytes[0..15]
Input Bytes[0..15]			%IB0	ARRAY [0..15] OF BYTE	Only subelements up...			Input Bytes[0..15]
Input Bytes[0..15][0]			%IB0	BYTE	16#05			Input Bytes[0..15]
Input Bytes[0..15][1]			%IB1	BYTE	16#37			Input Bytes[0..15]
Input Bytes[0..15][2]			%IB2	BYTE	16#16			Input Bytes[0..15]
Input Bytes[0..15][3]			%IB3	BYTE	16#00			Input Bytes[0..15]
Input Bytes[0..15][4]			%IB4	BYTE	16#00			Input Bytes[0..15]
Input Bytes[0..15][5]			%IB5	BYTE	16#00			Input Bytes[0..15]
Input Bytes[0..15][6]			%IB6	BYTE	16#00			Input Bytes[0..15]
Input Bytes[0..15][7]			%IB7	BYTE	16#03			Input Bytes[0..15]

之后给目标速度一个值，比如 1000000(0x000F4240)，注意 CANOPEN 是小端的，所以给控制字 Output Bytes[2..5]写 0x000F4240 是 Bytes[2..5]为 0x40、0xF2、0x0F、0x00。

Variable	Mapping	Channel	Address	Type	Current Value	Prepared Value	Unit	Description
		Output Bytes[0..15]	%QB0	ARRAY [0..15] OF BYTE	Only subelements up...			Output Bytes[0..15]
		Output Bytes[0..15][0]	%QB0	BYTE	16#0F			Output Bytes[0..15]
		Output Bytes[0..15][1]	%QB1	BYTE	16#00			Output Bytes[0..15]
		Output Bytes[0..15][2]	%QB2	BYTE	16#40			Output Bytes[0..15]
		Output Bytes[0..15][3]	%QB3	BYTE	16#42			Output Bytes[0..15]
		Output Bytes[0..15][4]	%QB4	BYTE	16#0F			Output Bytes[0..15]
		Output Bytes[0..15][5]	%QB5	BYTE	16#00			Output Bytes[0..15]
		Output Bytes[0..15][6]	%QB6	BYTE	16#03			Output Bytes[0..15]
		Output Bytes[0..15][7]	%QB7	BYTE	16#00			Output Bytes[0..15]
		Output Bytes[0..15][8]	%QB8	BYTE	16#00			Output Bytes[0..15]
		Output Bytes[0..15][9]	%QB9	BYTE	16#00			Output Bytes[0..15]
		Output Bytes[0..15][10]	%QB10	BYTE	16#00			Output Bytes[0..15]
		Output Bytes[0..15][11]	%QB11	BYTE	16#00			Output Bytes[0..15]
		Output Bytes[0..15][12]	%QB12	BYTE	16#00			Output Bytes[0..15]
		Output Bytes[0..15][13]	%QB13	BYTE	16#00			Output Bytes[0..15]
		Output Bytes[0..15][14]	%QB14	BYTE	16#00			Output Bytes[0..15]
		Output Bytes[0..15][15]	%QB15	BYTE	16#00			Output Bytes[0..15]
		Input Bytes[0..15]	%IB0	ARRAY [0..15] OF BYTE	Only subelements up...			Input Bytes[0..15]
		Input Bytes[0..15][0]	%IB0	BYTE	16#05			Input Bytes[0..15]
		Input Bytes[0..15][1]	%IB1	BYTE	16#37			Input Bytes[0..15]
		Input Bytes[0..15][2]	%IB2	BYTE	16#16			Input Bytes[0..15]
		Input Bytes[0..15][3]	%IB3	BYTE	16#EE			Input Bytes[0..15]
		Input Bytes[0..15][4]	%IB4	BYTE	16#EE			Input Bytes[0..15]
		Input Bytes[0..15][5]	%IB5	BYTE	16#0E			Input Bytes[0..15]
		Input Bytes[0..15][6]	%IB6	BYTE	16#00			Input Bytes[0..15]
		Input Bytes[0..15][7]	%IB7	BYTE	16#03			Input Bytes[0..15]
		Input Bytes[0..15][8]	%IB8	BYTE	16#00			Input Bytes[0..15]

之后电机缓慢转起来，位置反馈 Input Bytes[3..6]显示实际速度，实际速度和设置有很小偏差。

5.4 欧姆龙 Sysmac Studio 例程

先按 5.1 设置伺服驱动器和网关。

①首先去我司网站下载设备的 ESI 文件。

连接电脑、欧姆龙 PLC 和网关，注意 PLC 的 EtherCAT 口接网关的 Port1（ECAT 两个网口区分输入输出，别接反了）。

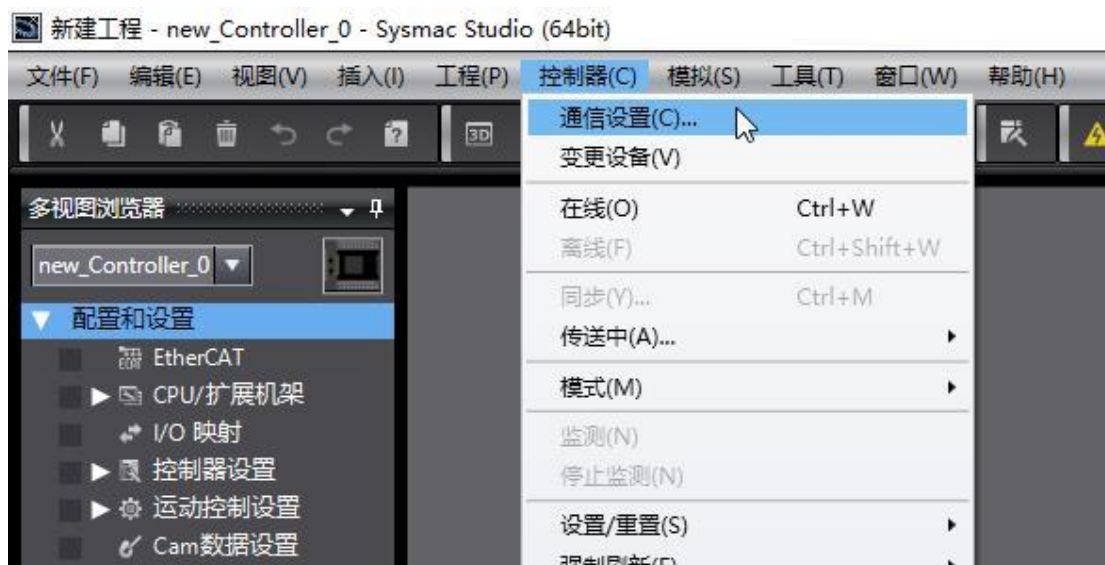
②打开 Sysmac Studio，创建工程



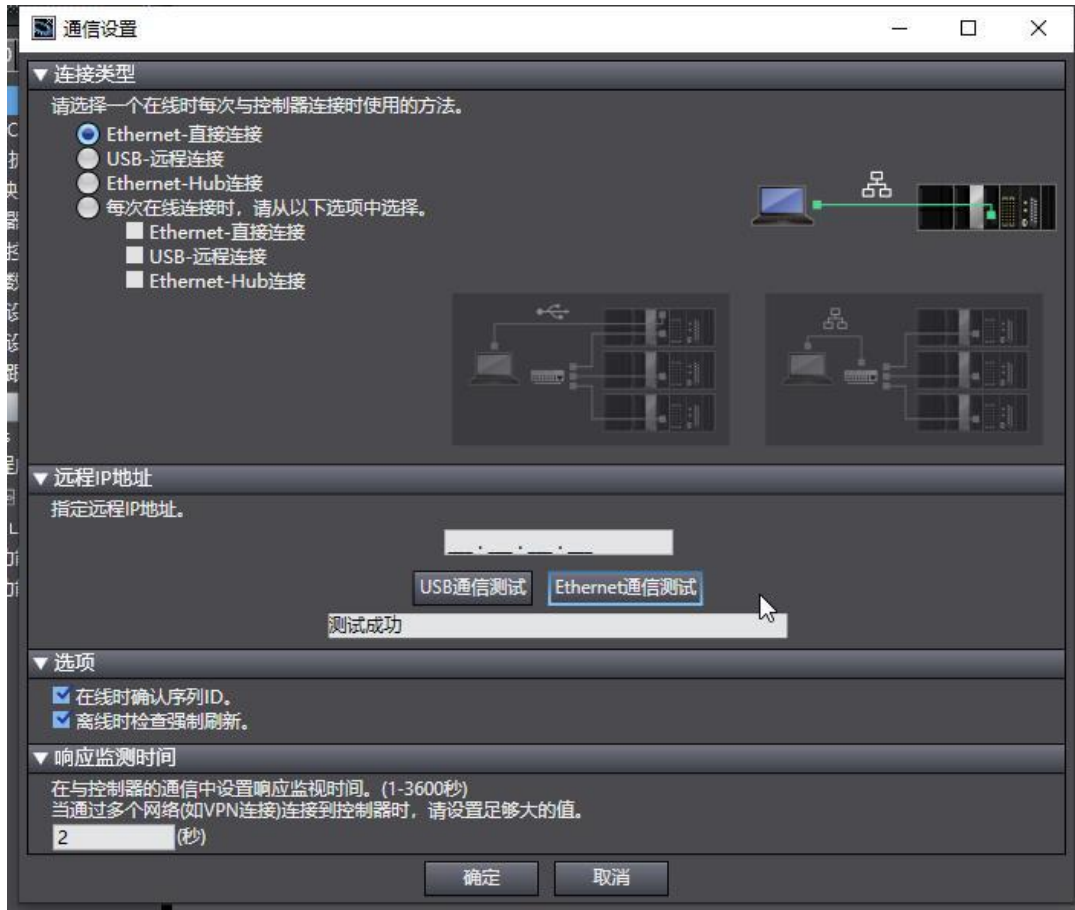
根据实际 PLC 型号和版本创建工程



通信测试，点击“控制器”->“通信设置”

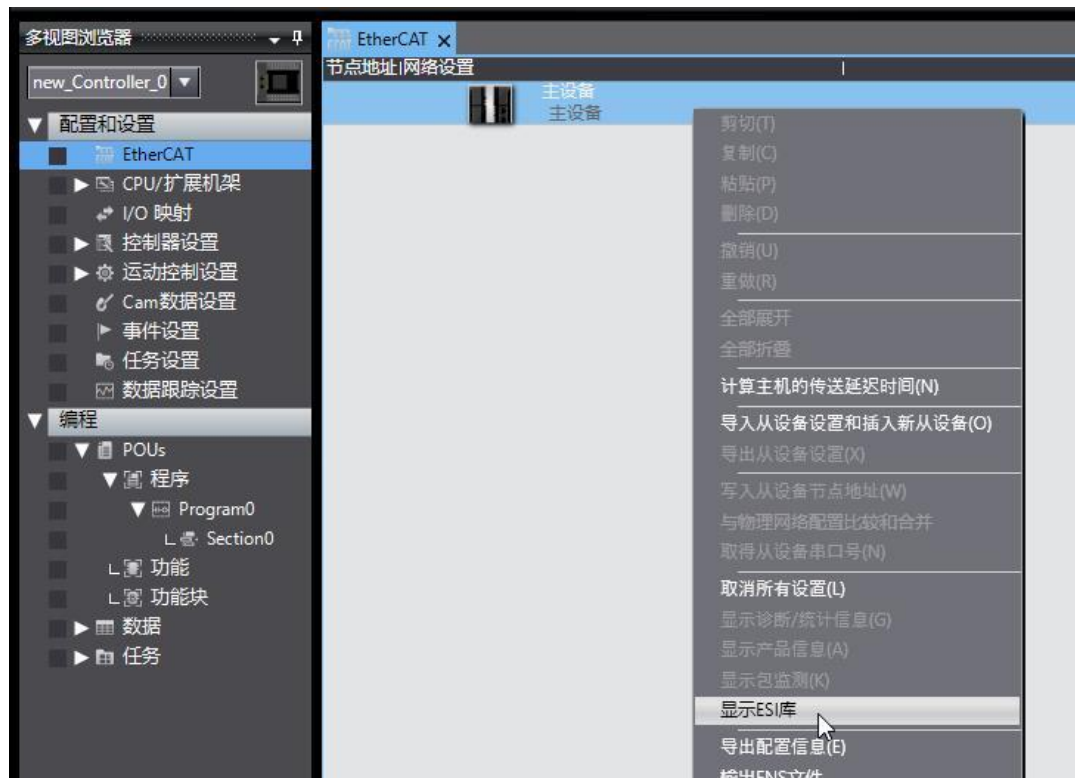


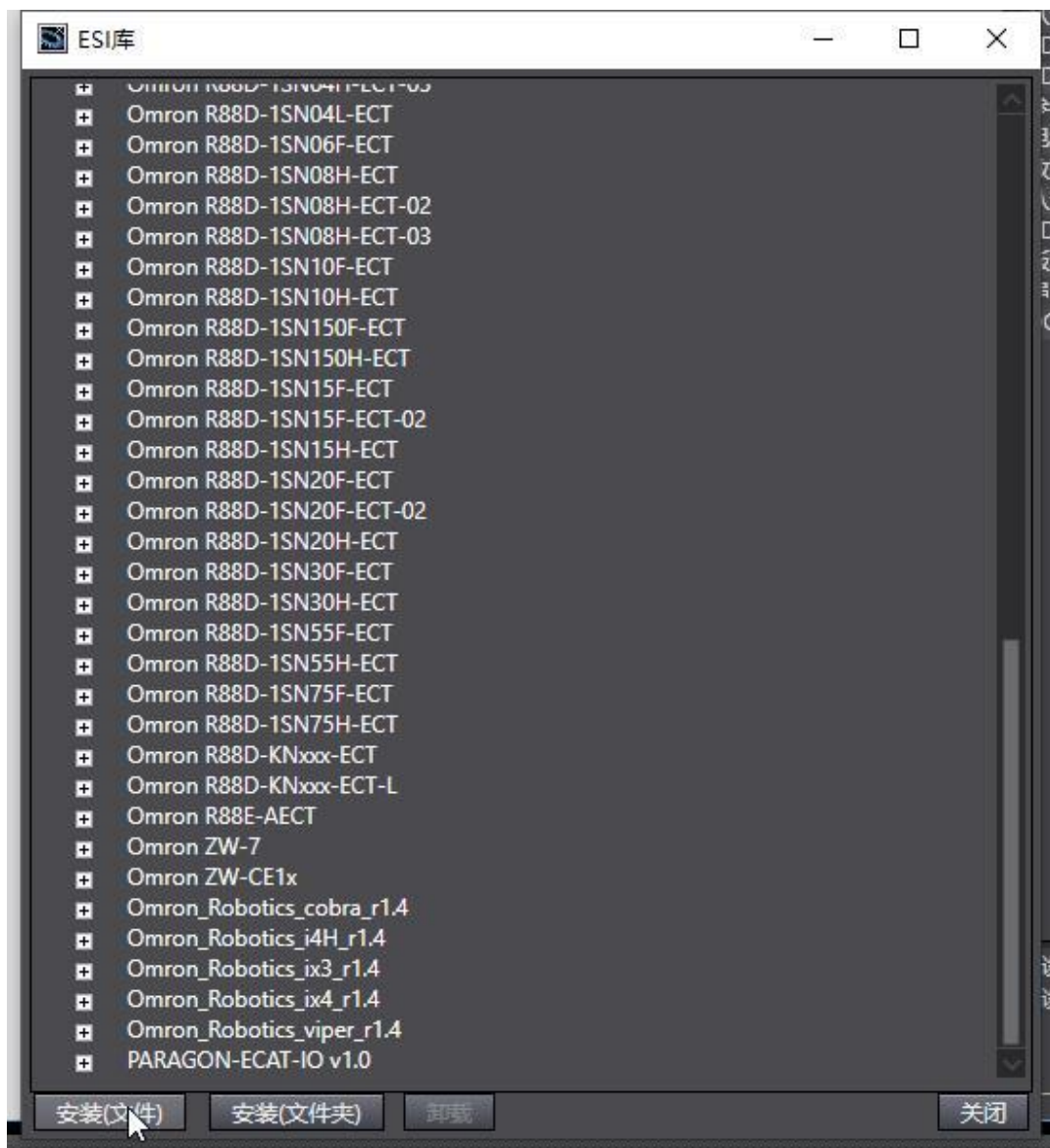
确保在此之前已经将电脑 IP 设置为 192.168.250.xx 网段，这是 PLC 要求的



点击 Ethernet 通信测试，显示测试成功，说明网络没问题、

③双击工程目录下的 EtherCAT，右击“主设备”选择“显示 ESI 库”

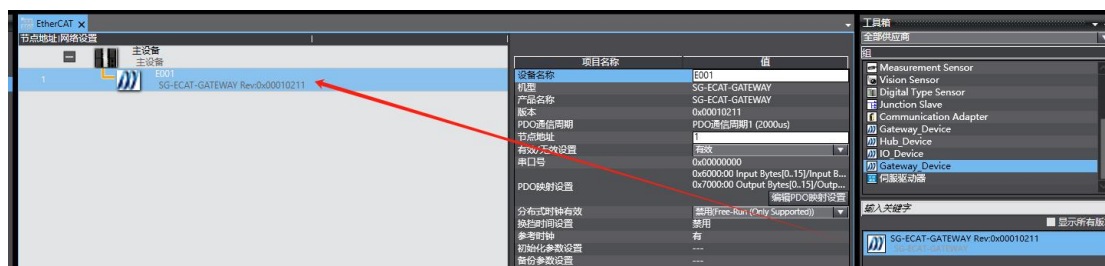




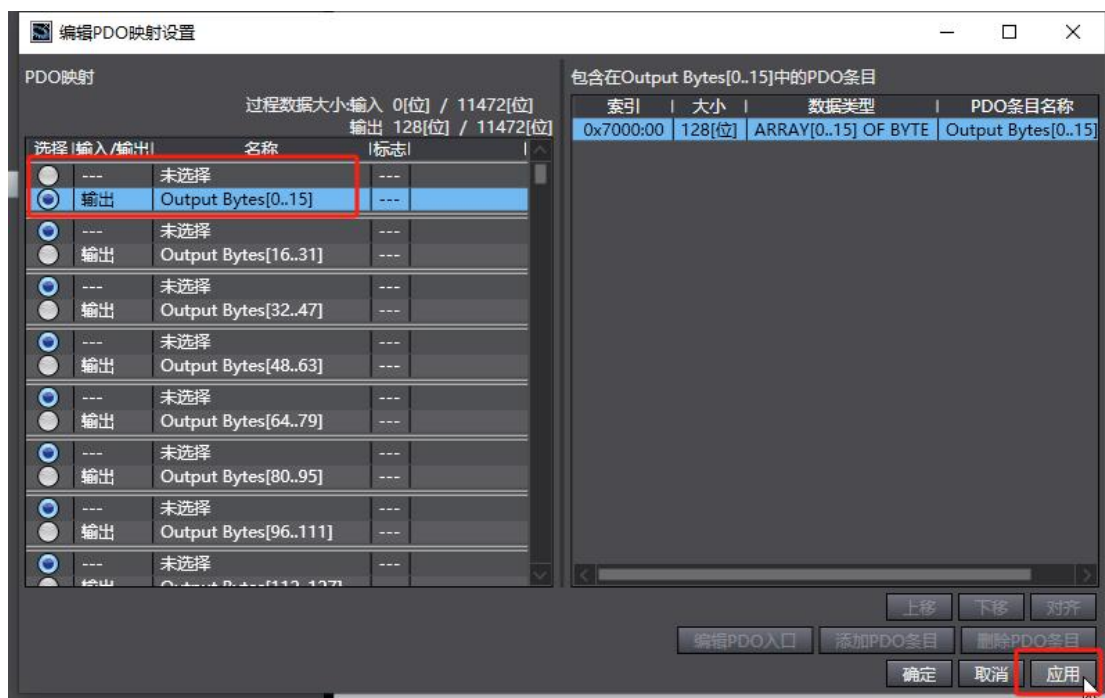
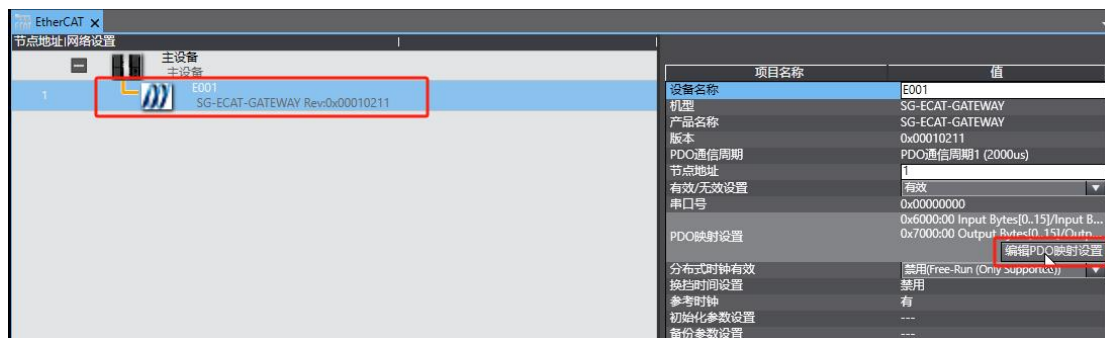
点击“安装（文件）”，选择下载的 ESI 文件并安装

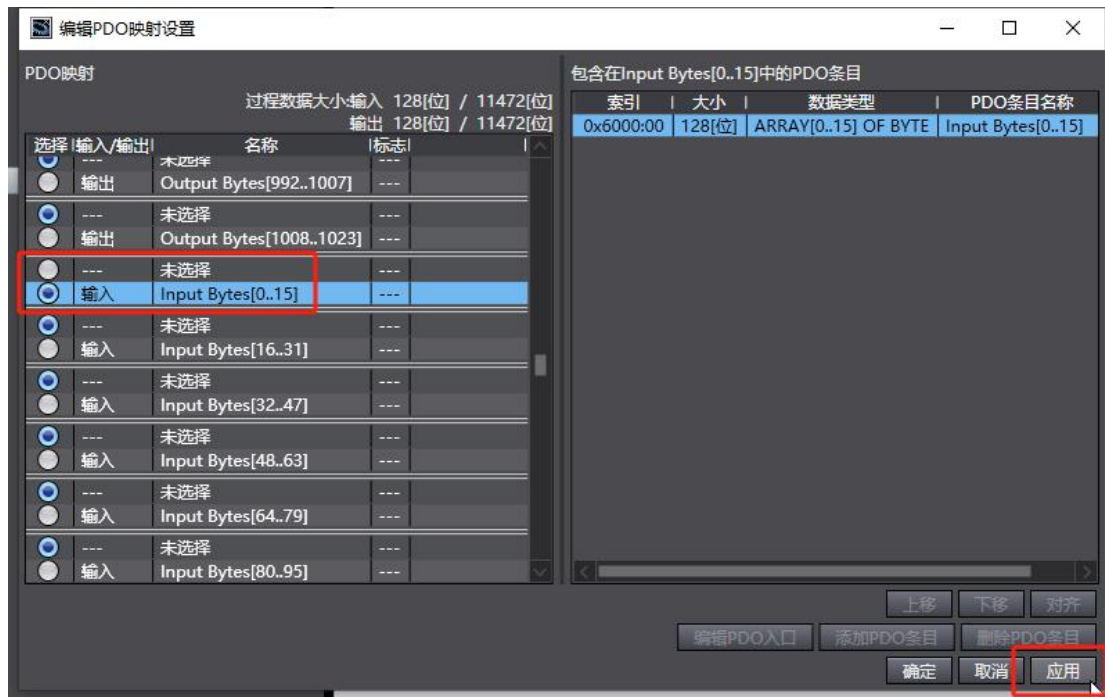


④在主站下添加网关，如下图所示，在右上角选择 Tianjin Sange Elc Co.Ltd 然后选择网关并拖动到主站下面。

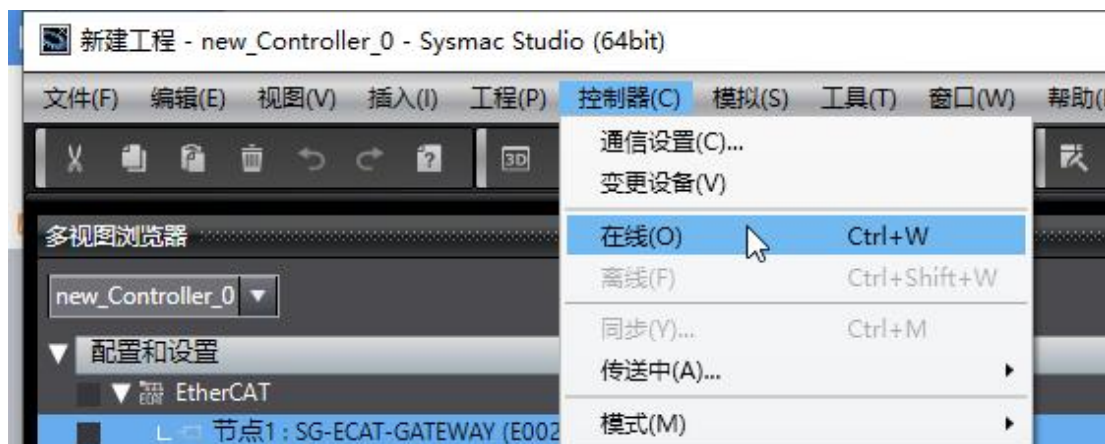


设置网关 PDO

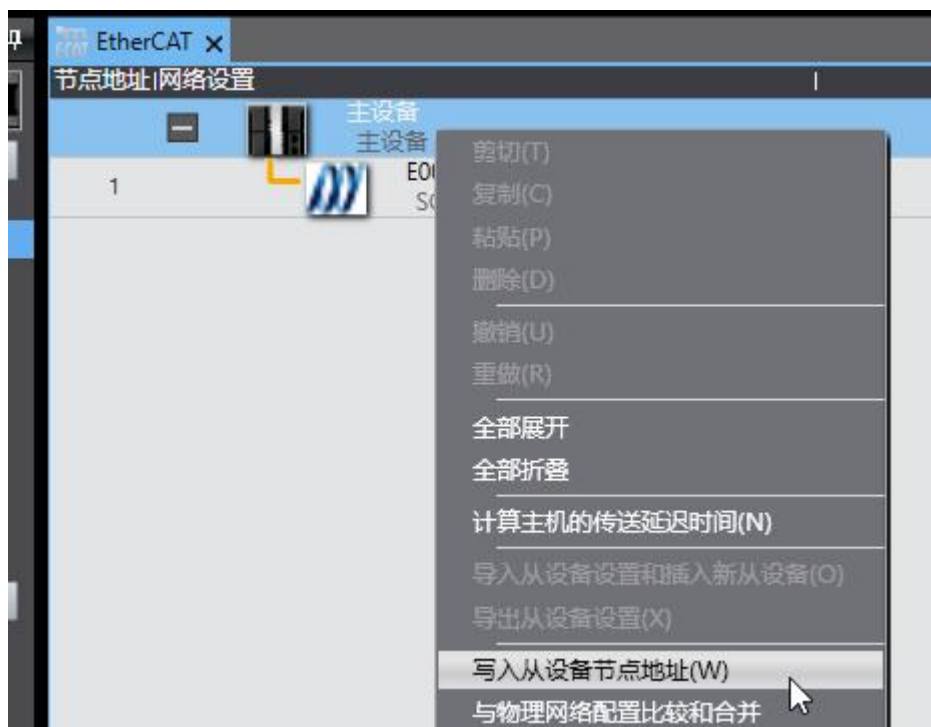




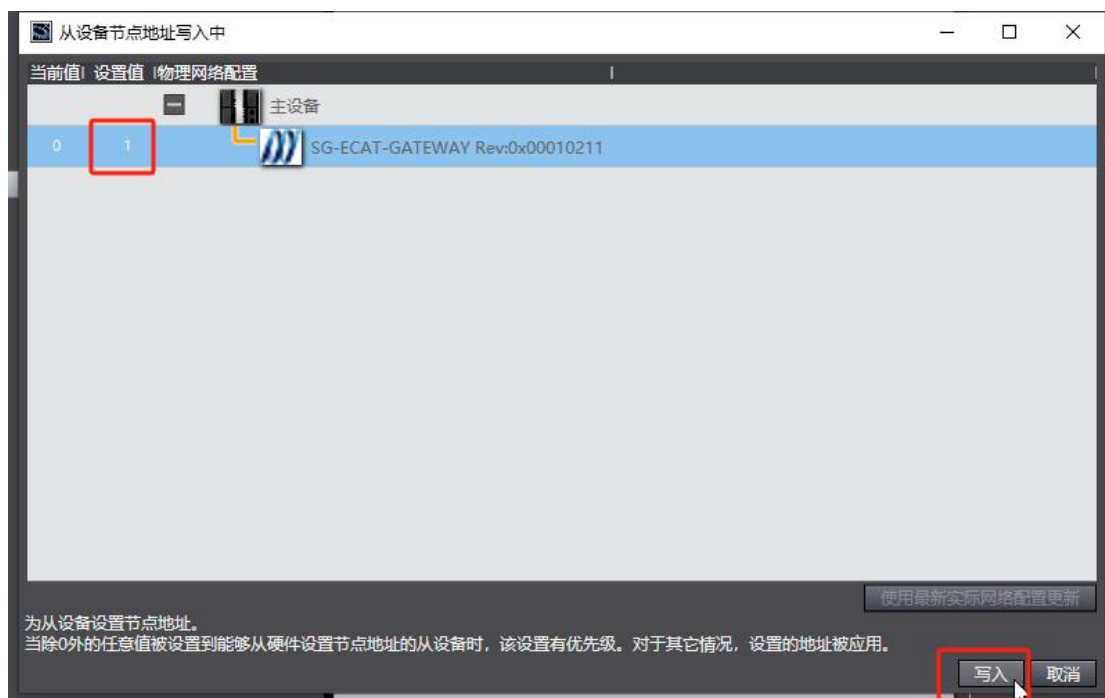
⑤点击控制器，选择“在线”

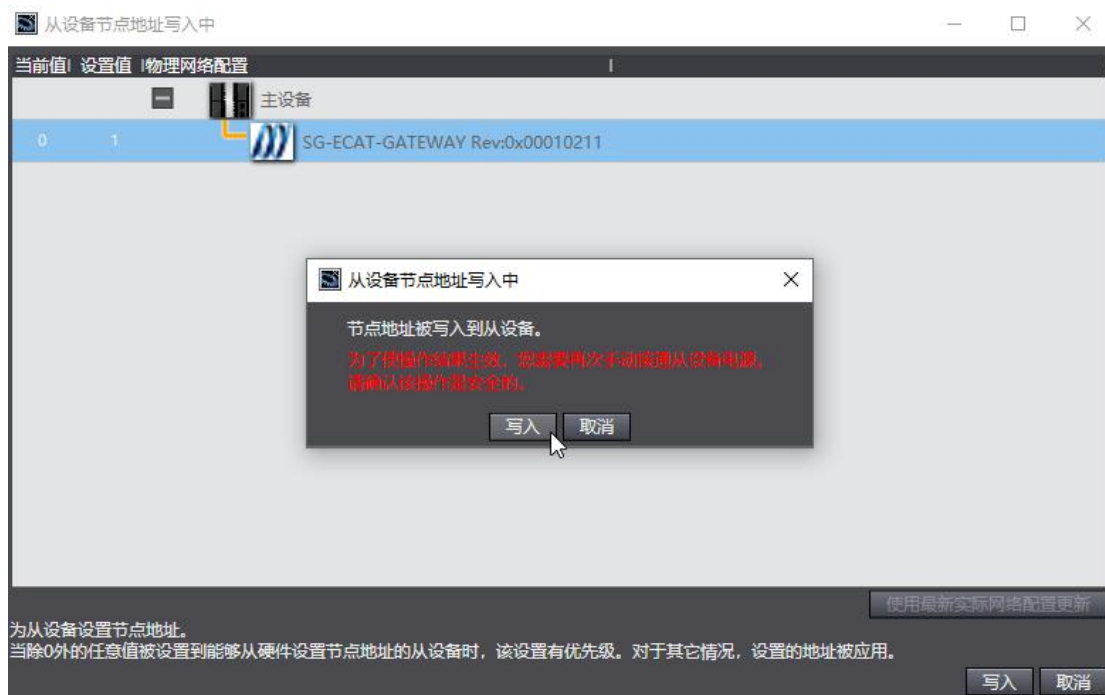


右击主设备，选择“写入从设备节点地址”



如下图，输入从设备节点地址，点击写入

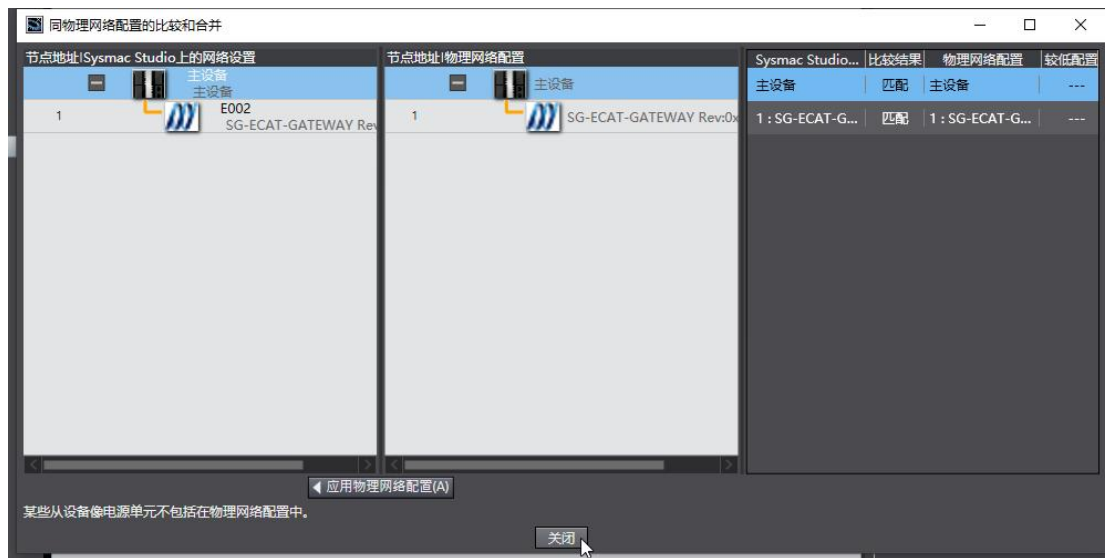




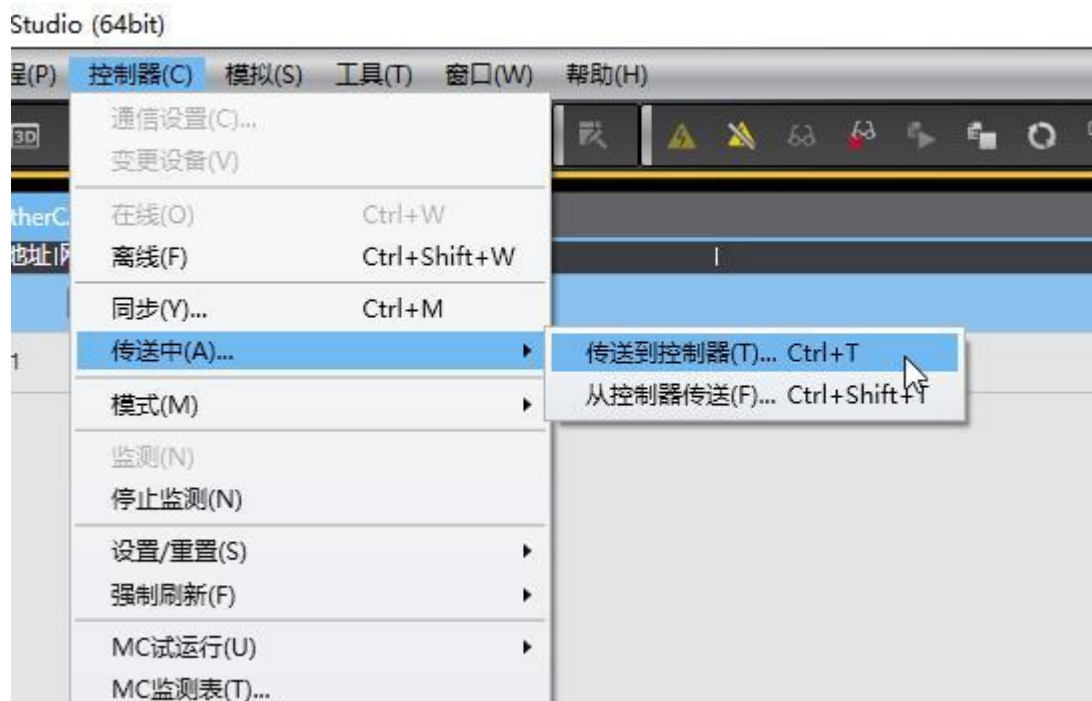
根据提示，给远程 IO 重上电

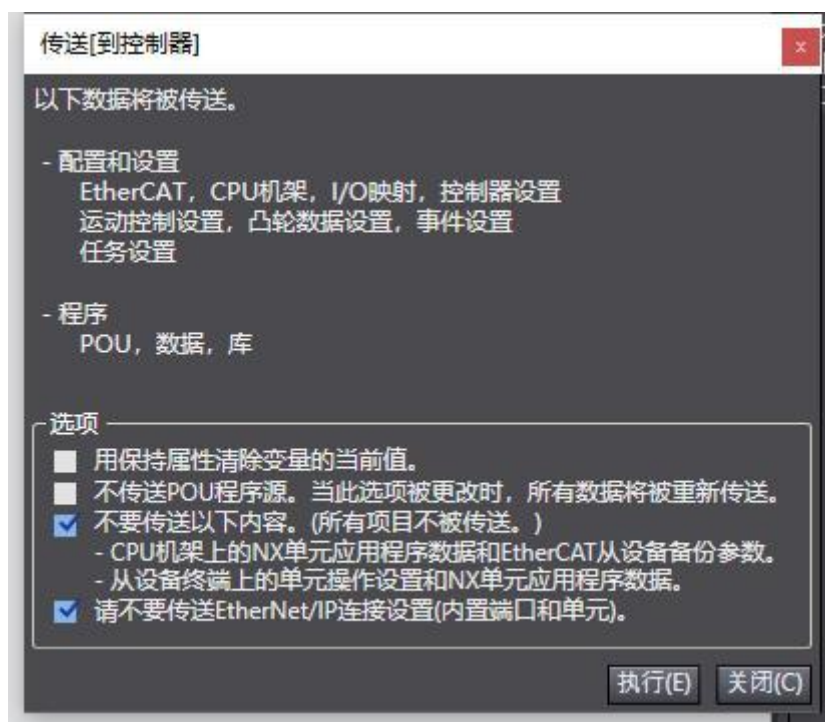
⑥右击主设备选择“与物理网络比较和合并”





如果弹出如下“匹配”，则没问题，否则检查配置和硬件
 点击控制器“传送中”->“传送到控制器”





后面一路点确定直到下载完组态。

⑦在 I/O 映射里面可以查看、写入数据

首先设置模式为轮廓速度模式，在 Output Bytes[6]写 3，之后 Input Bytes[7]显示 3。

位置	端口	说明	R/W	数据类型	值
	▼ EtherCAT网络配置				
节点1	▼ SG-ECAT-GATEWAY				
	▼ Input Bytes[0..15]_Input Bytes[0..15]_6		R	ARRAY[0..15]	
	[0]		R	BYTE	16#5
	[1]		R	BYTE	16#50
	[2]		R	BYTE	16#12
	[3]		R	BYTE	16#0
	[4]		R	BYTE	16#0
	[5]		R	BYTE	16#0
	[6]		R	BYTE	16#0
	[7]		R	BYTE	16#3
	[8]		R	BYTE	16#0
	[9]		R	BYTE	16#0
	[10]		R	BYTE	16#0
	[11]		R	BYTE	16#0
	[12]		R	BYTE	16#0
	[13]		R	BYTE	16#0
	[14]		R	BYTE	16#0
	[15]		R	BYTE	16#0
	▼ Output Bytes[0..15]_Output Bytes[0..15]_6		W	ARRAY[0..15]	
	[0]		W	BYTE	16#0
	[1]		W	BYTE	16#0
	[2]		W	BYTE	16#0
	[3]		W	BYTE	16#0
	[4]		W	BYTE	16#0
	[5]		W	BYTE	16#0
	[6]		W	BYTE	16#3
	[7]		W	BYTE	16#0
	[8]		W	BYTE	16#0
	[9]		W	BYTE	16#0
	[10]		W	BYTE	16#0
	[11]		W	BYTE	16#0
	[12]		W	BYTE	16#0
	[13]		W	BYTE	16#0
	[14]		W	BYTE	16#0
	[15]		W	BYTE	16#0

之后按 CIA402 规约引导驱动器运行，给控制字依次写入 6、7、F。注意，CANOPEN 是小端的，所以给控制字 Output Bytes[0..1]写 6、7、F 应该是 Bytes[0]写 6、7、F

位置	端口	说明	R/W	数据类型	值
节点1	▼ EtherCAT网络配置				
	▼ SG-ECAT-GATEWAY				
	▼ Input Bytes[0..15]_Input Bytes[0..15]_6		R	ARRAY[0..15]	
	[0]		R	BYTE	16#5
	[1]		R	BYTE	16#37
	[2]		R	BYTE	16#16
	[3]		R	BYTE	16#0
	[4]		R	BYTE	16#0
	[5]		R	BYTE	16#0
	[6]		R	BYTE	16#0
	[7]		R	BYTE	16#3
	[8]		R	BYTE	16#0
	[9]		R	BYTE	16#0
	[10]		R	BYTE	16#0
	[11]		R	BYTE	16#0
	[12]		R	BYTE	16#0
	[13]		R	BYTE	16#0
	[14]		R	BYTE	16#0
	[15]		R	BYTE	16#0
	▼ Output Bytes[0..15]_Output Bytes[0..15]_6		W	ARRAY[0..15]	
	[0]		W	BYTE	16#F
	[1]		W	BYTE	16#0
	[2]		W	BYTE	16#0
	[3]		W	BYTE	16#0
	[4]		W	BYTE	16#0
	[5]		W	BYTE	16#0
	[6]		W	BYTE	16#3
	[7]		W	BYTE	16#0
	[8]		W	BYTE	16#0
	[9]		W	BYTE	16#0
	[10]		W	BYTE	16#0
	[11]		W	BYTE	16#0
	[12]		W	BYTE	16#0
	[13]		W	BYTE	16#0
	[14]		W	BYTE	16#0
	[15]		W	BYTE	16#0

之后给目标速度一个值，比如 1000000(0x000F4240)，注意 CANOPEN 是小端的，所以给控制字 Output Bytes[2..5]写 0x000F4240 是 Bytes[2..5]为 0x40、0xF2、0x0F、0x00。

位置	端口	说明	R/W	数据类型	值
EtherCAT网络配置					
节点1	SG-ECAT-GATEWAY				
	Input Bytes[0..15]_Input Bytes[0..15]_6				
	[0]		R	ARRAY[0..15]	
	[1]		R	BYTE	16#5
	[2]		R	BYTE	16#37
	[3]		R	BYTE	16#16
	[4]		R	BYTE	16#EE
	[5]		R	BYTE	16#EE
	[6]		R	BYTE	16#E
	[7]		R	BYTE	16#0
	[8]		R	BYTE	16#3
	[9]		R	BYTE	16#0
	[10]		R	BYTE	16#0
	[11]		R	BYTE	16#0
	[12]		R	BYTE	16#0
	[13]		R	BYTE	16#0
	[14]		R	BYTE	16#0
	[15]		R	BYTE	16#0
	Output Bytes[0..15]_Output Bytes[0..15]				
	[0]		W	ARRAY[0..15]	
	[1]		W	BYTE	16#F
	[2]		W	BYTE	16#0
	[3]		W	BYTE	16#40
	[4]		W	BYTE	16#F2
	[5]		W	BYTE	16#F
	[6]		W	BYTE	16#0
	[7]		W	BYTE	16#3
	[8]		W	BYTE	16#0
	[9]		W	BYTE	16#0
	[10]		W	BYTE	16#0
	[11]		W	BYTE	16#0
	[12]		W	BYTE	16#0
	[13]		W	BYTE	16#0
	[14]		W	BYTE	16#0
	[15]		W	BYTE	16#0

之后电机缓慢转起来，位置反馈 Input Bytes[3..6]显示实际速度，实际速度和设置有很小偏差。

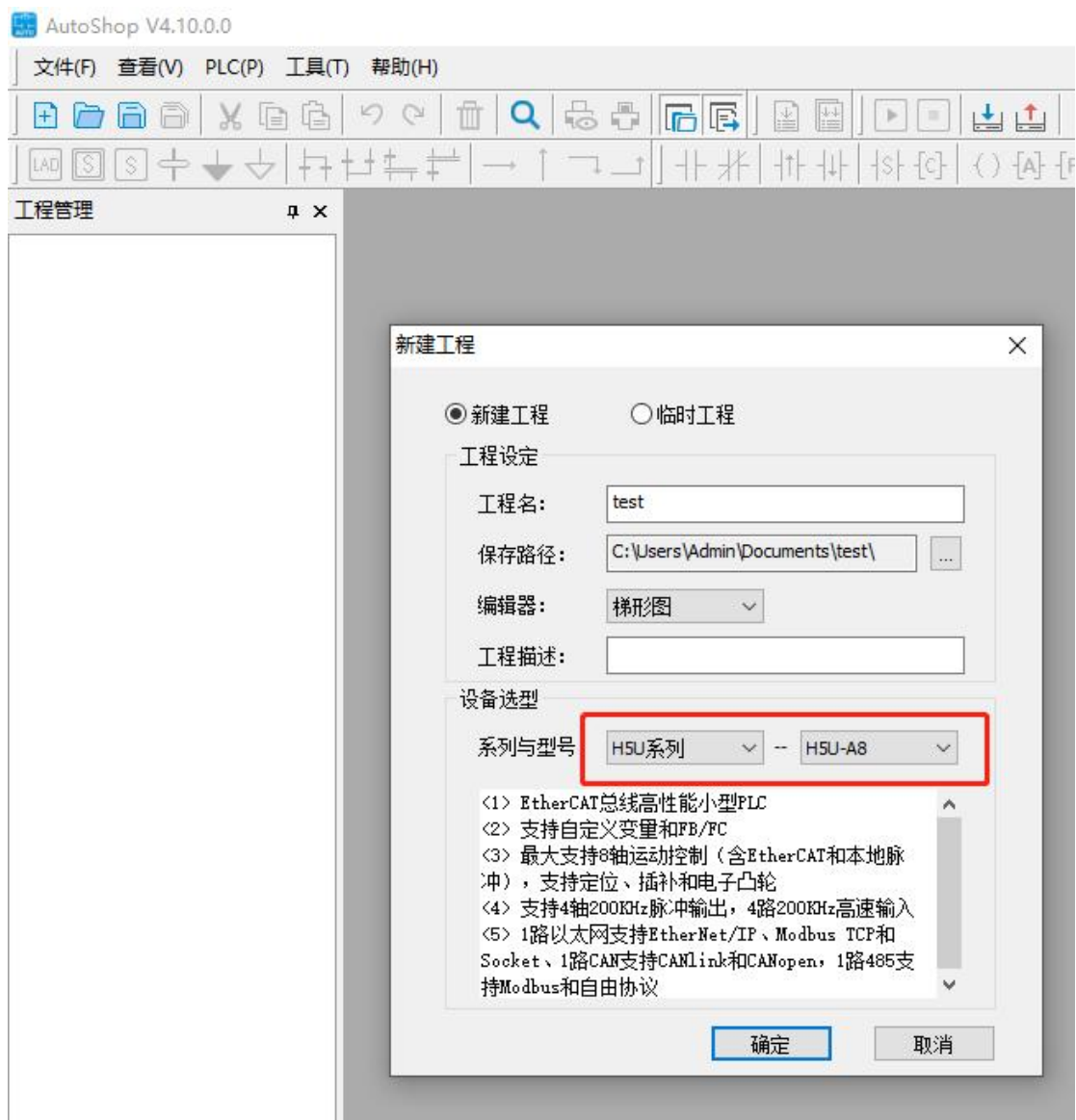
5.5 汇川 H5U-A8 例程

先按 5.1 设置伺服驱动器和网关。

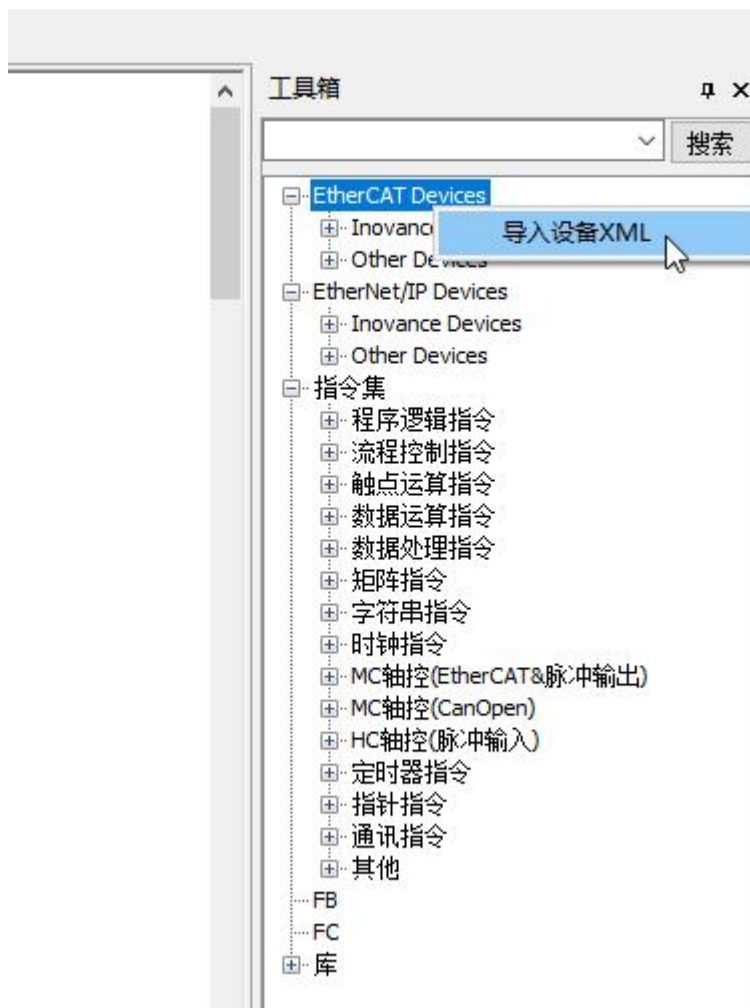
①首先去我司网站下载设备的 ESI 文件。

连接电脑、汇川 H5U-A8 和网关，注意 PLC 的 EtherCAT 口接网关的 Port1（ECAT 两个网口区分输入输出，别接反了）。

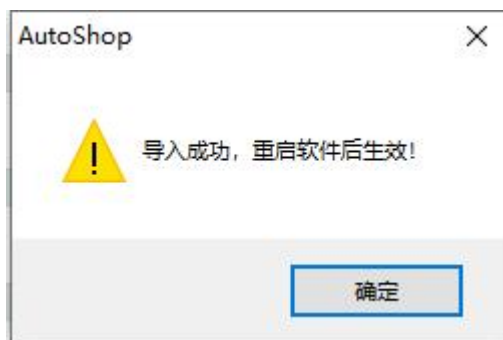
②打开 AutoShop，创建工程，注意 PLC 型号



③安装 XML 文件。

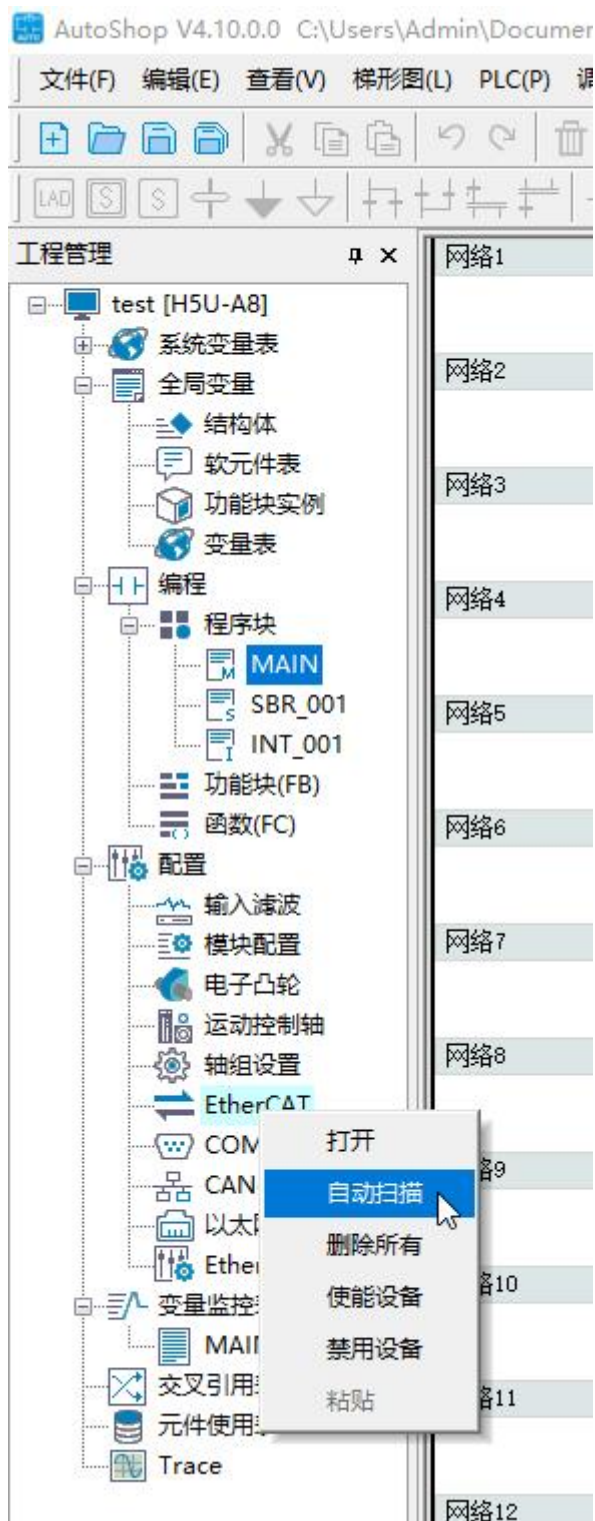


在 XML 存放目录选择 XML 文件



安装完成之后需要重启 AutoShop 并重新建立工程

④在工程管理下右键 EtherCAT，选择“自动扫描”

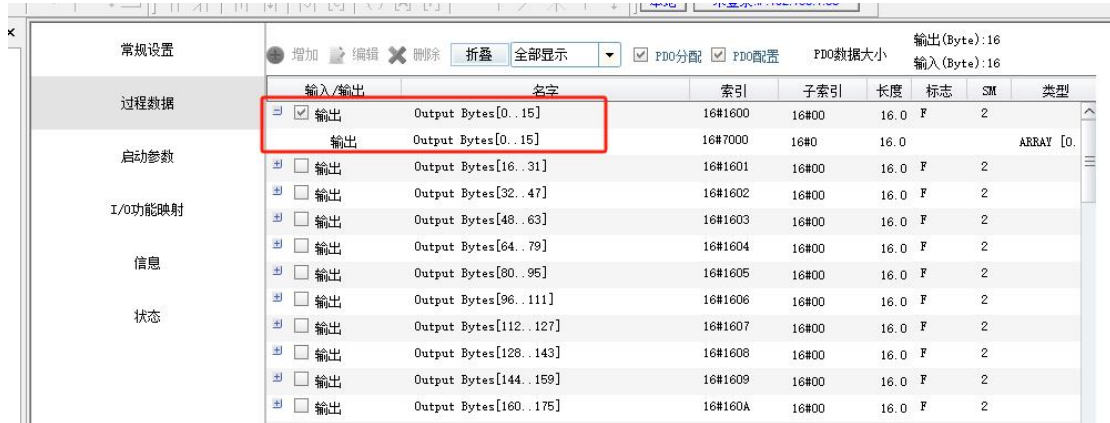




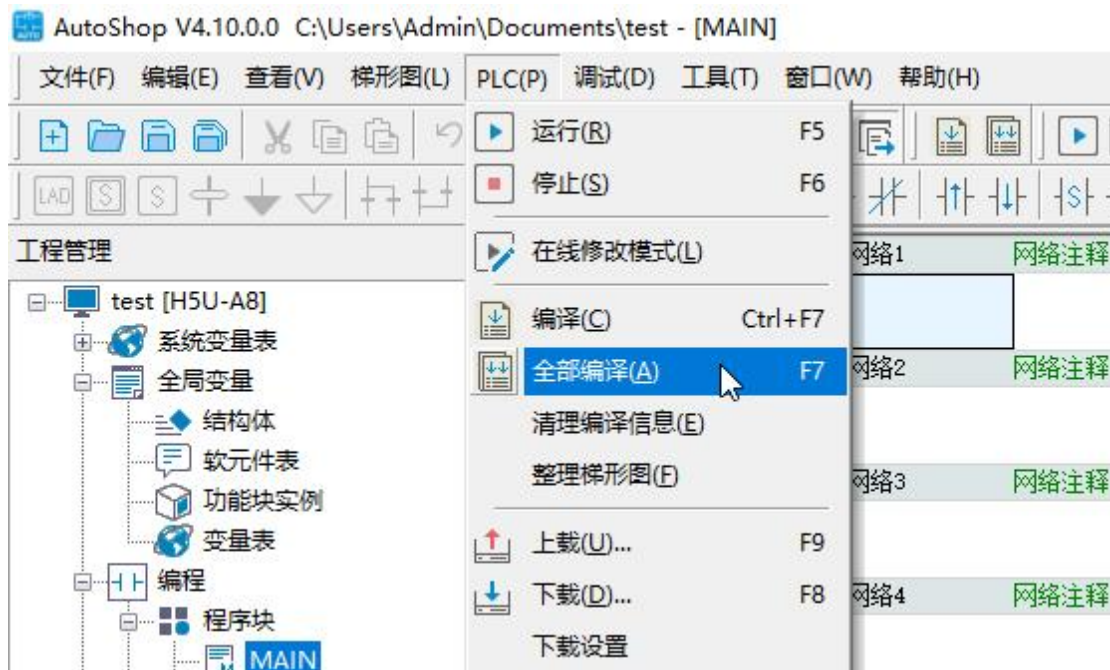
自动扫描列表会出现本公司的网关，之后点击“更新组态”

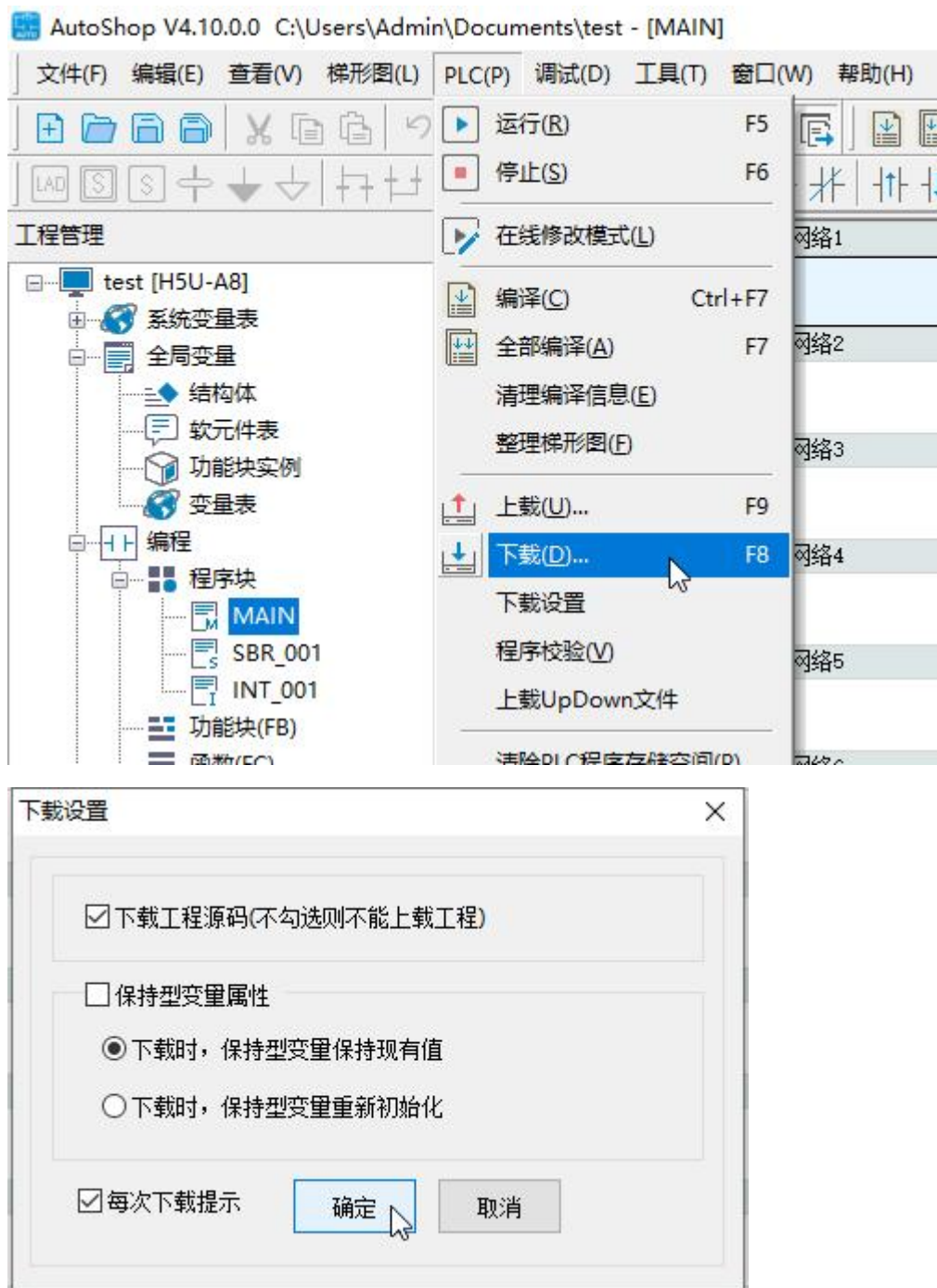


⑤根据所需要的 IO 数据量选择适当的 TPDO 和 RPDO

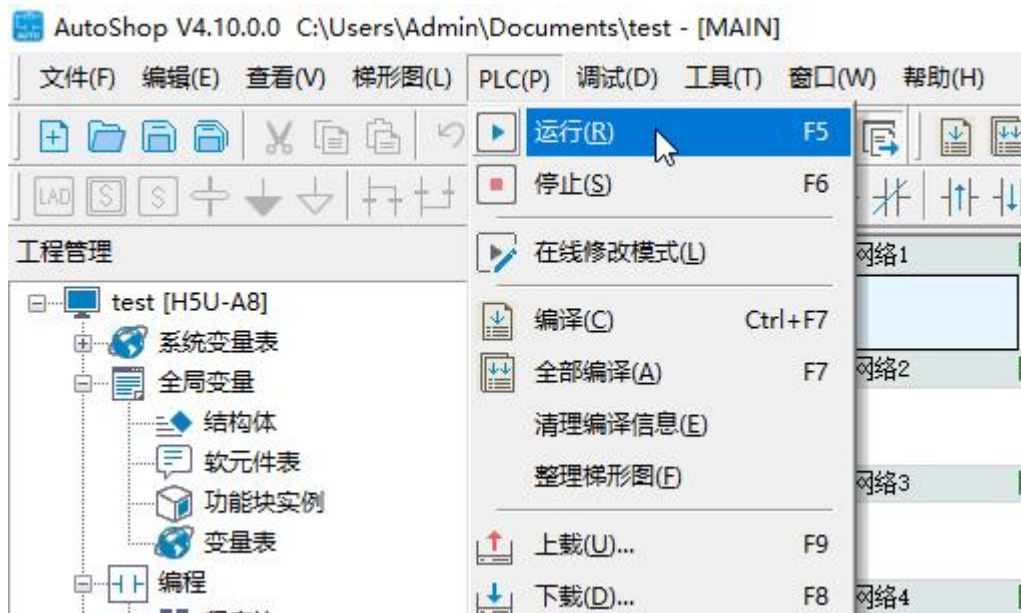


⑥编译并下载程序



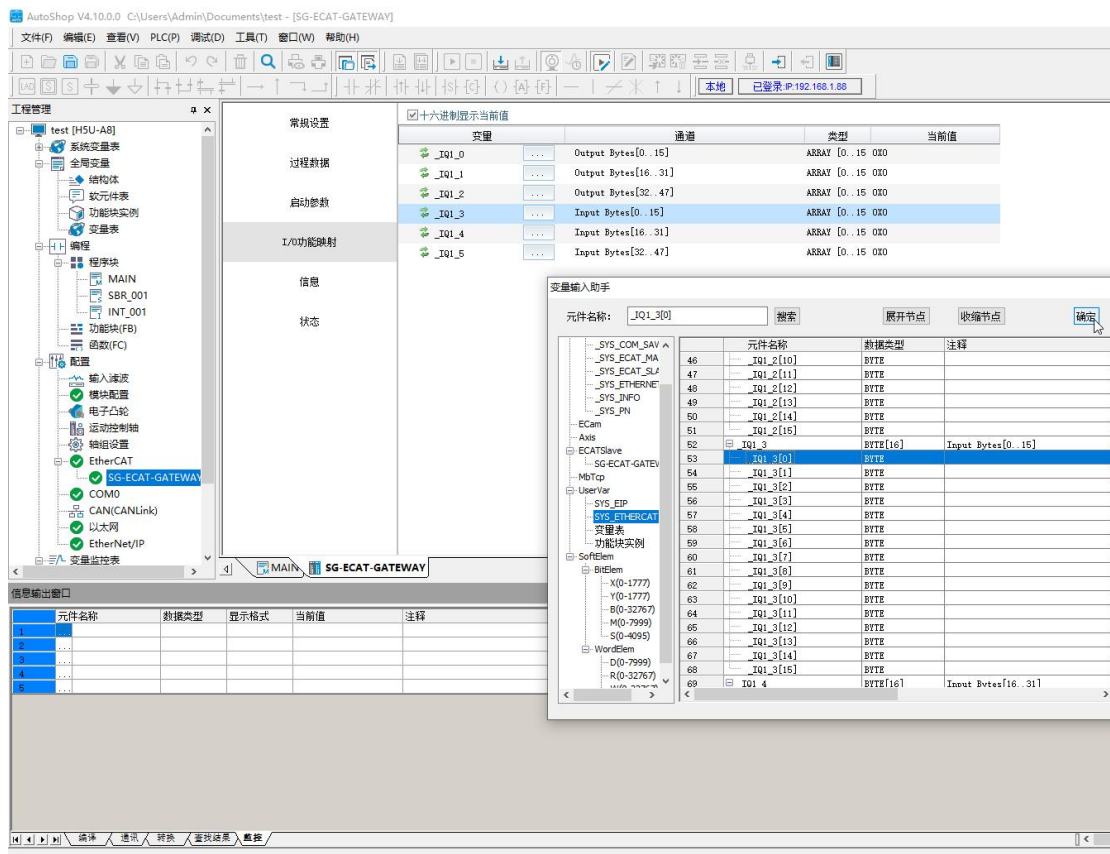


⑦运行 PLC



⑧之后网关 OP 灯亮起，说明进入了 OP 状态

在信息输出窗口查看数据，因为每个 PDO 映射的是一个 16 个字节数组，所以需要一个个添加上去：



⑨在信息输出窗口里面可以查看、写入数据

首先设置模式为轮廓速度模式，在 Output Bytes[6]写 3，之后 Input Bytes[7]

显示 3。

信息输出窗口					
	元件名称	数据类型	显示格式	当前值	注释
1	... ☐ IQ1_1	BYTE[16]			Input Bytes[0..15]
2	... IQ1_1[0]	BYTE	十六进制	0x5	
3	... IQ1_1[1]	BYTE	十六进制	0x50	
4	... IQ1_1[2]	BYTE	十六进制	0x12	
5	... IQ1_1[3]	BYTE	十六进制	0x0	
6	... IQ1_1[4]	BYTE	十六进制	0x0	
7	... IQ1_1[5]	BYTE	十六进制	0x0	
8	... IQ1_1[6]	BYTE	十六进制	0x0	
9	... IQ1_1[7]	BYTE	十六进制	0x3	
10	... IQ1_1[8]	BYTE	十进制	0	
11	... IQ1_1[9]	BYTE	十进制	0	
12	... IQ1_1[10]	BYTE	十进制	0	
13	... IQ1_1[11]	BYTE	十进制	0	
14	... IQ1_1[12]	BYTE	十进制	0	
15	... IQ1_1[13]	BYTE	十进制	0	
16	... IQ1_1[14]	BYTE	十进制	0	
17	... IQ1_1[15]	BYTE	十进制	0	
18	... ☐ IQ1_0	BYTE[16]			Output Bytes[0..15]
19	... IQ1_0[0]	BYTE	十六进制	0x0	
20	... IQ1_0[1]	BYTE	十六进制	0x0	
21	... IQ1_0[2]	BYTE	十六进制	0x0	
22	... IQ1_0[3]	BYTE	十六进制	0x0	
23	... IQ1_0[4]	BYTE	十六进制	0x0	
24	... IQ1_0[5]	BYTE	十六进制	0x0	
25	... IQ1_0[6]	BYTE	十六进制	0x3	
26	... IQ1_0[7]	BYTE	十进制	0	
27	... IQ1_0[8]	BYTE	十进制	0	
28	... IQ1_0[9]	BYTE	十进制	0	
29	... IQ1_0[10]	BYTE	十进制	0	
30	... IQ1_0[11]	BYTE	十进制	0	
31	... IQ1_0[12]	BYTE	十进制	0	
32	... IQ1_0[13]	BYTE	十进制	0	
33	... IQ1_0[14]	BYTE	十进制	0	
34	... IQ1_0[15]	BYTE	十进制	0	

之后按 CIA402 规约引导驱动器运行，给控制字依次写入 6、7、F。注意，CANOPEN 是小端的，所以给控制字 Output Bytes[0..1]写 6、7、F 应该是 Bytes[0]写 6、7、F

信息输出窗口					
	元件名称	数据类型	显示格式	当前值	注释
1	... ☐ _IQ1_1	BYTE[16]			Input Bytes[0..15]
2	... _IQ1_1[0]	BYTE	十六进制	0x5	
3	... _IQ1_1[1]	BYTE	十六进制	0x37	
4	... _IQ1_1[2]	BYTE	十六进制	0x16	
5	... _IQ1_1[3]	BYTE	十六进制	0x0	
6	... _IQ1_1[4]	BYTE	十六进制	0x0	
7	... _IQ1_1[5]	BYTE	十六进制	0x0	
8	... _IQ1_1[6]	BYTE	十六进制	0x0	
9	... _IQ1_1[7]	BYTE	十六进制	0x3	
10	... _IQ1_1[8]	BYTE	十进制	0	
11	... _IQ1_1[9]	BYTE	十进制	0	
12	... _IQ1_1[10]	BYTE	十进制	0	
13	... _IQ1_1[11]	BYTE	十进制	0	
14	... _IQ1_1[12]	BYTE	十进制	0	
15	... _IQ1_1[13]	BYTE	十进制	0	
16	... _IQ1_1[14]	BYTE	十进制	0	
17	... _IQ1_1[15]	BYTE	十进制	0	
18	... ☐ _IQ1_0	BYTE[16]			Output Bytes[0..15]
19	... _IQ1_0[0]	BYTE	十六进制	0xF	
20	... _IQ1_0[1]	BYTE	十六进制	0x0	
21	... _IQ1_0[2]	BYTE	十六进制	0x0	
22	... _IQ1_0[3]	BYTE	十六进制	0x0	
23	... _IQ1_0[4]	BYTE	十六进制	0x0	
24	... _IQ1_0[5]	BYTE	十六进制	0x0	
25	... _IQ1_0[6]	BYTE	十六进制	0x3	
26	... _IQ1_0[7]	BYTE	十进制	0	
27	... _IQ1_0[8]	BYTE	十进制	0	
28	... _IQ1_0[9]	BYTE	十进制	0	
29	... _IQ1_0[10]	BYTE	十进制	0	
30	... _IQ1_0[11]	BYTE	十进制	0	
31	... _IQ1_0[12]	BYTE	十进制	0	
32	... _IQ1_0[13]	BYTE	十进制	0	
33	... _IQ1_0[14]	BYTE	十进制	0	
34	... _IQ1_0[15]	BYTE	十进制	0	

之后给目标速度一个值，比如 1000000(0x000F4240)，注意 CANOPEN 是小端的，所以给控制字 Output Bytes[2..5]写 0x000F4240 是 Bytes[2..5]为 0x40、0xF2、0x0F、0x00。

信息输出窗口					
	元件名称	数据类型	显示格式	当前值	注释
1	... ☐ IQ1_1	BYTE[16]			Input Bytes[0..15]
2	... IQ1_1[0]	BYTE	十六进制	0x5	
3	... IQ1_1[1]	BYTE	十六进制	0x37	
4	... IQ1_1[2]	BYTE	十六进制	0x16	
5	... IQ1_1[3]	BYTE	十六进制	0xEE	
6	... IQ1_1[4]	BYTE	十六进制	0xEE	
7	... IQ1_1[5]	BYTE	十六进制	0xE	
8	... IQ1_1[6]	BYTE	十六进制	0x0	
9	... IQ1_1[7]	BYTE	十六进制	0x3	
10	... IQ1_1[8]	BYTE	十进制	0	
11	... IQ1_1[9]	BYTE	十进制	0	
12	... IQ1_1[10]	BYTE	十进制	0	
13	... IQ1_1[11]	BYTE	十进制	0	
14	... IQ1_1[12]	BYTE	十进制	0	
15	... IQ1_1[13]	BYTE	十进制	0	
16	... IQ1_1[14]	BYTE	十进制	0	
17	... IQ1_1[15]	BYTE	十进制	0	
18	... ☐ IQ1_0	BYTE[16]			Output Bytes[0..15]
19	... IQ1_0[0]	BYTE	十六进制	0xF	
20	... IQ1_0[1]	BYTE	十六进制	0x0	
21	... IQ1_0[2]	BYTE	十六进制	0x40	
22	... IQ1_0[3]	BYTE	十六进制	0xF2	
23	... IQ1_0[4]	BYTE	十六进制	0xF	
24	... IQ1_0[5]	BYTE	十六进制	0x0	
25	... IQ1_0[6]	BYTE	十六进制	0x3	
26	... IQ1_0[7]	BYTE	十进制	0	
27	... IQ1_0[8]	BYTE	十进制	0	
28	... IQ1_0[9]	BYTE	十进制	0	
29	... IQ1_0[10]	BYTE	十进制	0	
30	... IQ1_0[11]	BYTE	十进制	0	
31	... IQ1_0[12]	BYTE	十进制	0	
32	... IQ1_0[13]	BYTE	十进制	0	
33	... IQ1_0[14]	BYTE	十进制	0	
34	... IQ1_0[15]	BYTE	十进制	0	

之后电机缓慢转起来，位置反馈 Input Bytes[3..6]显示实际速度，实际速度和设置有很小偏差。

六、ECAT-CAN(自由口)应用实例

实例拓扑如下图，ECAT 主站与 CAN 分析仪交互数据。



ECAT 主站分别使用 TwinCAT3、CODESYS、欧姆龙 NX1P2 和汇川 H5U-A8。

6.1 实例预操作

1、先按如下配置网关，这个配置包含到了常用功能，作为一个典型用例。

CAN波特率 (5000-1000000) 数据帧发送周期(按ID收发CAN帧)

远程帧发送周期(按ID收发CAN帧) (1-65535)ms 数据帧接收超时(按ID收发CAN帧)

按ID收发CAN帧 通用收发CAN帧

ID收发CAN数量 (1-200)

序号	发送/接收	帧ID类型	数据/远程帧	数据长度	帧ID(HEX)	数据偏移/长度
1	按ID发送	标准帧	数据帧	1	00000123	输出+0/1
2	按ID发送	标准帧	远程帧	0	00000234	---
3	按ID接收	扩展帧	数据帧	1	12345678	输入+0/1

ECAT-CAN v1.1

CAN波特率 (5000-1000000) 数据帧发送周期(按ID收发CAN帧) (0-65535)

远程帧发送周期(按ID收发CAN帧) (1-65535)ms 数据帧接收超时(按ID收发CAN帧) (0-65535)

按ID收发CAN帧 通用收发CAN帧

通用收发CAN数量 (1-200)

序号	发送/接收	PLC序号偏移/长度	网关序号偏移/长度	帧标志偏移/长度	帧ID偏移/长度	数据偏移/长度
1	通用发送	输出+1/1	输入+1/1	输出+2/1	输出+3/4	输出+7/8
2	通用接收	输出+15/1	输入+2/1	输入+3/1	输入+4/4	输入+8/8

2、根据上图我们确定下来 ECAT 最少需要 15+1=16 个输入字节和 8+9=16 个输出字节。

3、配置网关，弹出网关配置成功。数据映射表如下：

按 ID 发送 0x123 数据位置	Output Bytes[0]
按 ID 发送 0x234 远程帧	不携带数据
按 ID 接收 0x12345678 数据接收位置	Input Bytes[0]
通用发送 PLC 序号位置	Output Bytes[1]
通用发送网关序号位置	Input Bytes[1]
通用发送帧标志位置	Output Bytes[2]
通用发送帧 ID 位置	Output Bytes[3..6]
通用发送帧数据位置	Output Bytes[7..14]
通用接收 PLC 序号位置	Output Bytes[15]
通用接收网关序号位置	Input Bytes[2]
通用接收帧标志位置	Input Bytes[3]
通用接收帧 ID 位置	Input Bytes[4..7]
通用接收帧数据位置	Input Bytes[8..15]

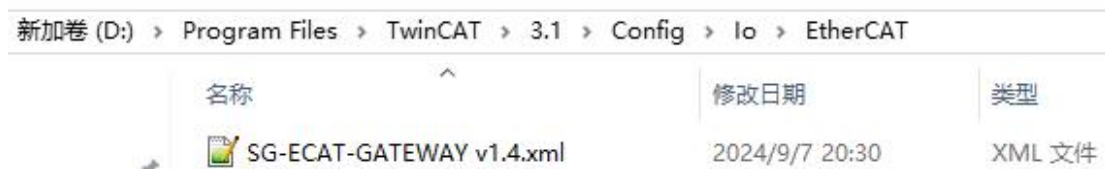
6.2 TwinCAT 实例

PS: TwinCAT 对网卡型号有要求，要确保所使用的网卡支持 TwinCAT。

先按 6.1 设置网关。

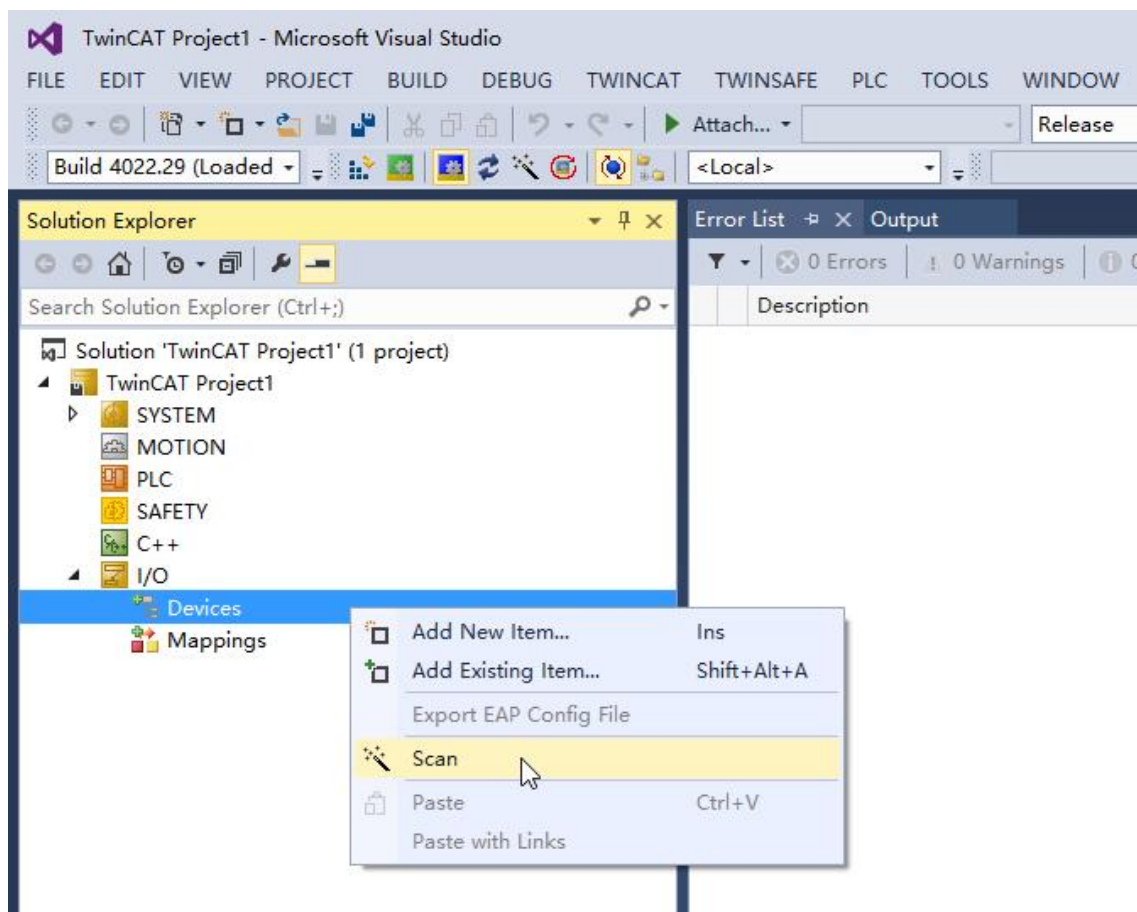
①去我司网站下载网关的 ESI 文件。

复制 ESI 文件到 TwinCAT 安装目录的 EtherCAT 目录下，如：D:\Program Files\TwinCAT 3\3.1\Config\Io\EtherCAT。

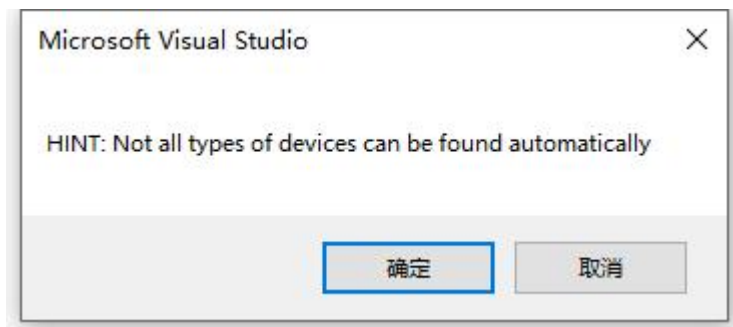


②连接电脑和网关，注意电脑连接网关的 Port1（ECAT 两个网口区分输入输出，别接反了）。

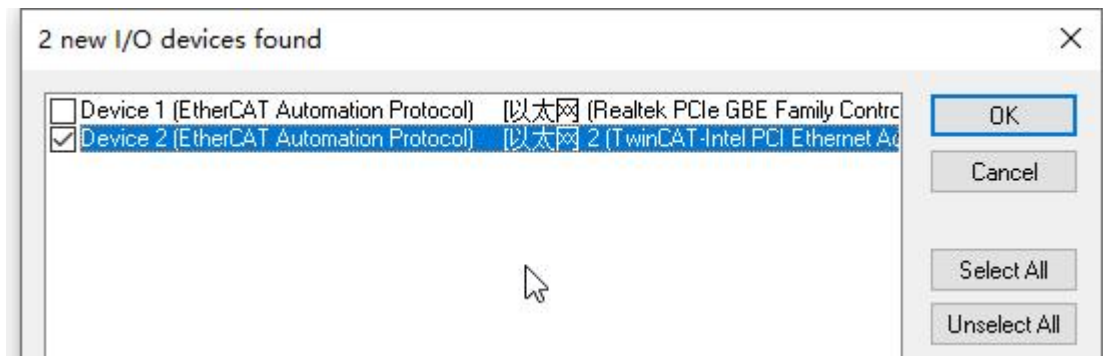
建立 TwinCAT 工程，右键 Device 并点 Scan。



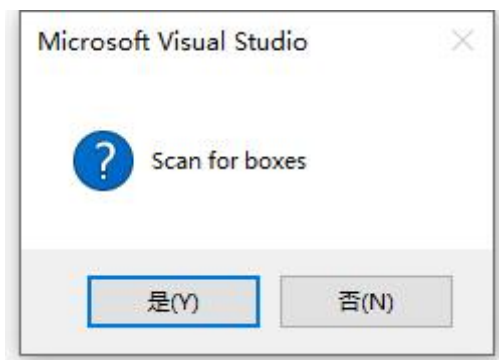
在下面这个对话框选择“确定”



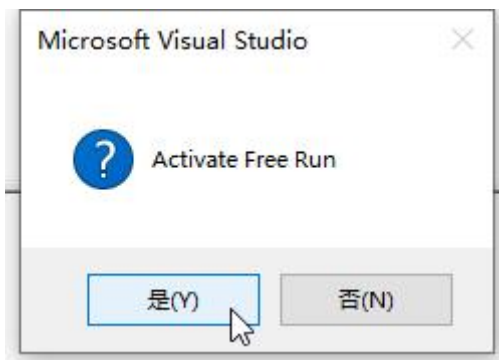
在下面这个对话框选择 TwinCAT 使用的网卡并选择“OK”。



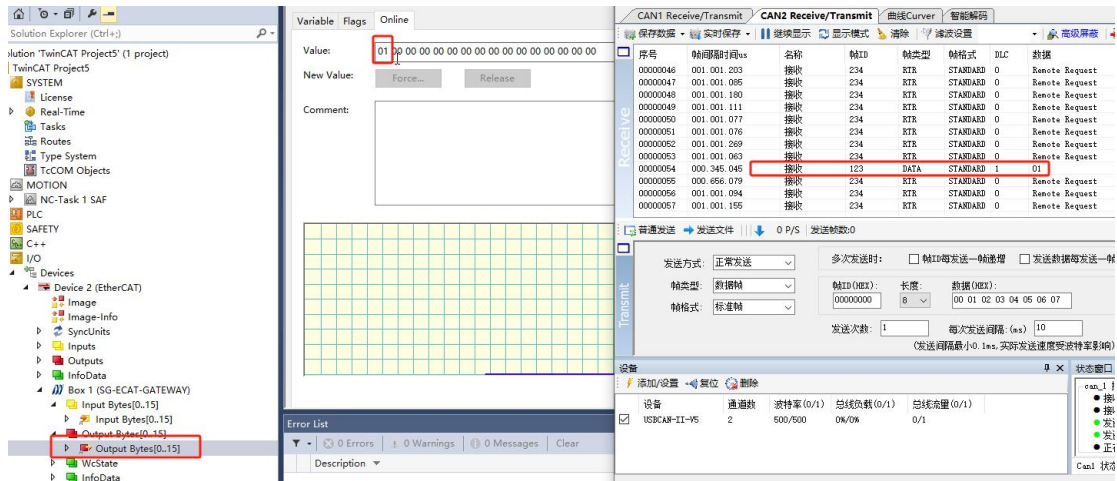
在下面这个对话框选择“是”



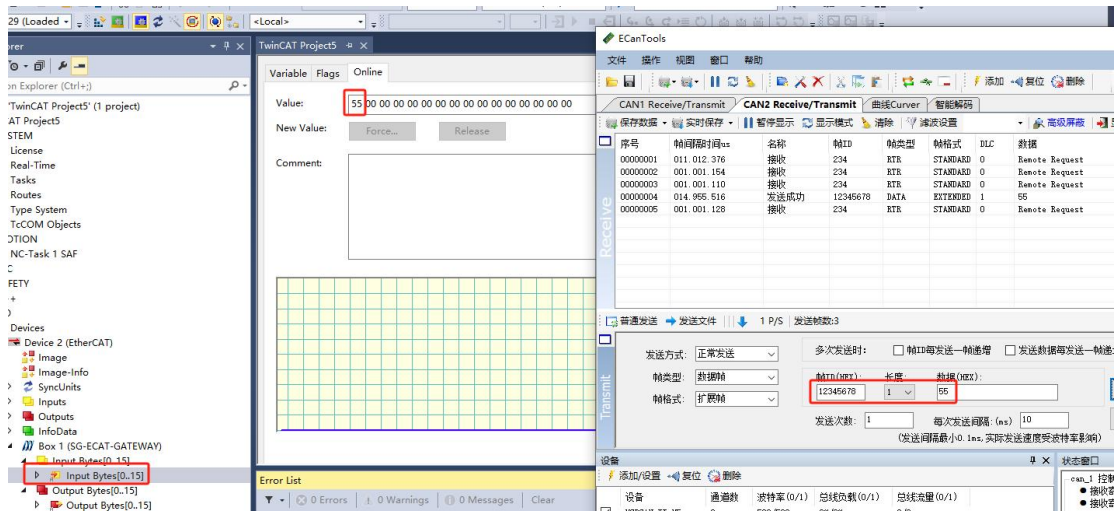
在下面这个对话框选择“是”



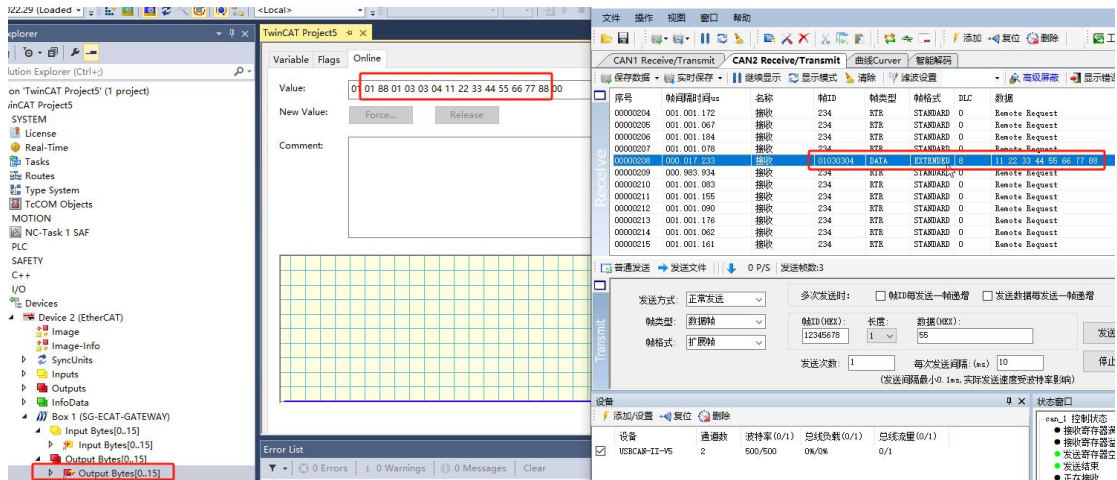
③在 Input Bytes、Output Bytes 可以查看、写入数据

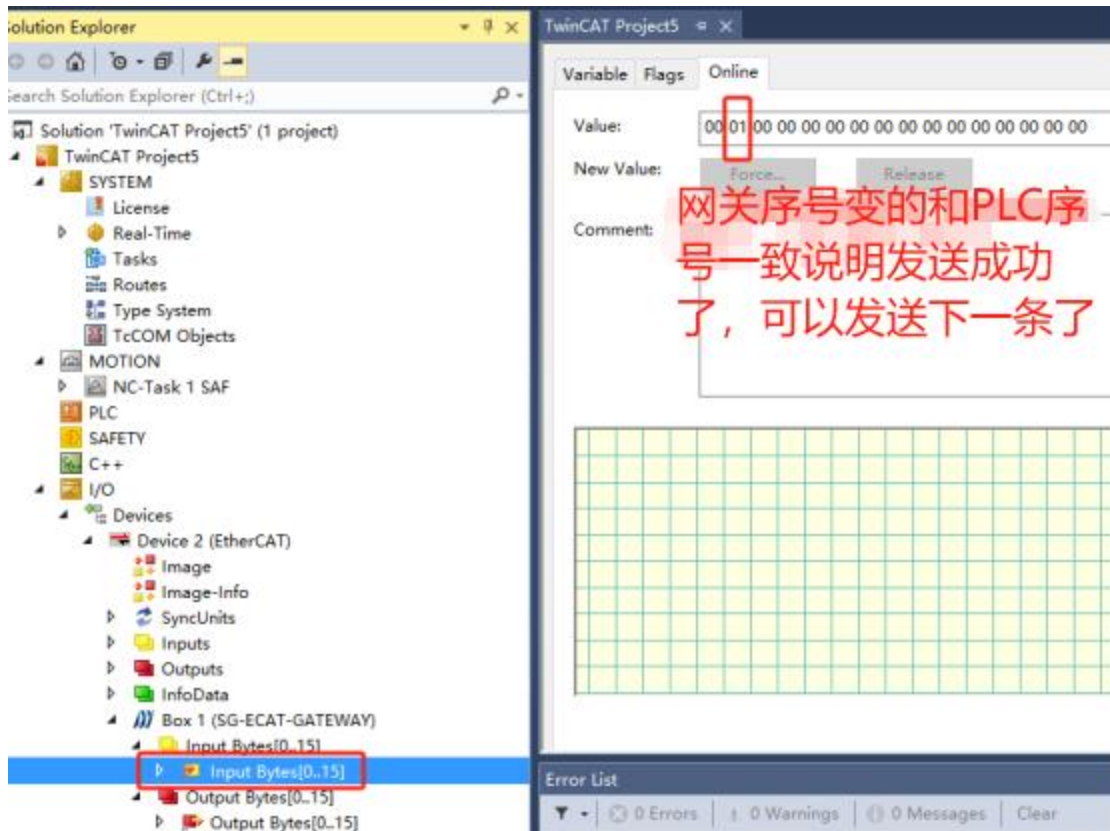


上图可以看出网关在周期发送 ID 为 0x234 的远程帧，Output Bytes[0]改变时发送了一帧 ID 为 0x123 的数据帧

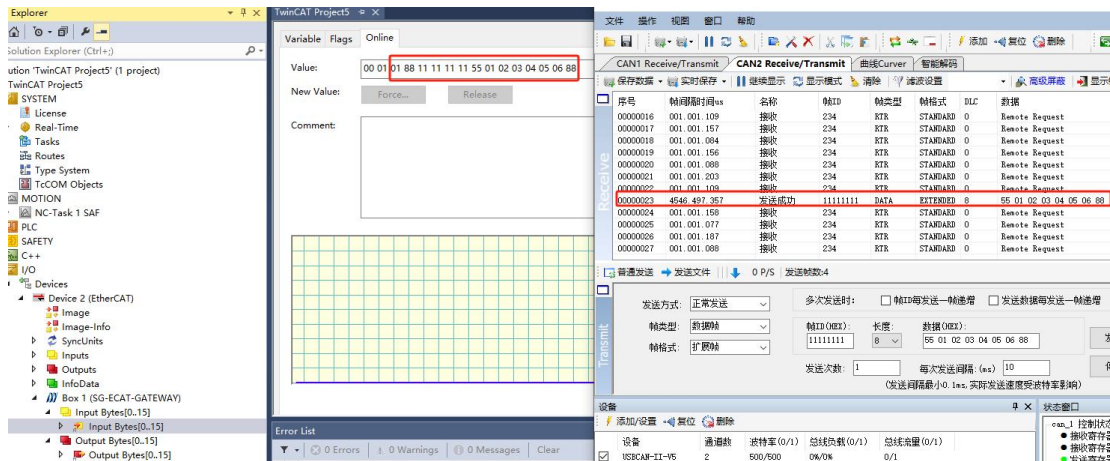


上图可以看到 CAN 分析仪发送 ID 为 0x12345678 的扩展帧时 ECAT 主机能收到对应数据。

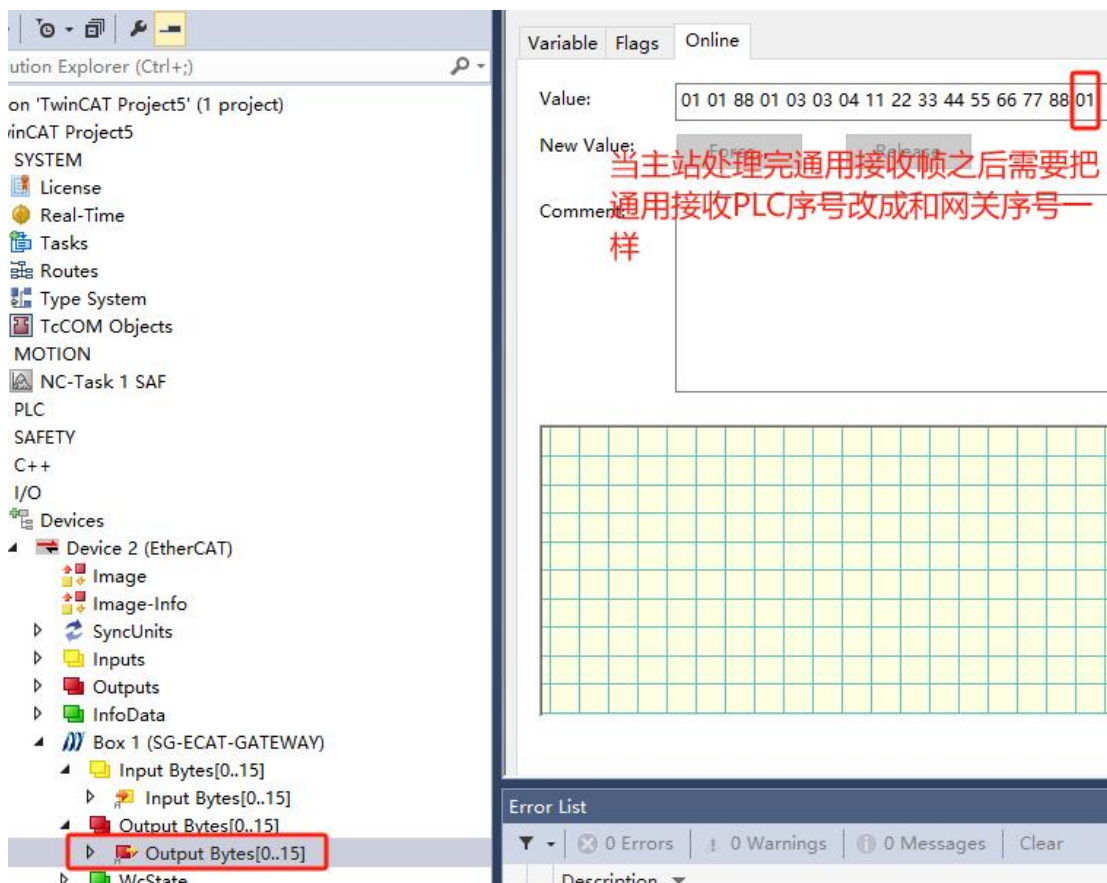




以上两个图可以看到使用通用发送发送了一帧数据，同时网关序号也变到和PLC 序号一致。



上图可以看到 CAN 分析仪发送了一条不在“按 ID 接收”CAN 列表的帧，网关把这个帧放到了通用 CAN 接收，如果 ECAT 主站已经确认了这一帧，需要把该通用接收的 PLC 序号改成和网关序号一致：



6.3 CODESYS 例程

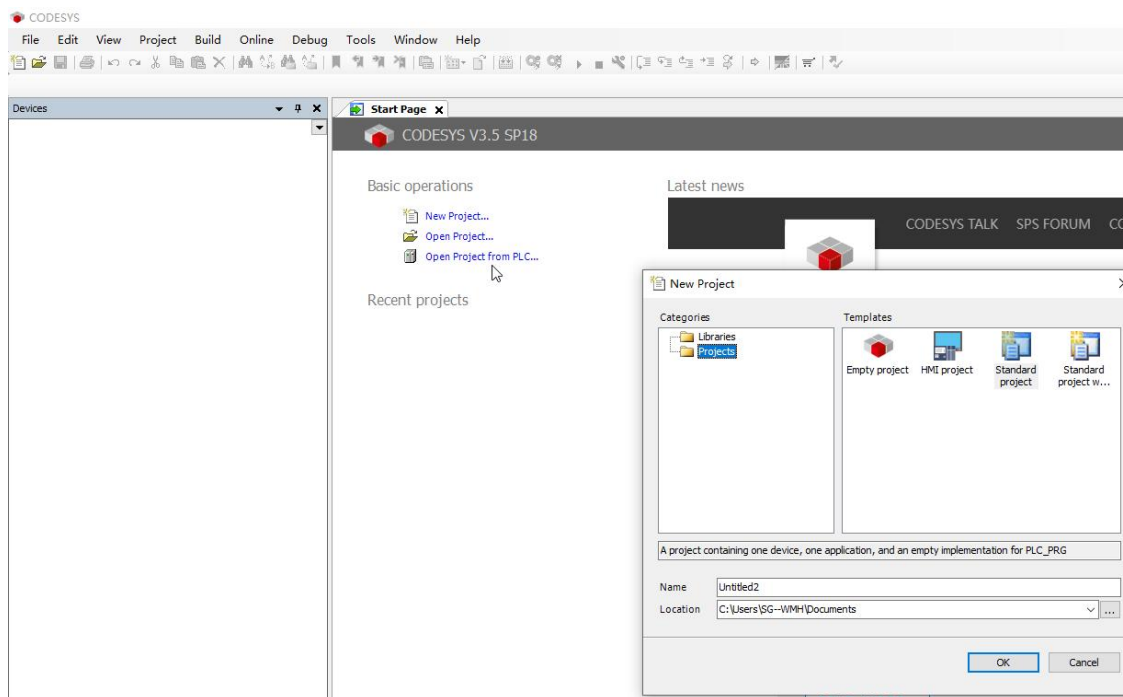
PS: CODESYS 对网卡型号有要求，要确保所使用的网卡支持 CODESYS。

先按 6.1 设置网关。

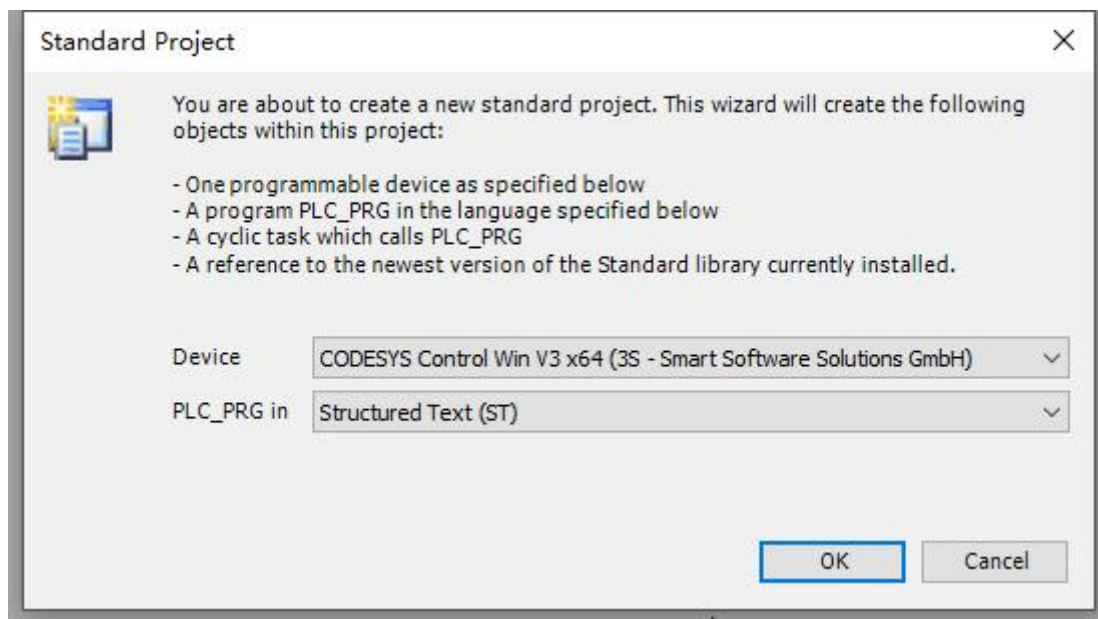
①去我司网站下载设备的 ESI 文件。

连接电脑和网关，注意电脑连接网关的 Port1（ECAT 两个网口区分输入输出，别接反了）。

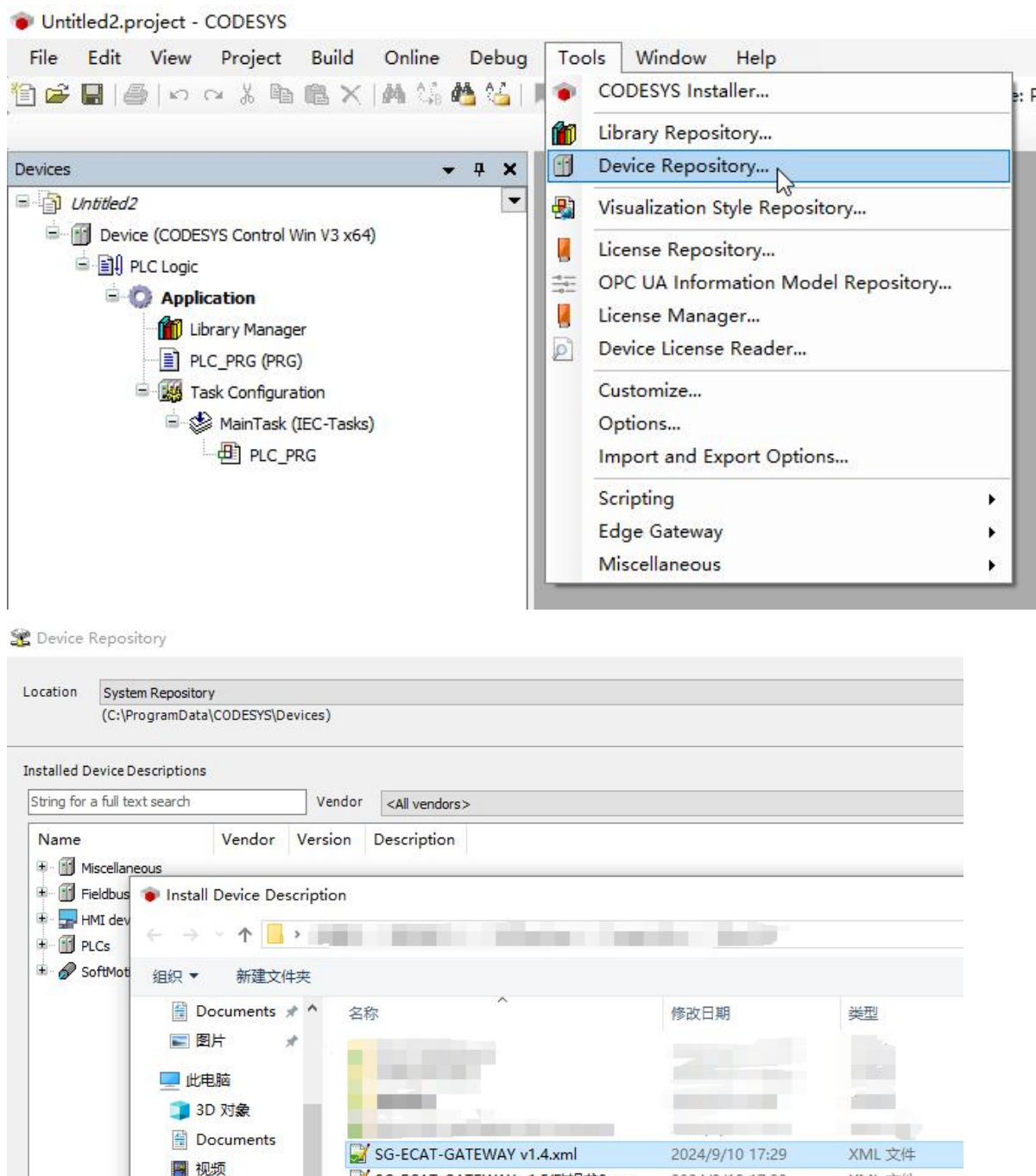
②打开 CODESYS，创建工程



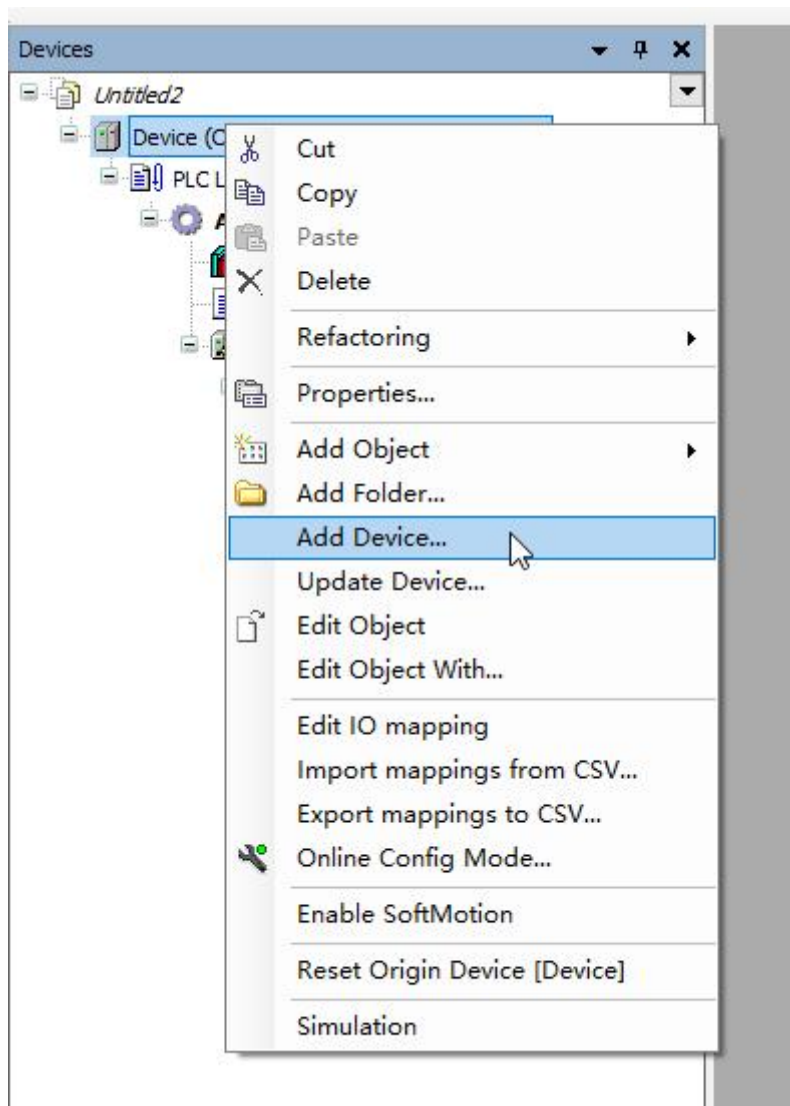
在下面这个对话框选择“OK”

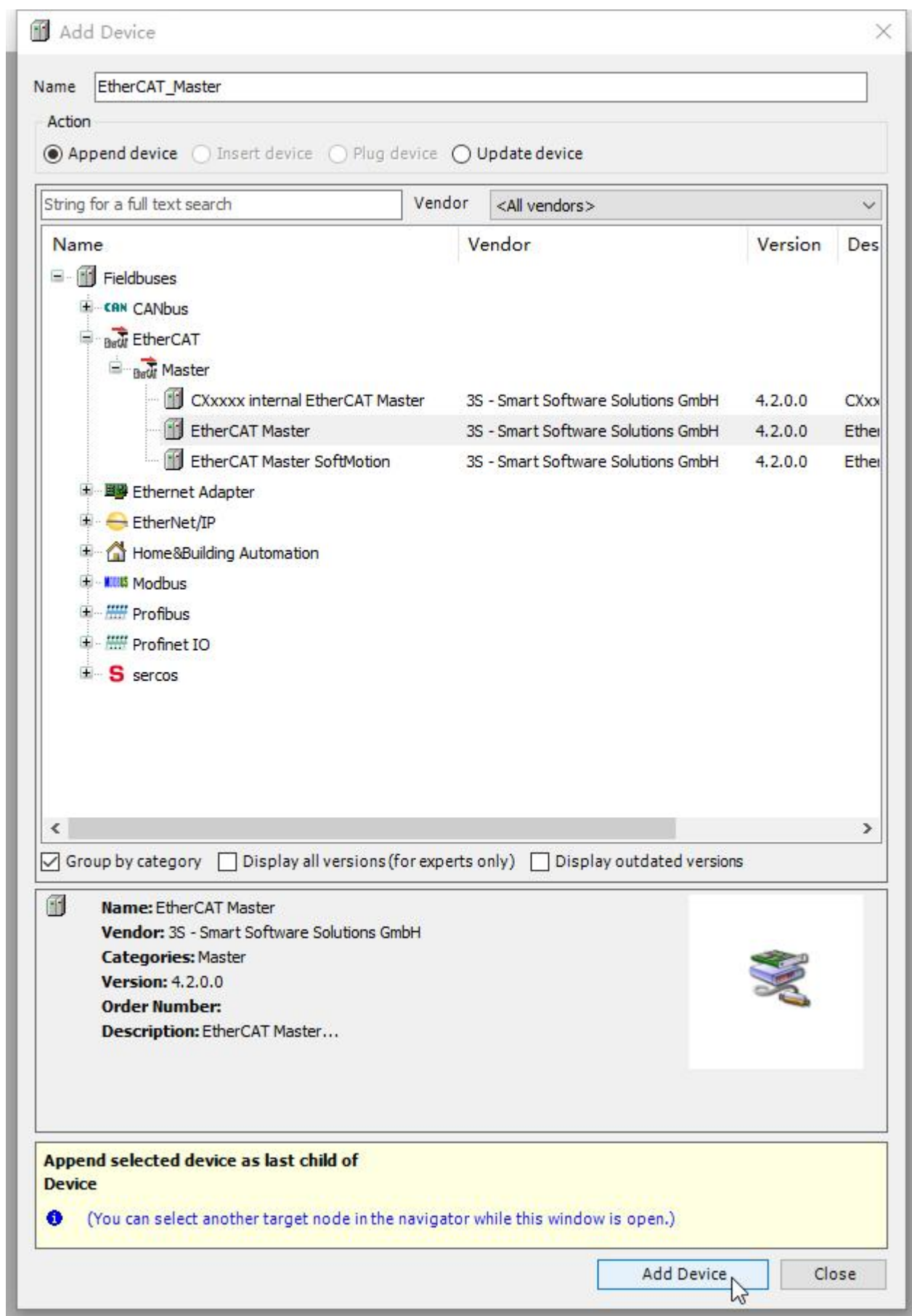


安装 ESI 文件，点击“Tools” ->Device Repository,点击“Install” 选择 ESI 文件并安装。



③工程目录右击“Device”选择“Add Device”,添加一个 EtherCAT 主站。



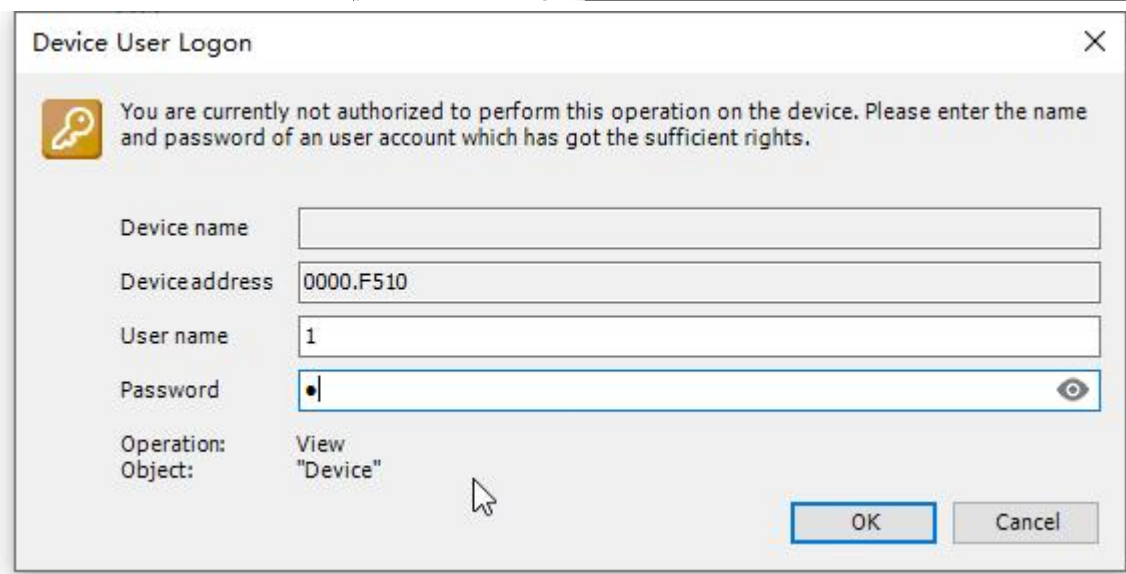
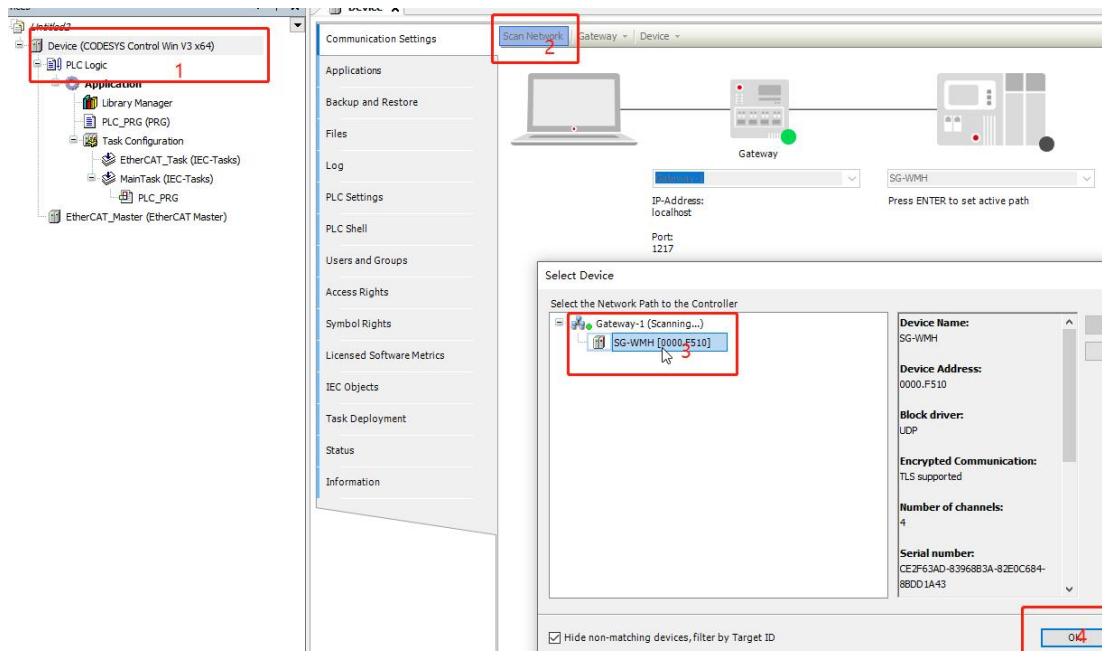


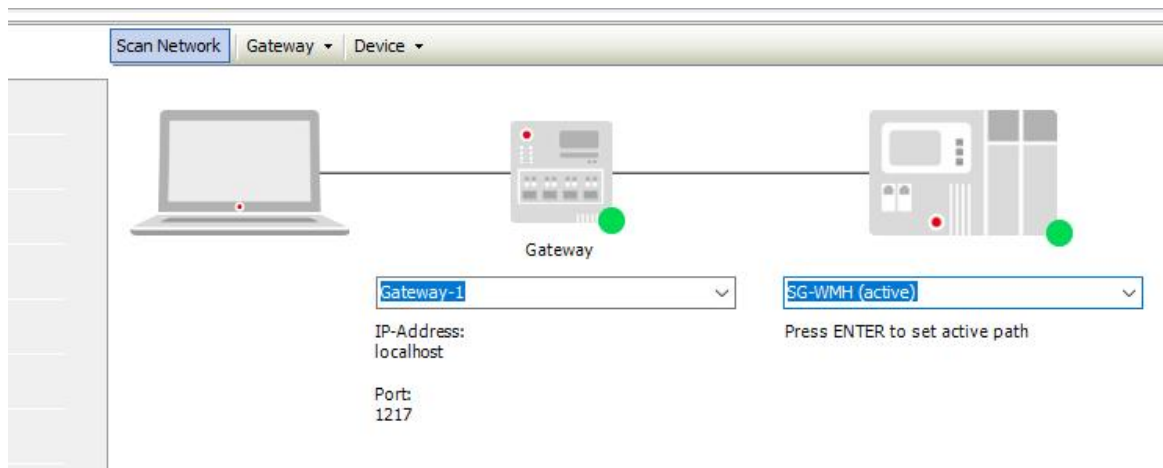
④启动 CODESYS 自带的 Gateway 和 PLC

确保任务栏的这两个图标处于运行状态。



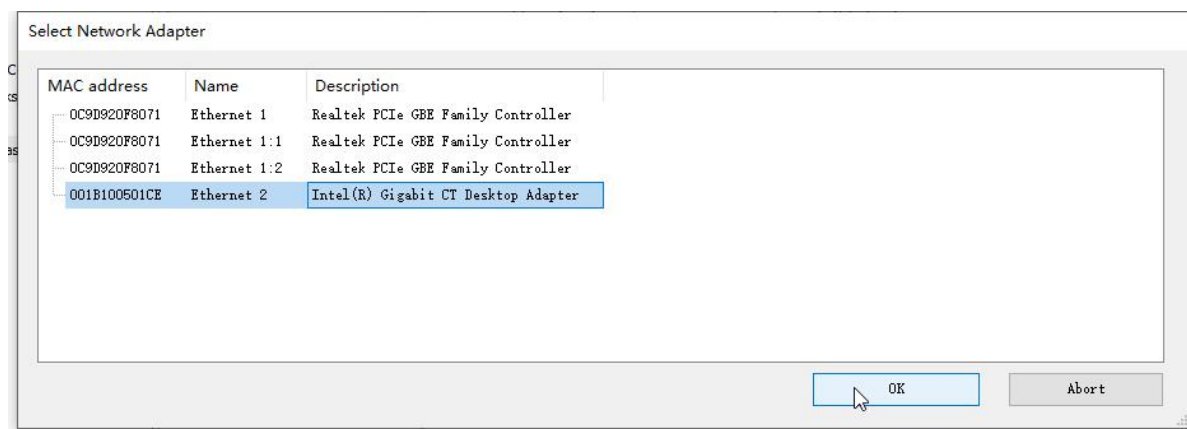
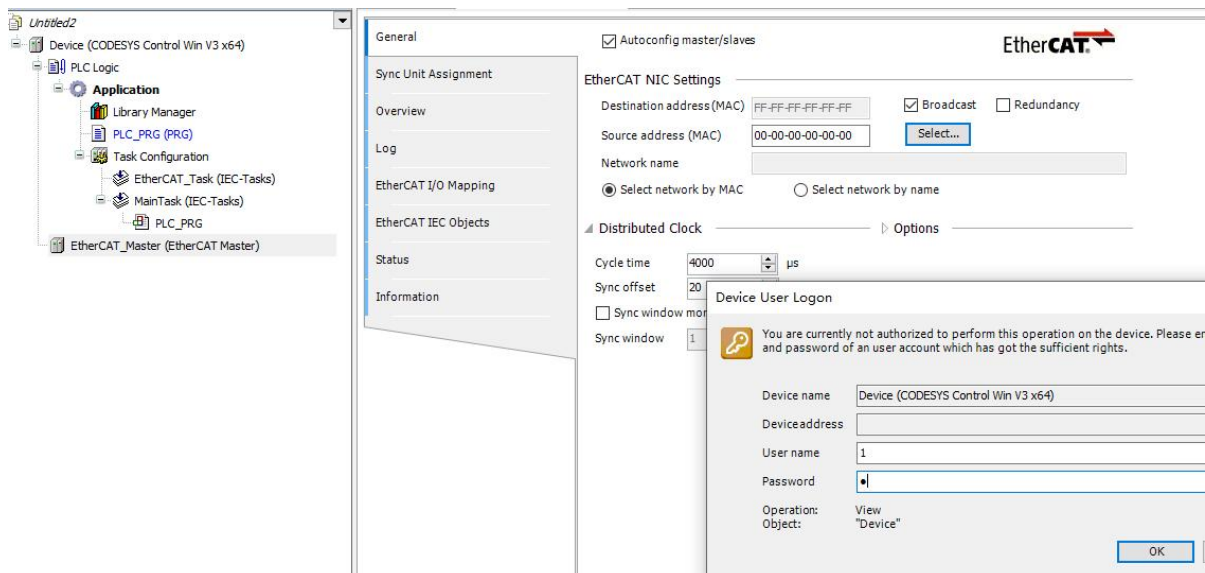
右击“Device”连接网关和 PLC，如下图所示



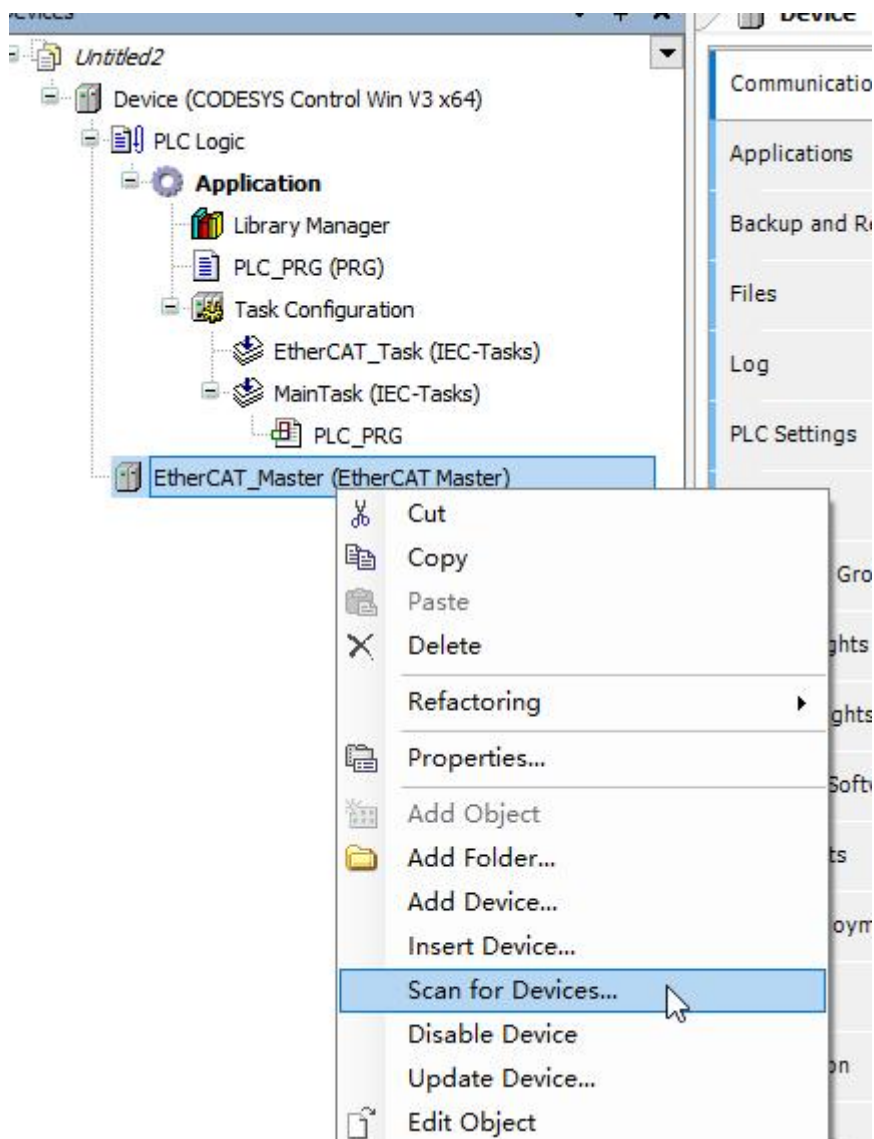


连接成功如上图所示。

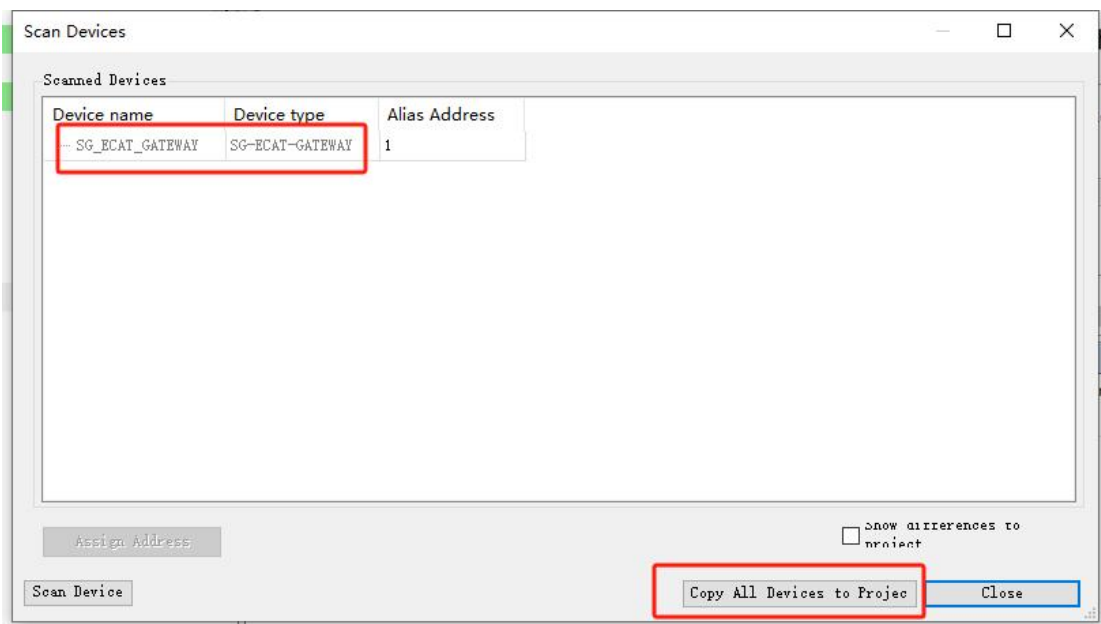
⑤给 EtherCAT 主站选择网卡，双击 EtherCAT 主站，在 General 页面选择网卡。



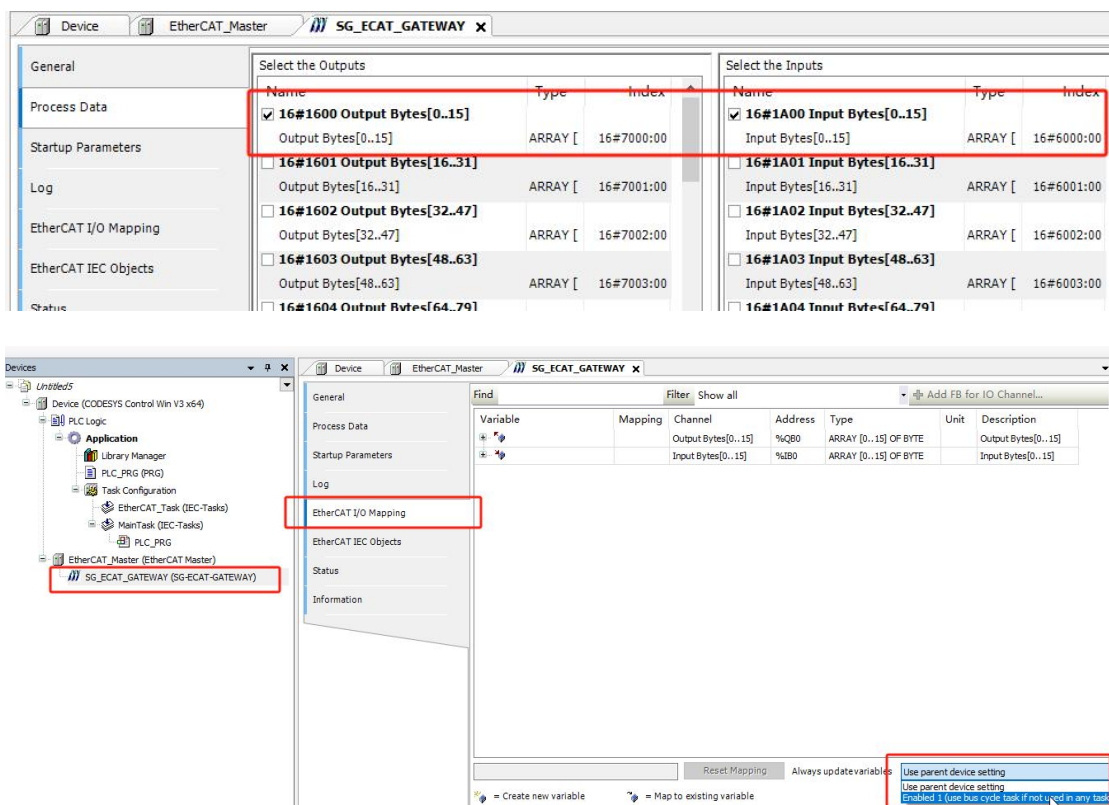
⑥右击工程目录主站点击“Scan for Devices”



弹出如下页面



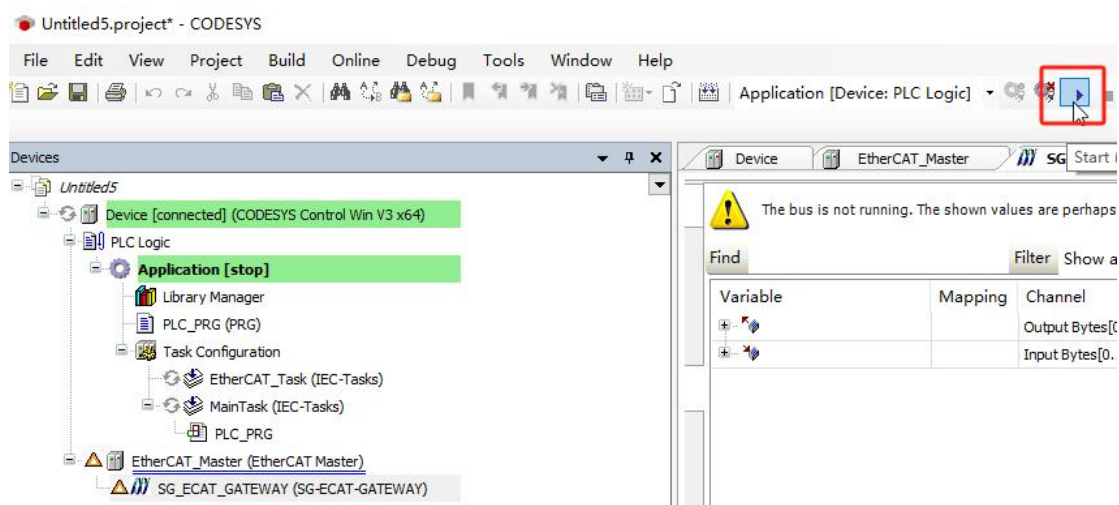
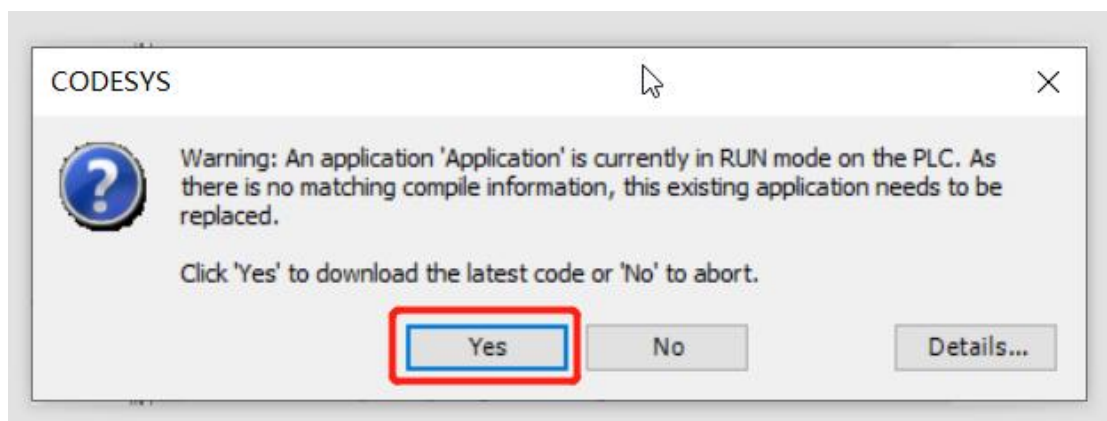
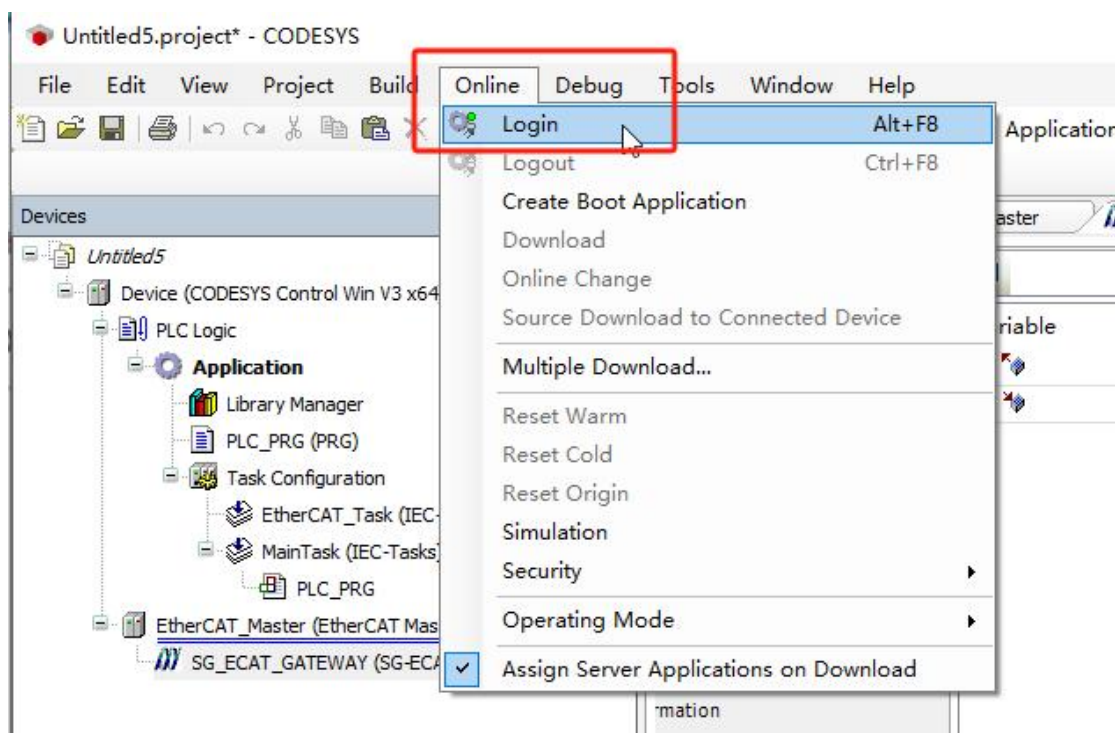
⑦在 Process Data 选择 PDO



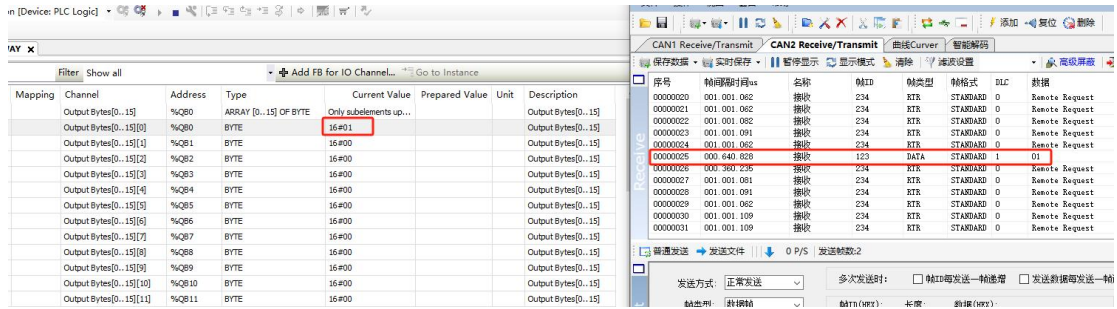
编译一下



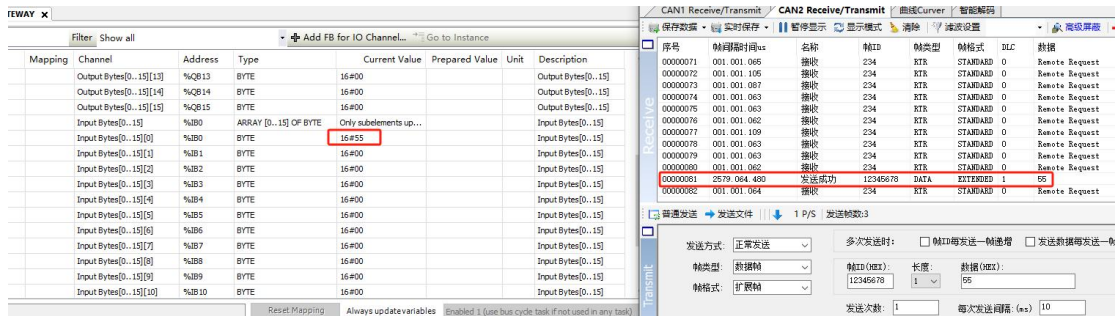
⑧登录、下载、运行



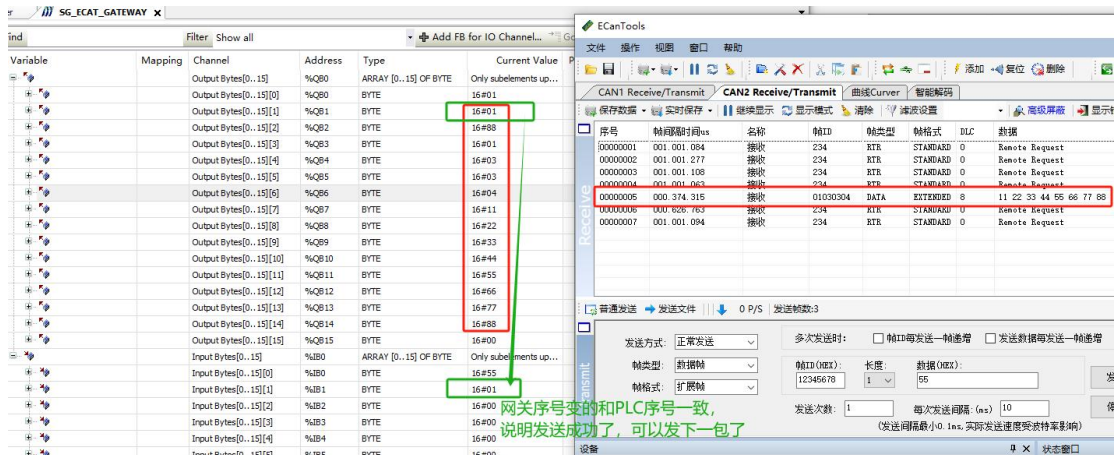
⑨在 EtherCAT IO Mapping 查看写入数据



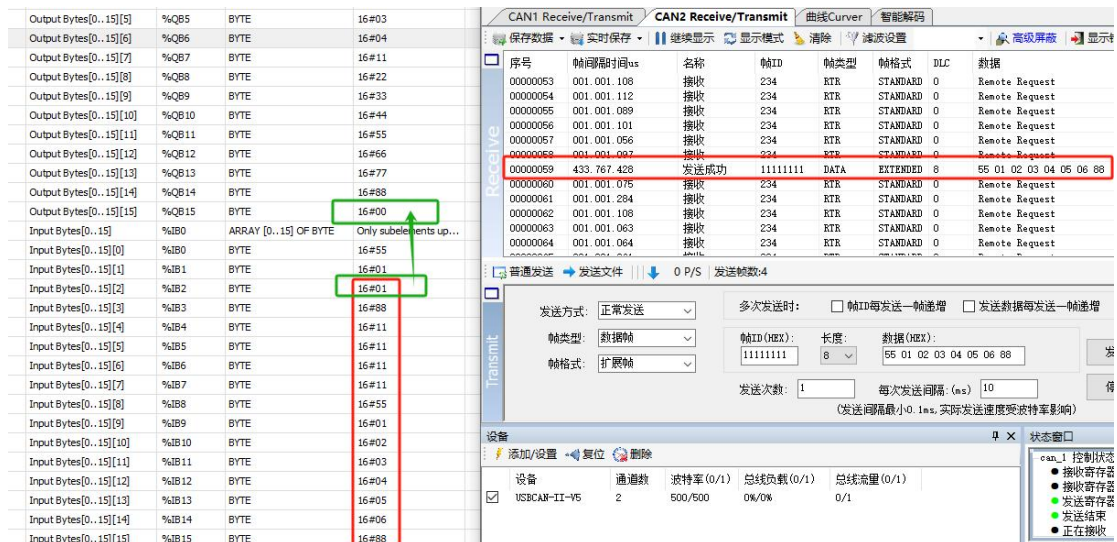
上图可以看出网关在周期发送 ID 为 0x234 的远程帧，Output Bytes[0]改变时发送了一帧 ID 为 0x123 的数据帧。



上图可以看到 CAN 分析仪发送 ID 为 0x12345678 的扩展帧时 ECAT 主机能收到对应数据。



以上两个图可以看到使用通用发送发送了一帧数据，同时网关序号也变到和 PLC 一致。



上图可以看到 CAN 分析仪发送了一条不在“按 ID 接收”CAN 列表的帧，网关把这个帧放到了通用 CAN 接收，如果 ECAT 主站已经确认了这一帧，需要把该通用接收的 PLC 序号改成和网关序号一致。

6.4 欧姆龙 Sysmac Studio 例程

先按 6.1 设置网关。

①首先去我司网站下载设备的 ESI 文件。

连接电脑、欧姆龙 PLC 和网关，注意 PLC 的 EtherCAT 口接网关的 Port1（ECAT 两个网口区分输入输出，别接反了）。

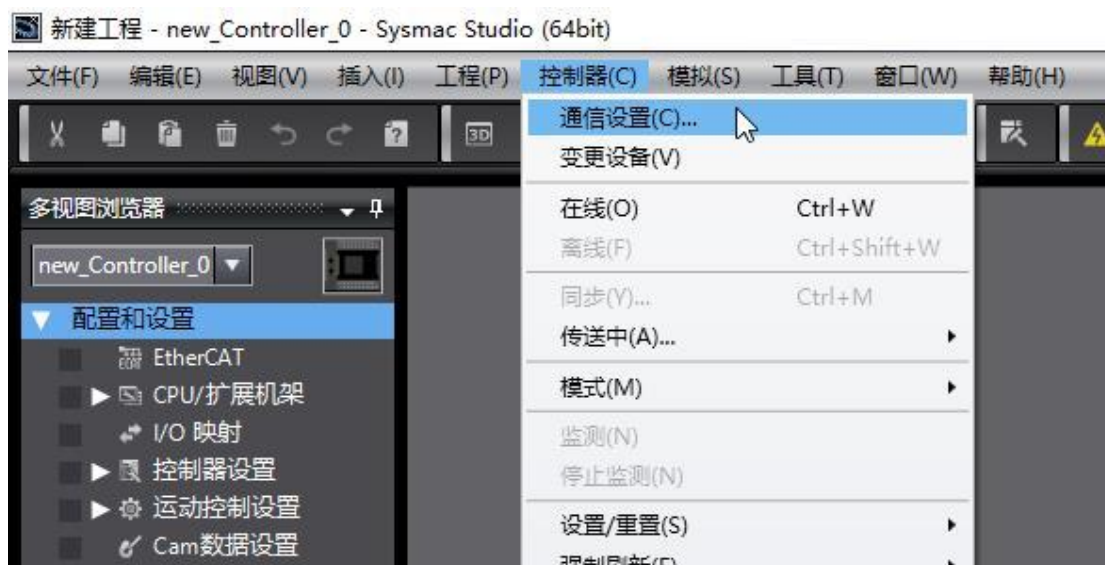
②打开 Sysmac Studio，创建工程



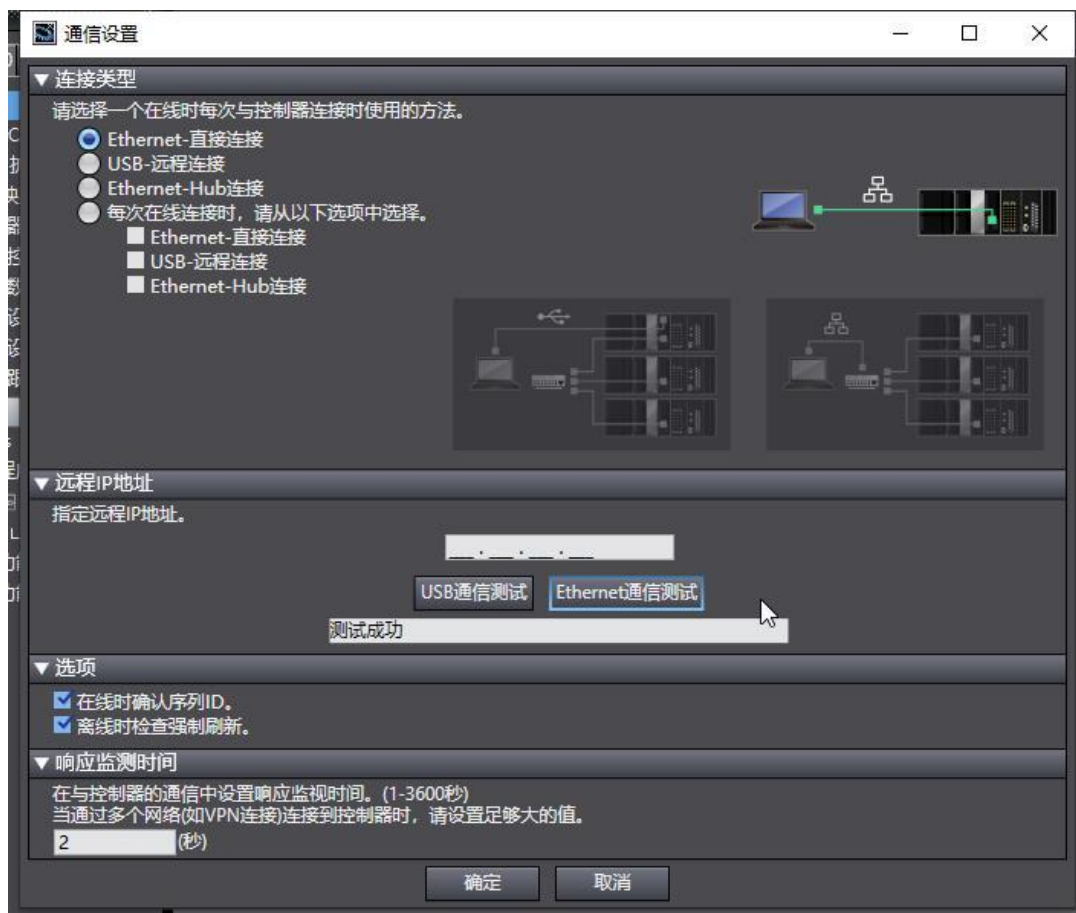
根据实际 PLC 型号和版本创建工程



通信测试，点击“控制器”->“通信设置”

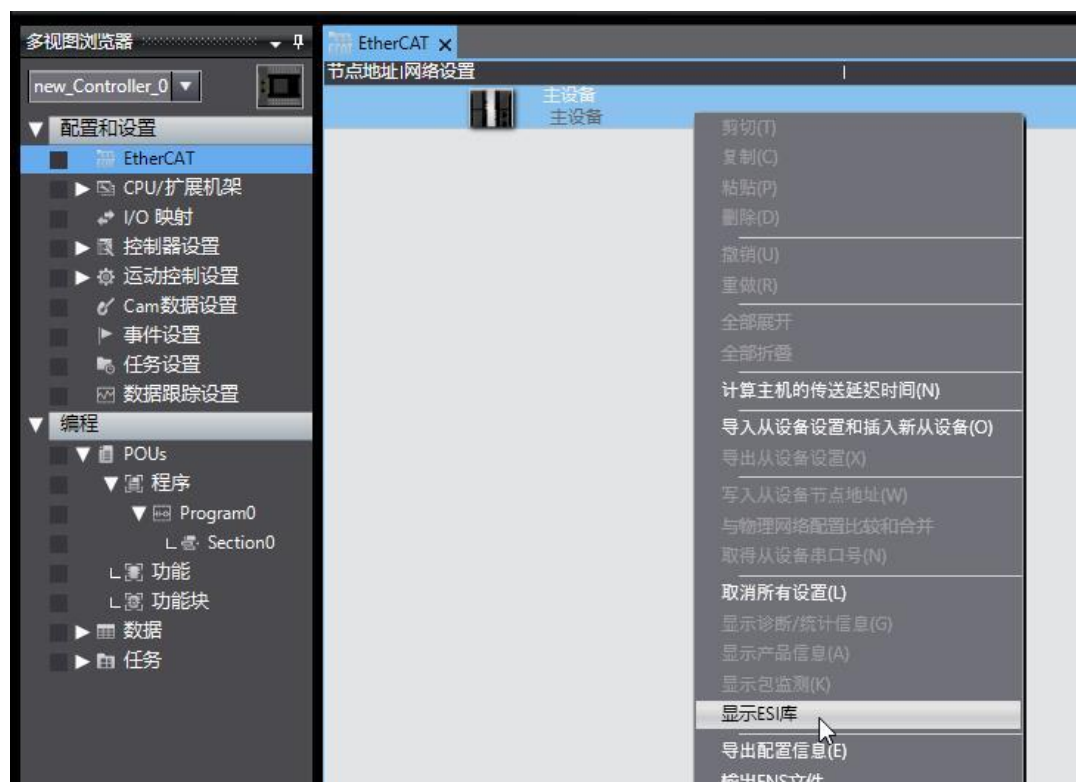


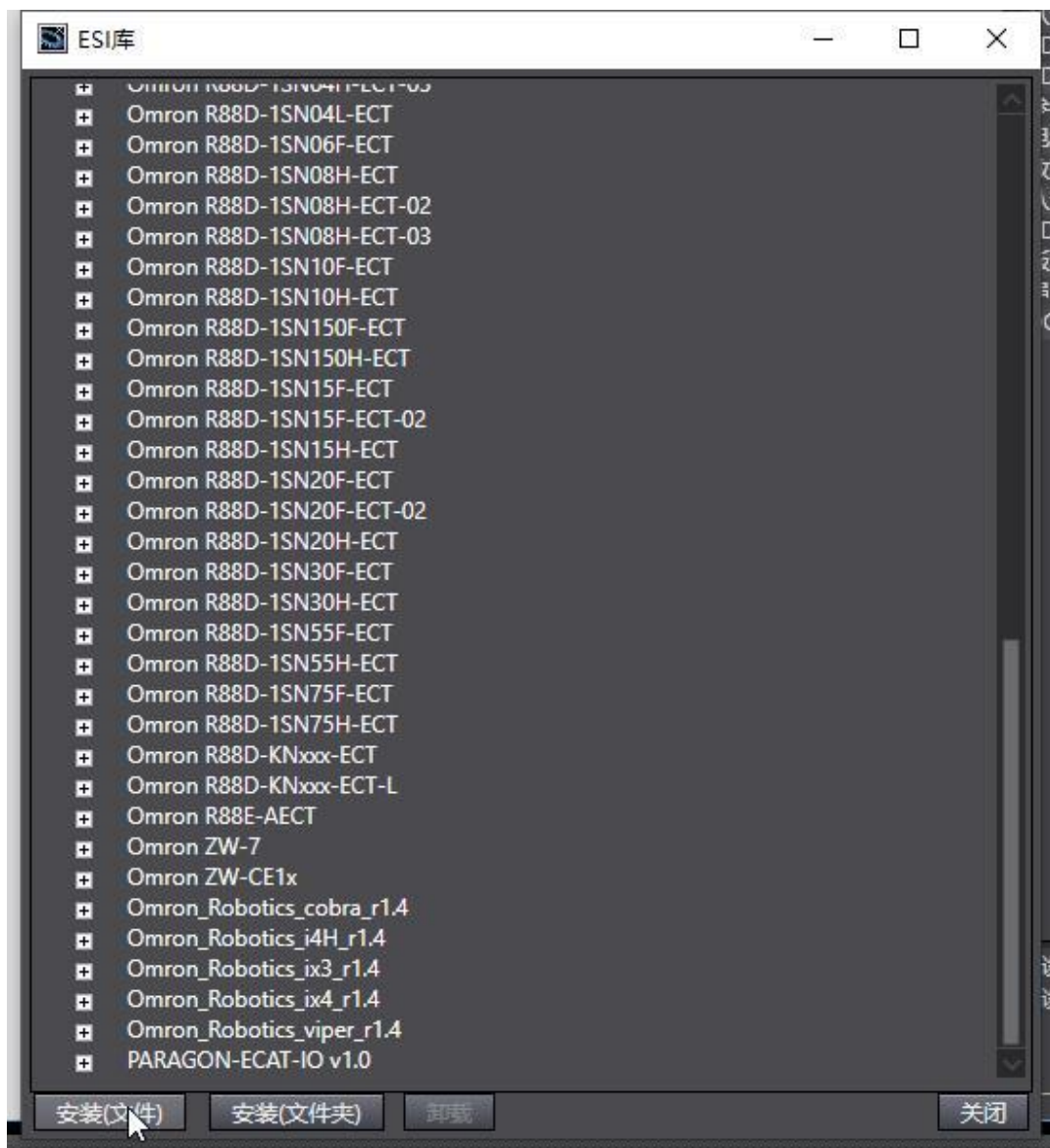
确保在此之前已经将电脑 IP 设置为 192.168.250.xx 网段，这是 PLC 要求的



点击 Ethernet 通信测试，显示测试成功，说明网络没问题、

③双击工程目录下的 EtherCAT，右击“主设备”选择“显示 ESI 库”

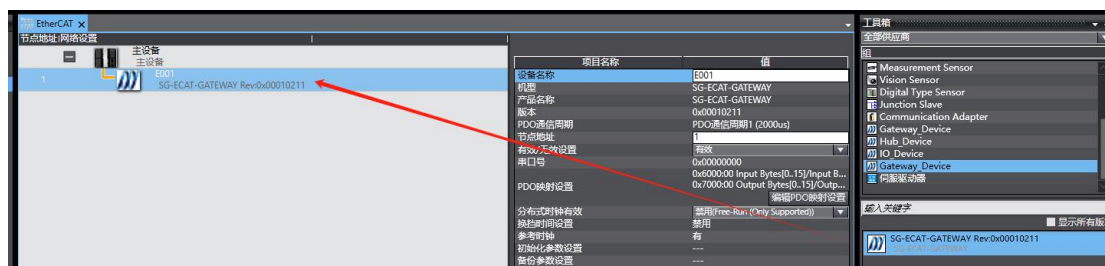




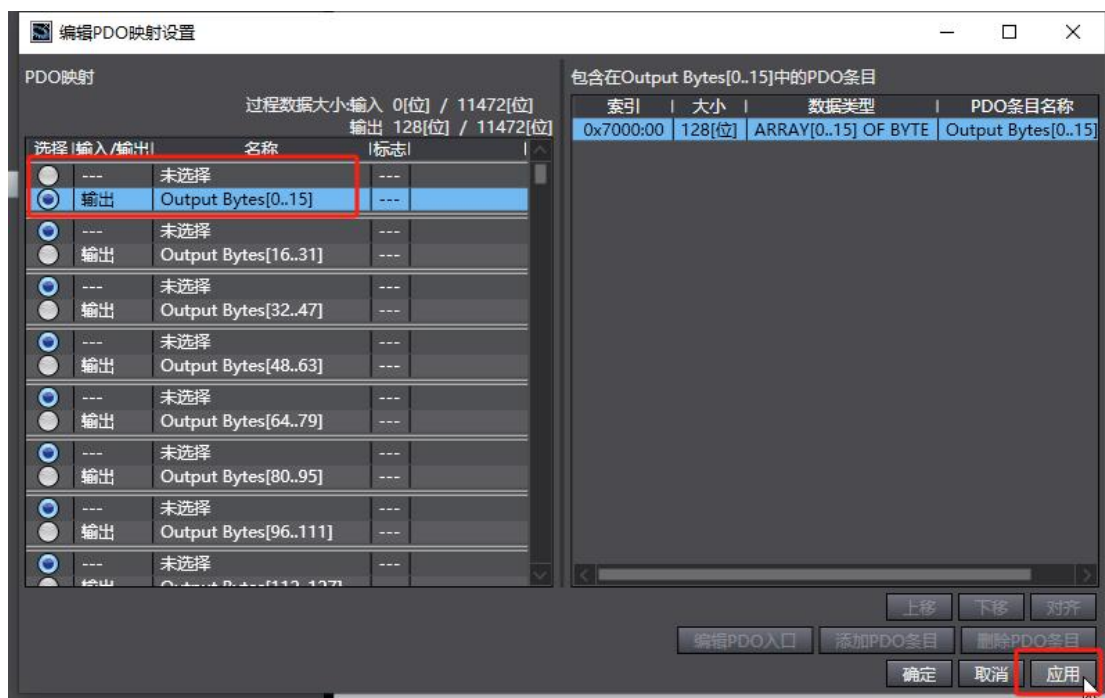
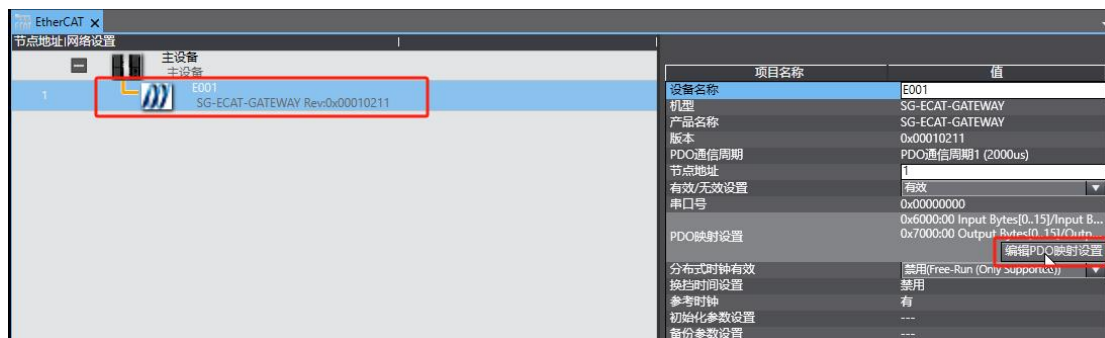
点击“安装（文件）”，选择下载的 ESI 文件并安装

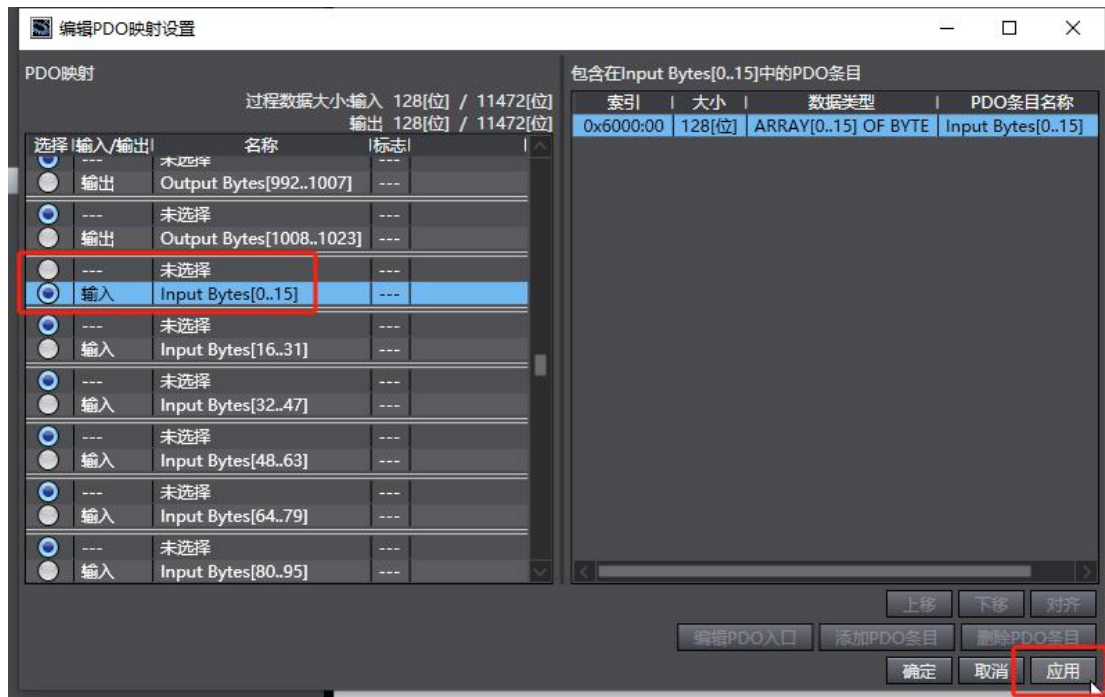


④在主站下添加网关，如下图所示，在右上角选择 Tianjin Sange Elc Co.Ltd 然后选择网关并拖动到主站下面。



设置网关 PDO

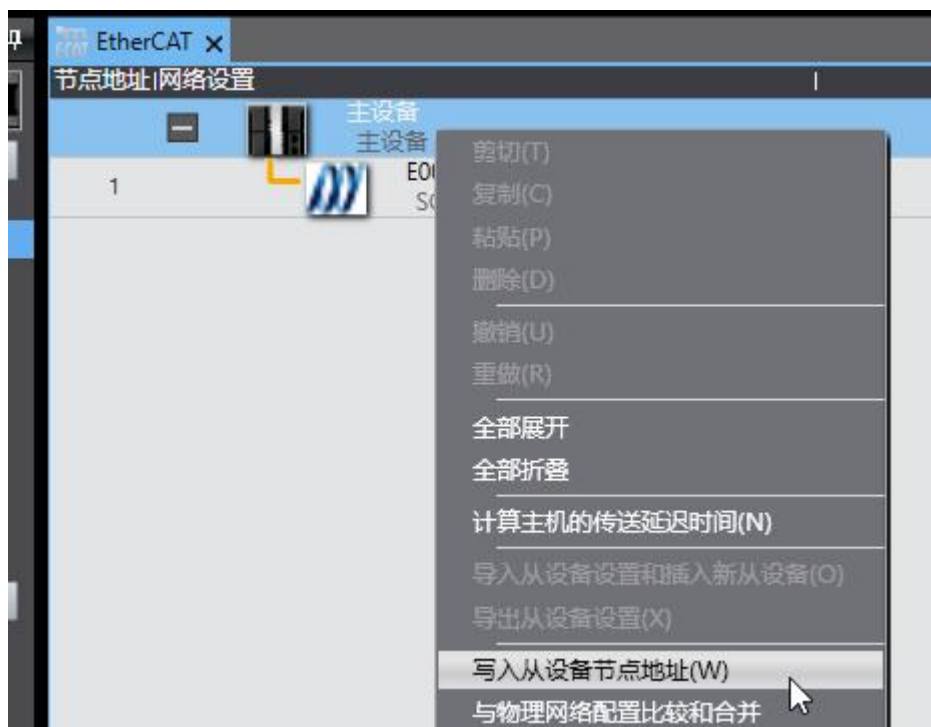




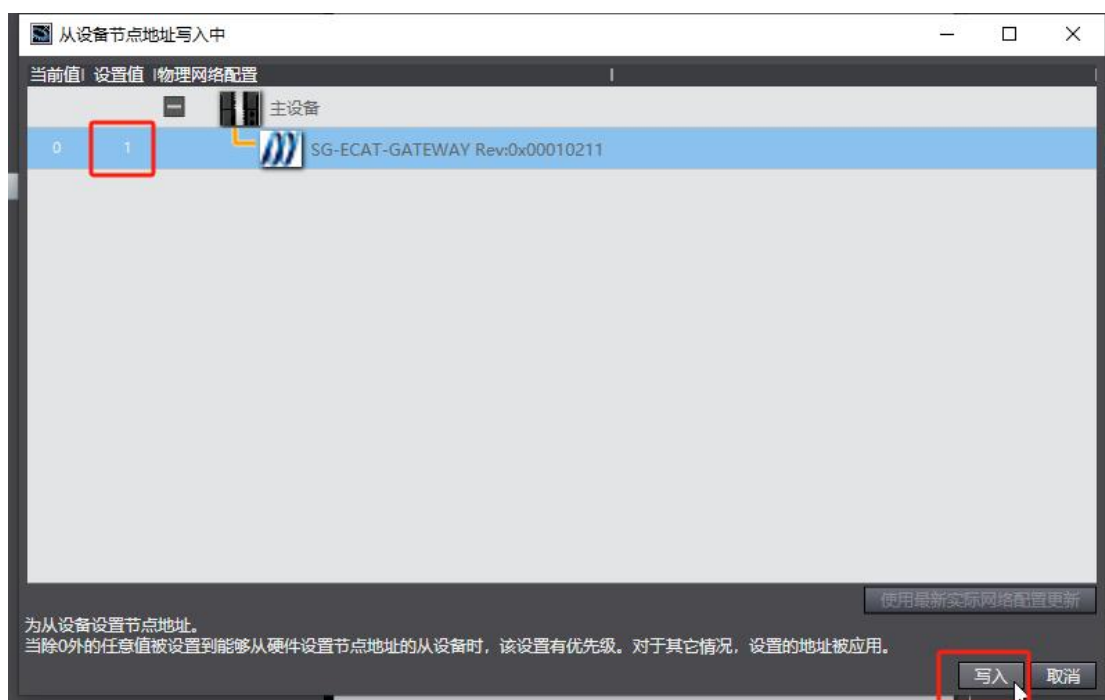
⑤点击控制器，选择“在线”

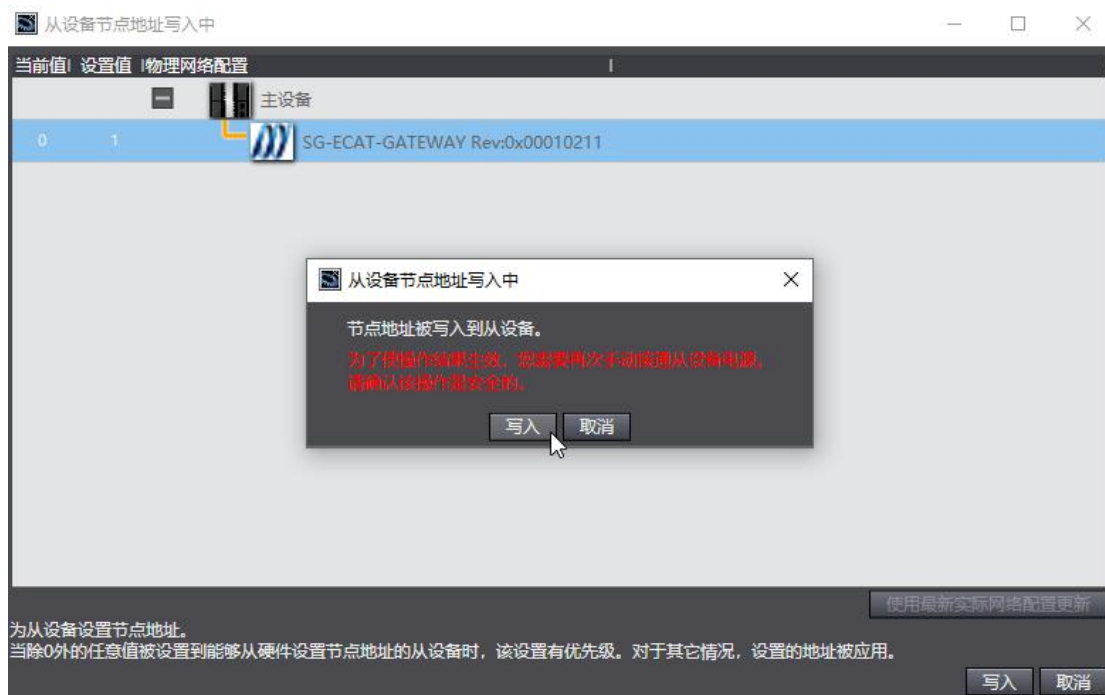


右击主设备，选择“写入从设备节点地址”



如下图，输入从设备节点地址，点击写入

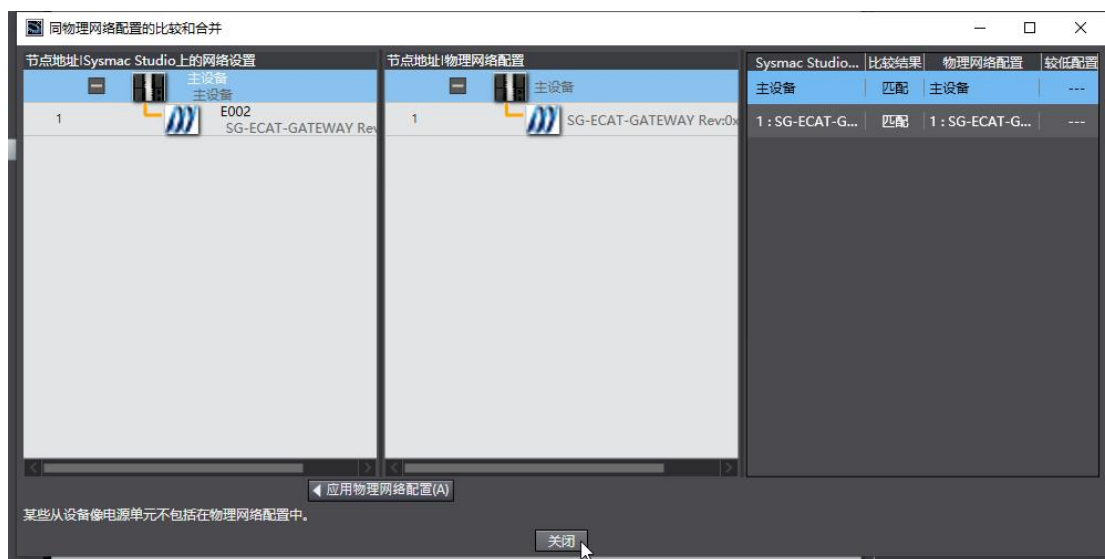




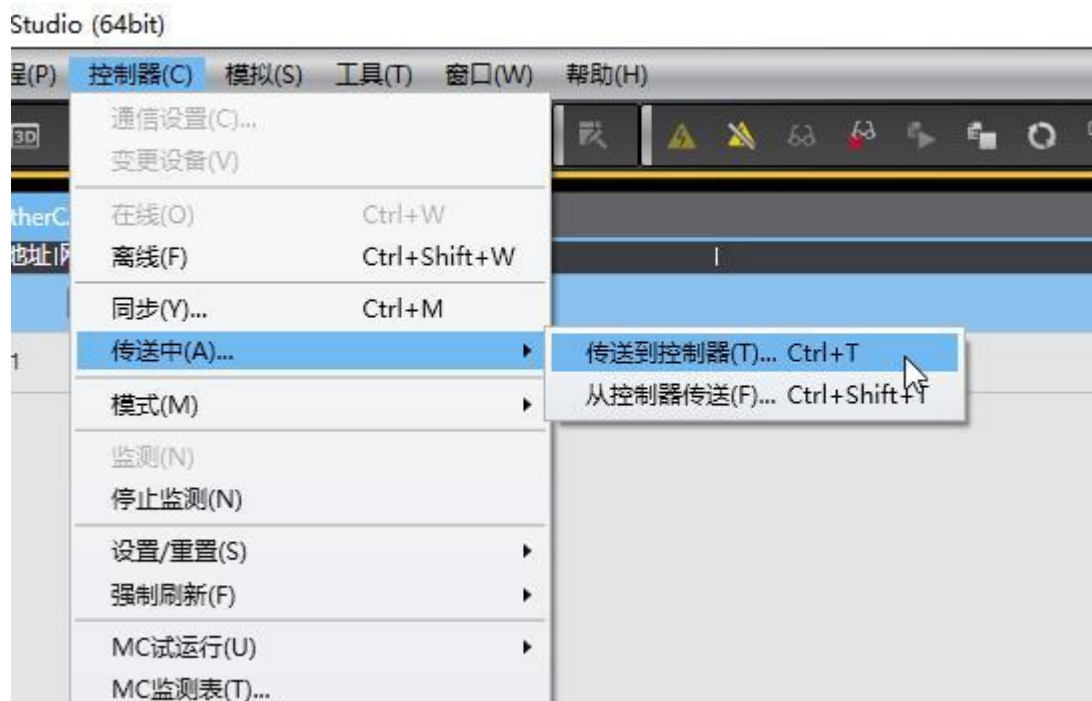
根据提示，给远程 IO 重上电

⑥右击主设备选择“与物理网络比较和合并”





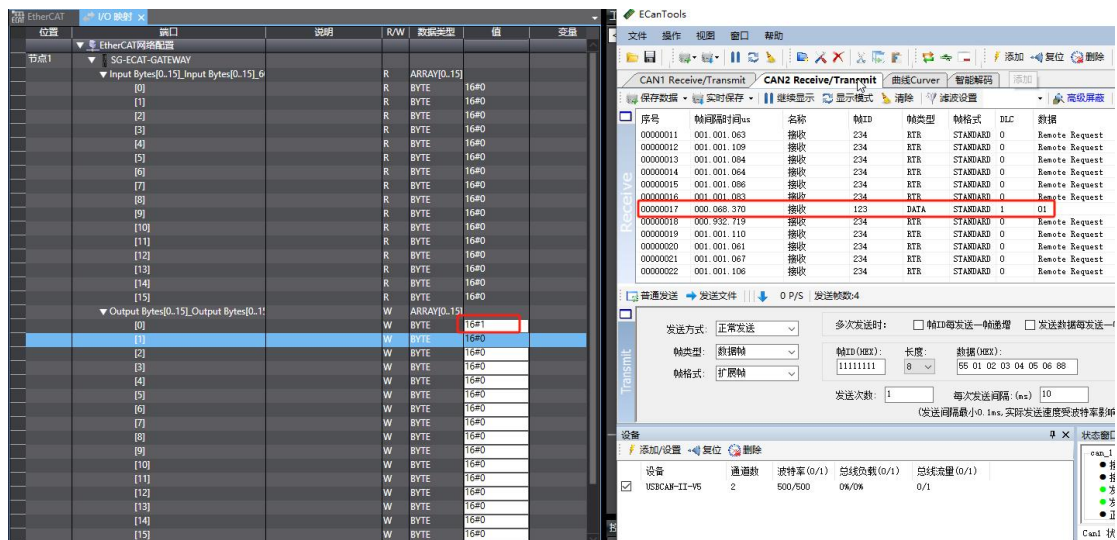
如果弹出如下“匹配”，则没问题，否则检查配置和硬件
 点击控制器“传送中”->“传送到控制器”



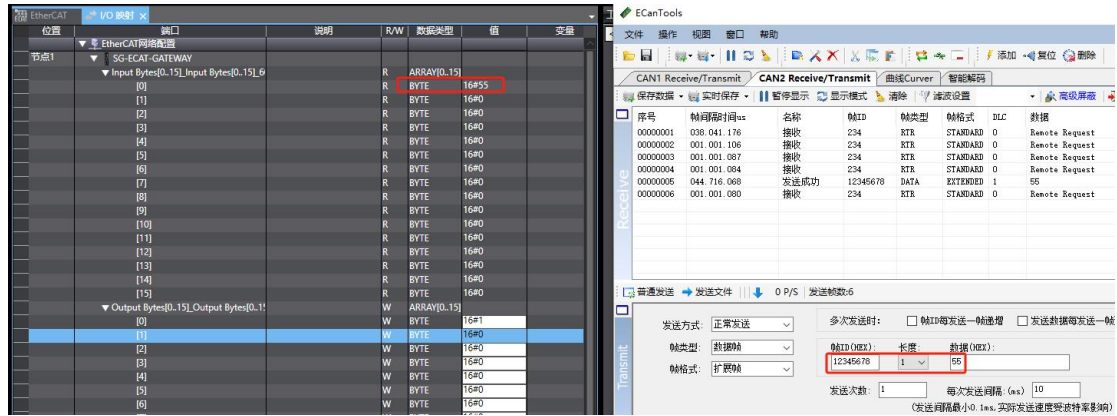


后面一路点确定直到下载完组态。

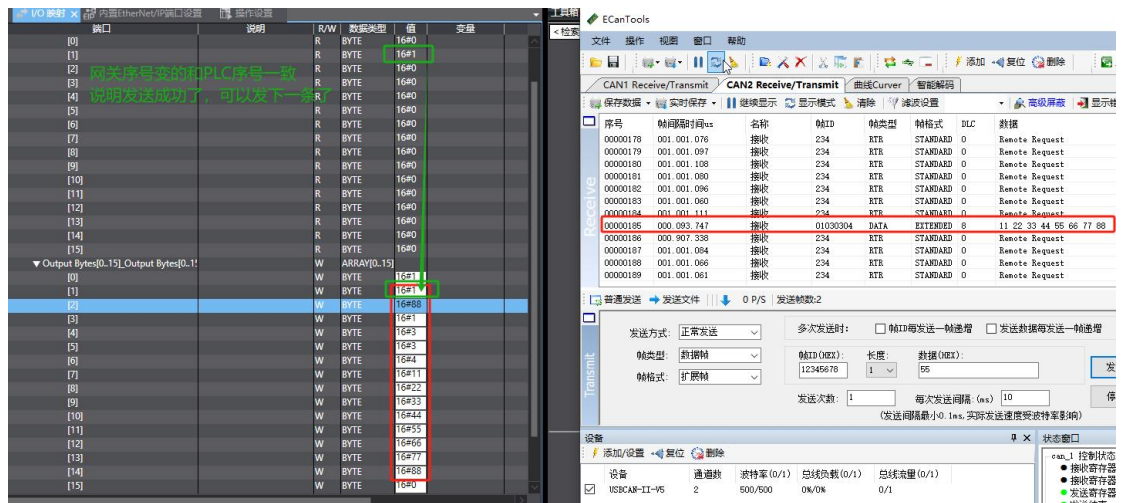
⑦在 I/O 映射里面可以查看、写入数据



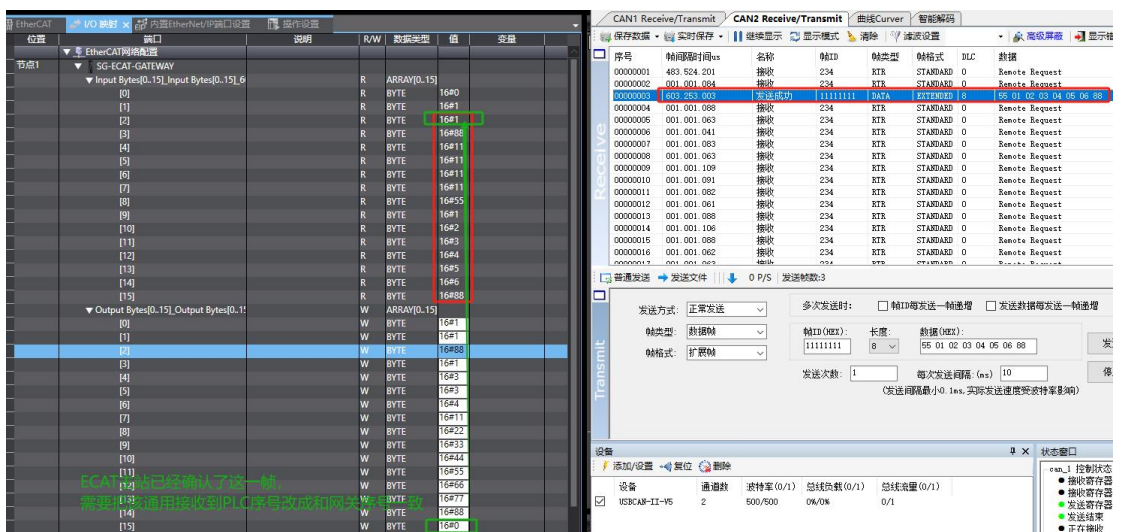
上图可以看出网关在周期发送 ID 为 0x234 的远程帧，Output Bytes[0]改变时发送了一帧 ID 为 0x123 的数据帧。



上图可以看到 CAN 分析仪发送 ID 为 0x12345678 的扩展帧时 ECAT 主机能收到对应数据。



以上两个图可以看到使用通用发送发送了一帧数据，同时网关序号也变到和 PLC 一致。



上图可以看到 CAN 分析仪发送了一条不在“按 ID 接收”CAN 列表的帧，网关把这个帧放到了通用 CAN 接收，如果 ECAT 主站已经确认了这一帧，需要

把该通用接收的 PLC 序号改成和网关序号一致。

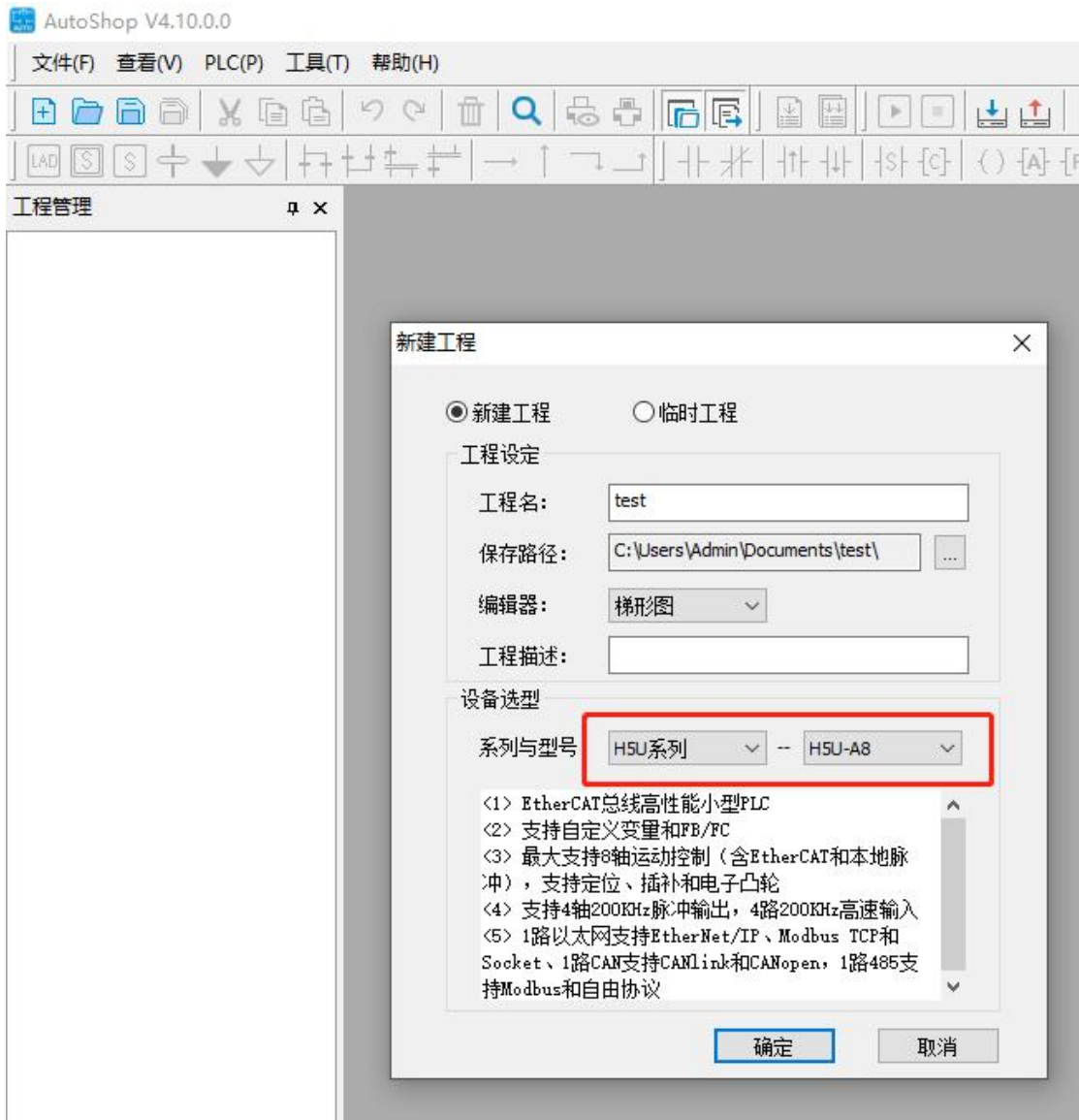
6.5 汇川 H5U-A8 例程

先按 6.1 设置网关。

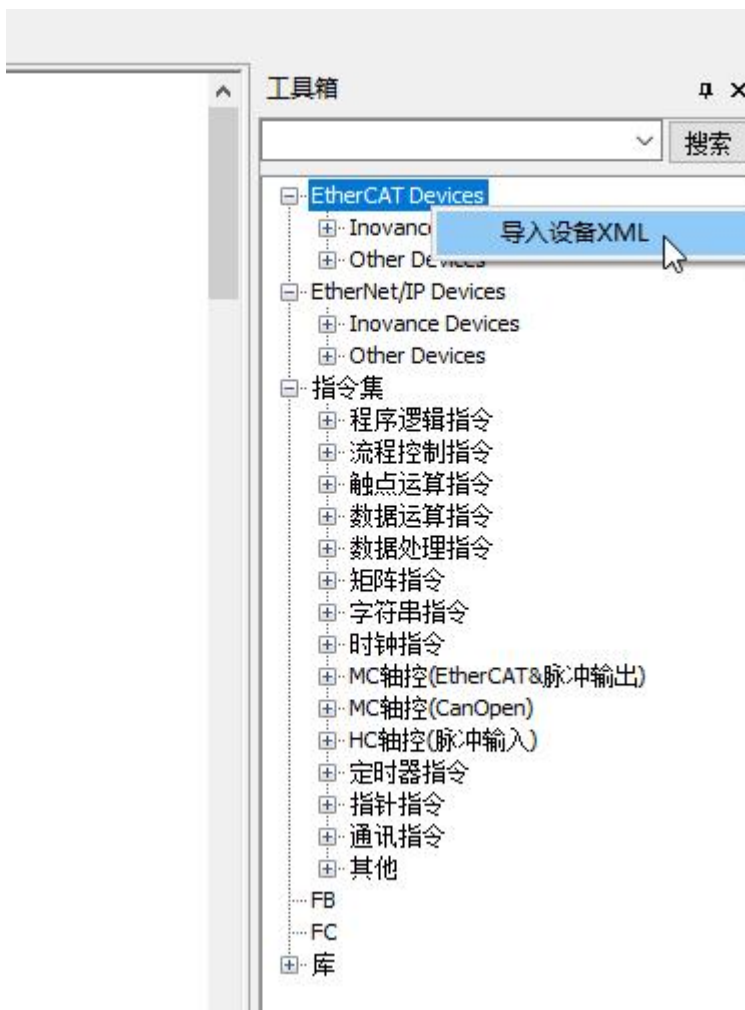
①首先去我司网站下载设备的 ESI 文件。

连接电脑、汇川 H5U-A8 和网关，注意 PLC 的 EtherCAT 口接网关的 Port1（ECAT 两个网口区分输入输出，别接反了）。

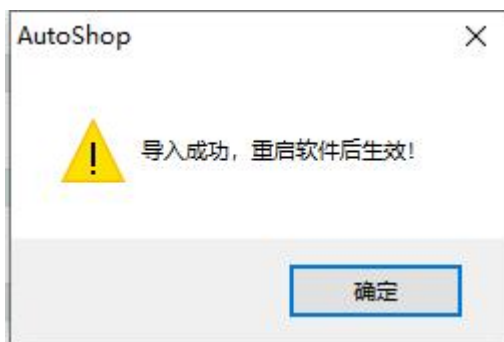
②打开 AutoShop，创建工程，注意 PLC 型号



③安装 XML 文件。

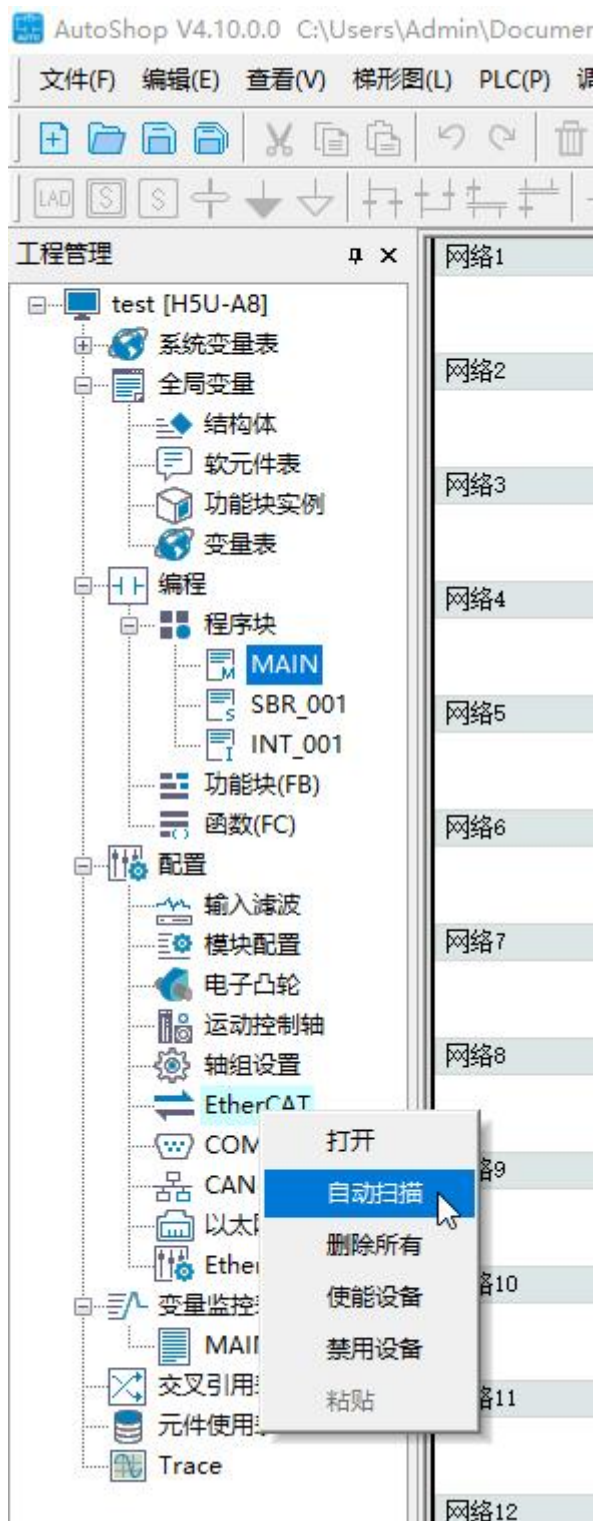


在 XML 存放目录选择 XML 文件



安装完成之后需要重启 AutoShop 并重新建立工程

④在工程管理下右键 EtherCAT，选择“自动扫描”

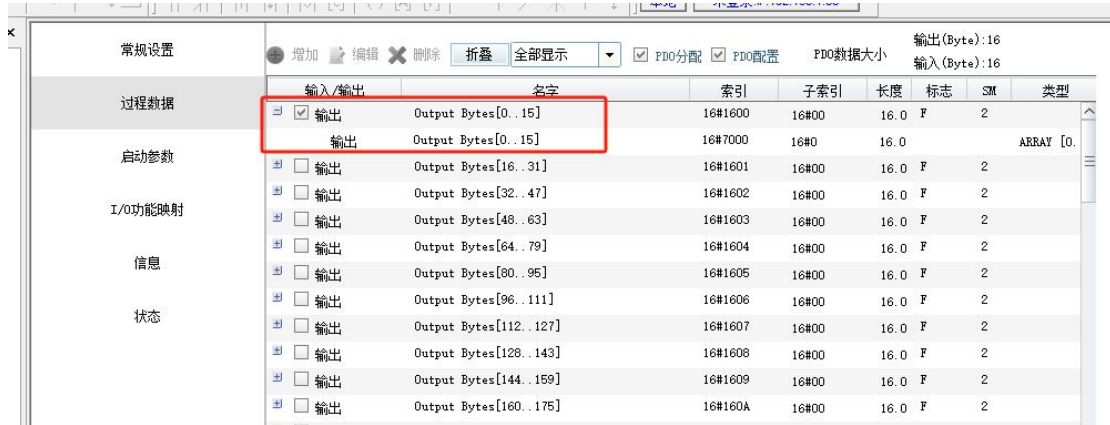




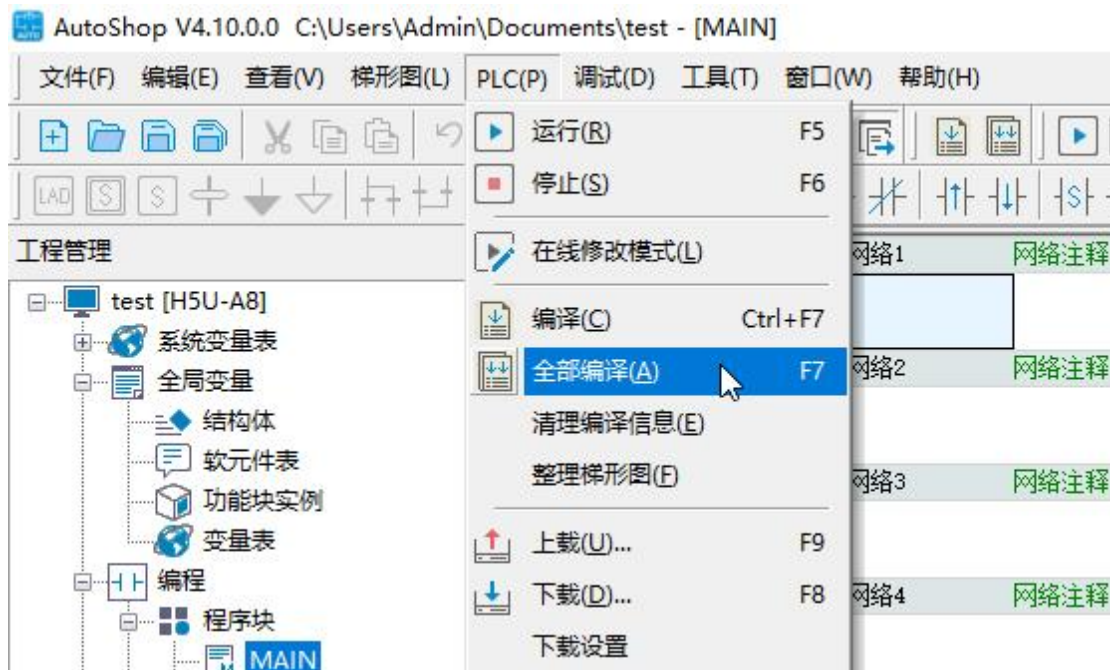
自动扫描列表会出现本公司的网关，之后点击“更新组态”

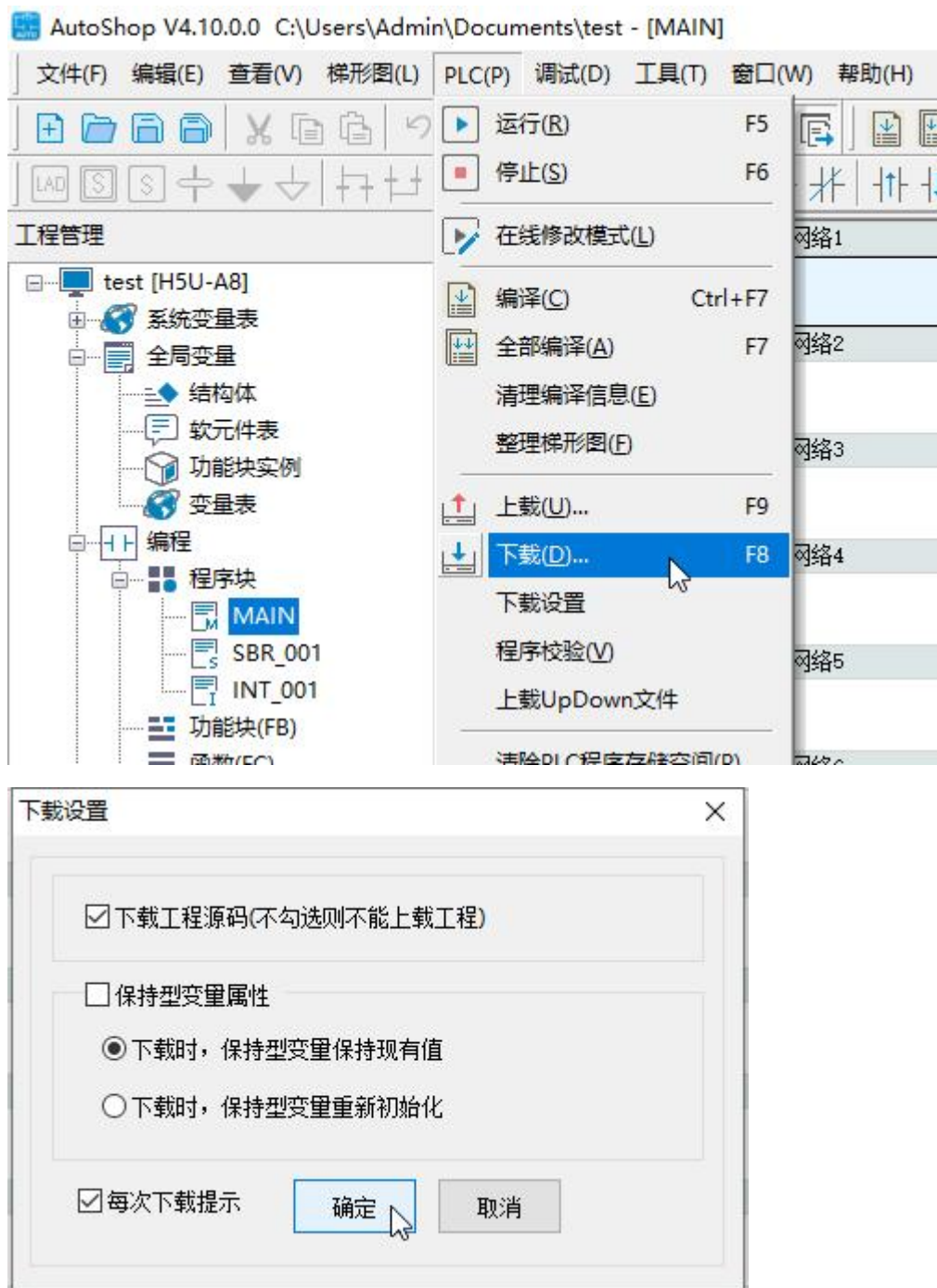


⑤根据所需要的 IO 数据量选择适当的 TPDO 和 RPDO

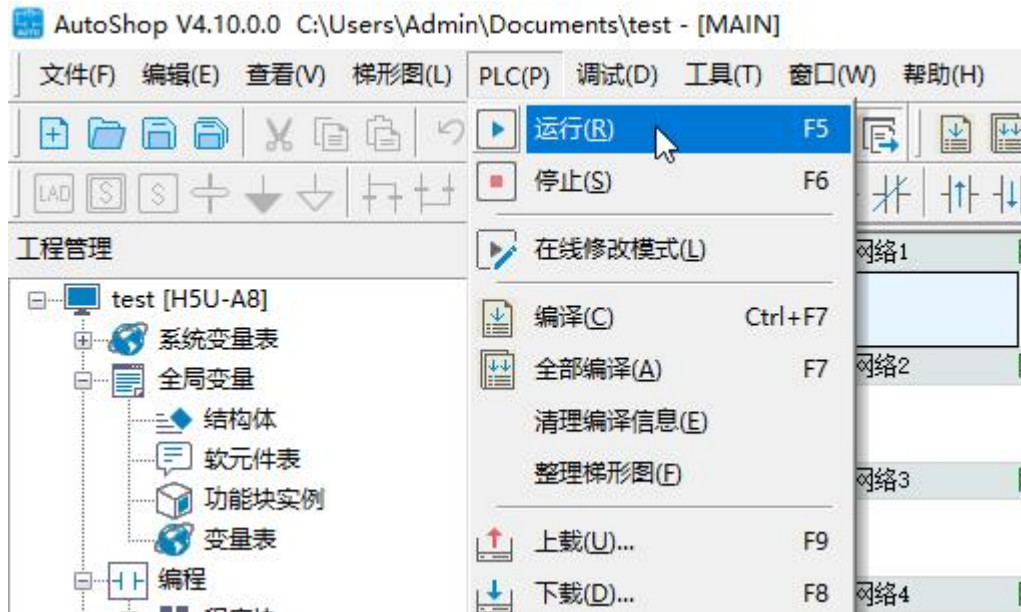


⑥编译并下载程序

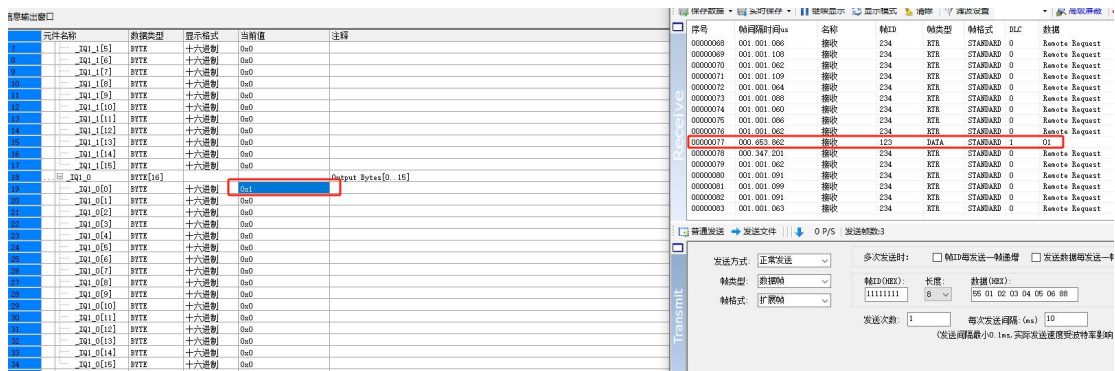




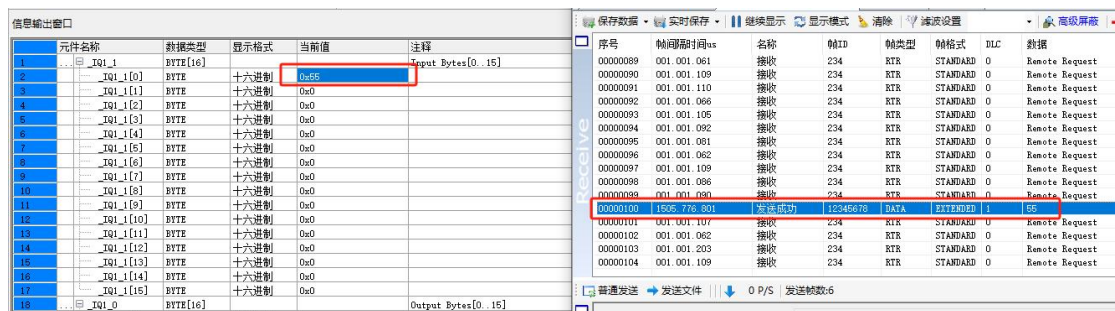
⑦运行 PLC



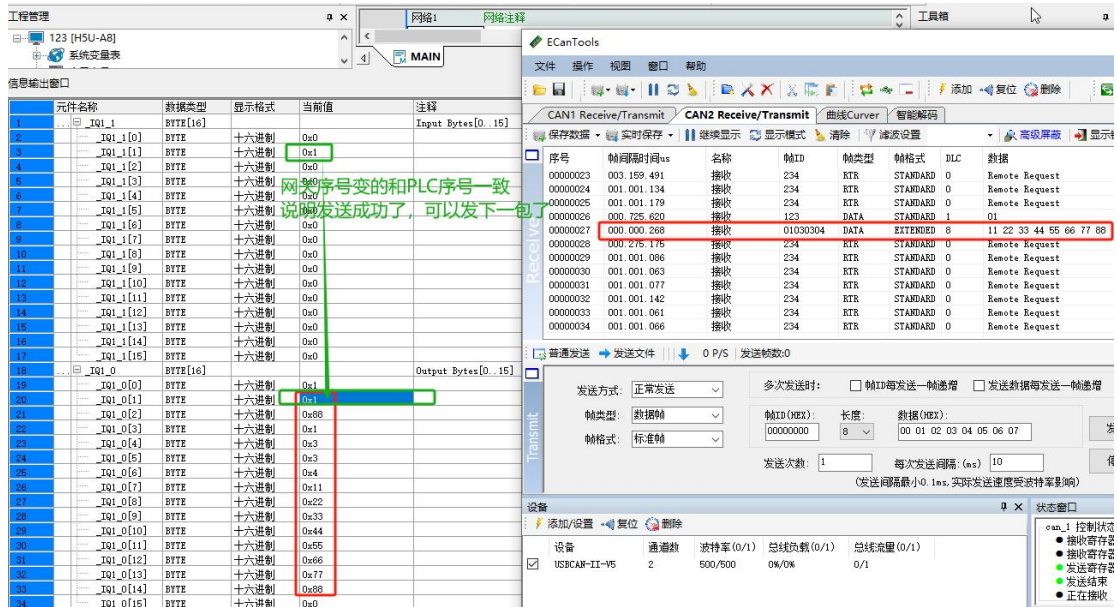
⑧之后网关 SYS 灯常亮，说明进入了 OP 状态
在信息输出窗口里面可以查看、写入数据



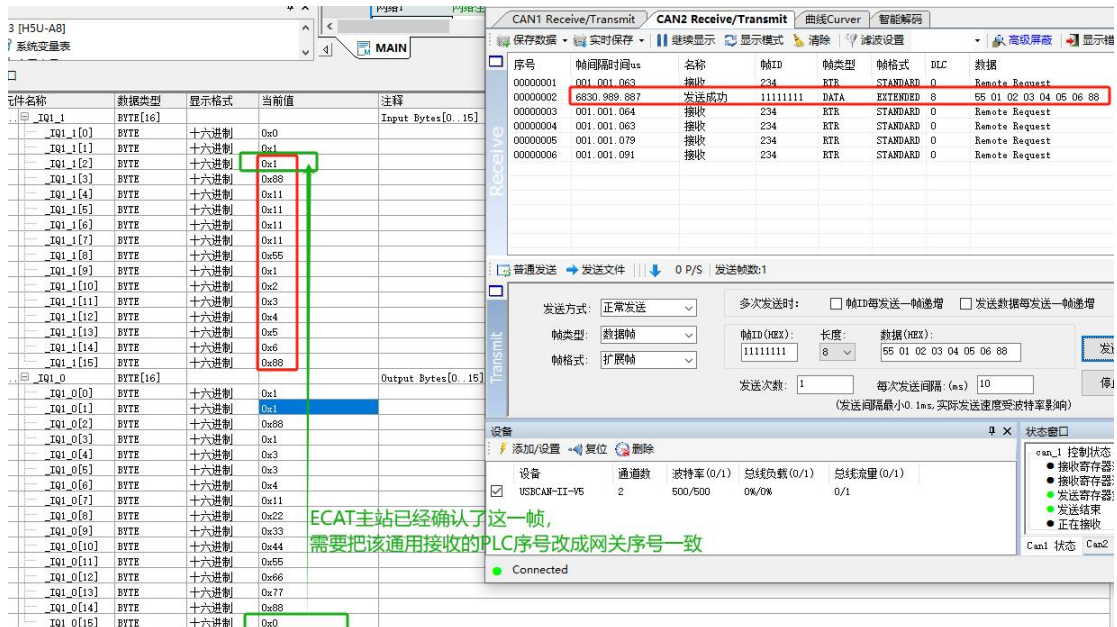
上图可以看出网关在周期发送 ID 为 0x234 的远程帧，Output Bytes[0]改变时发送了一帧 ID 为 0x123 的数据帧。



上图可以看到 CAN 分析仪发送 ID 为 0x12345678 的扩展时 ECAT 主机能收到对应数据。



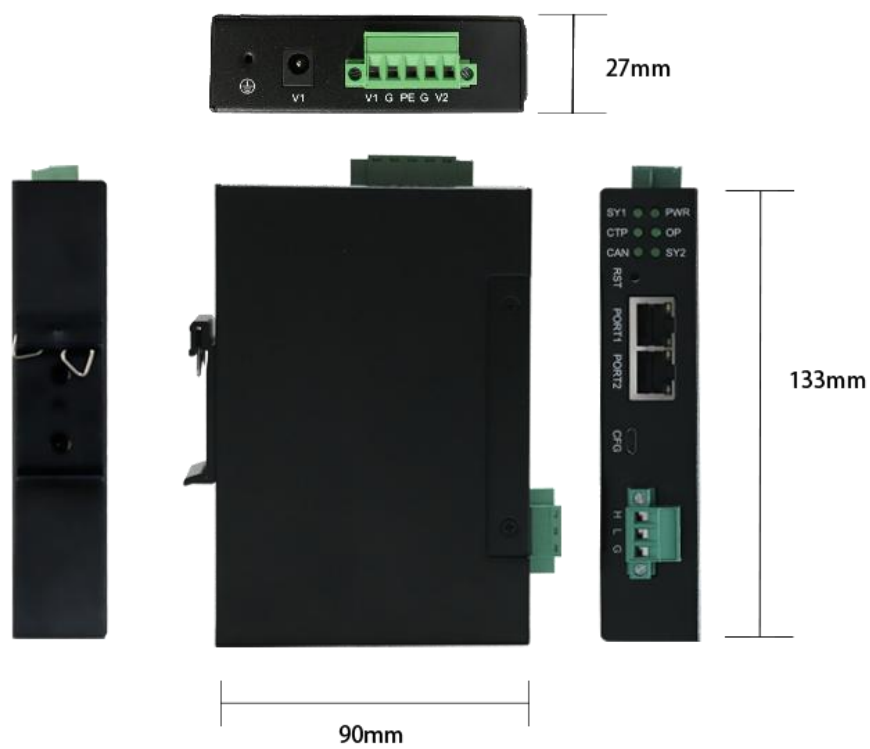
以上两个图可以看到使用通用发送发送了一帧数据，同时网关序号也变到和PLC一致。



上图可以看到 CAN 分析仪发送了一条不在“按 ID 接收”CAN 列表的帧，网关把这个帧放到了通用 CAN 接收，如果 ECAT 主站已经确认了这一帧，需要把该通用接收到 PLC 序号改成和网关序号一致。

七、产品尺寸

产品尺寸如下图，导轨安装。



附录：

CANopen 通信协议简介

CANopen 是一种完全开放和公共的现场总线协议，它是基于 CAN 芯片的面向工业自动化过程的应用层通讯协议。

2.1 CANopen 报文结构

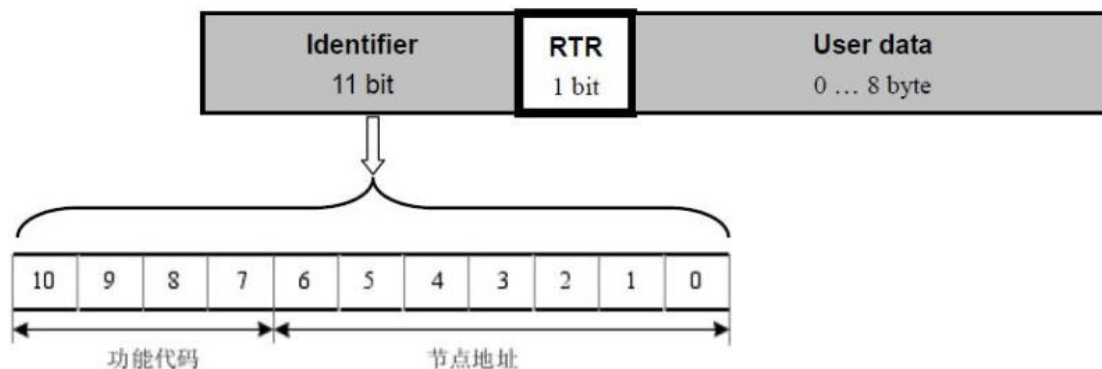


图 3-1 CANopen 报文结构

节点地址由系统集成商定义，例如通过拨码开关设置，范围是 1~127（0 不允许被使用）。

缺省 ID 分配表如下表所示：

Object	Function code	CAN Identifier	Communication parameters at Index
NMT	0000 _b	0	-
SYNC	0001 _b	128 (080 _h)	1005 _h , 1006 _h , 1007 _h
TIME STAMP	0010 _b	256 (100 _h)	1012 _h , 1013 _h
EMERGENCY	0001 _b	129 (081 _h) - 255 (0FF _h)	1014 _h , 1015 _h
TPDO1 (tx)	0011 _b	385 (181 _h) - 511 (1FF _h)	1800 _h
RPDO1 (rx)	0100 _b	513 (201 _h) - 639 (27F _h)	1400 _h
TPDO2 (tx)	0101 _b	641 (281 _h) - 767 (2FF _h)	1801 _h
RPDO2 (rx)	0110 _b	769 (301 _h) - 895 (37F _h)	1401 _h
TPDO3 (tx)	0111 _b	897 (381 _h) - 1023 (3FF _h)	1802 _h
RPDO3 (rx)	1000 _b	1025 (401 _h) - 1151 (47F _h)	1402 _h
TPDO4 (tx)	1001 _b	1153 (481 _h) - 1279 (4FF _h)	1803 _h
RPDO4 (rx)	1010 _b	1281 (501 _h) - 1407 (57F _h)	1403 _h
SSDO (tx)	1011 _b	1409 (581 _h) - 1535 (5FF _h)	1200 _h
SSDO (rx)	1100 _b	1537 (601 _h) - 1663 (67F _h)	1200 _h
NMT ERROR CONTROL	1110 _b	1793 (701 _h) - 1919 (77F _h)	1016 _h , 1017 _h

2.2 CANopen 从站设备的状态机

一个 CANopen 节点从上电开始的 6 种状态，包括：

初始化（**Initializing**）：节点上电后对功能部件包括 CAN 控制器进行初始化；

应用层复位（**Application Reset**）：节点中的应用程序复位（开始），比如开关量输出、模拟量输出的初始值；

通讯复位（**Communication reset**）：节点中的 CANopen 通讯复位（开始），从这个时刻起，此节点就可以进行 CANopen 通讯了；

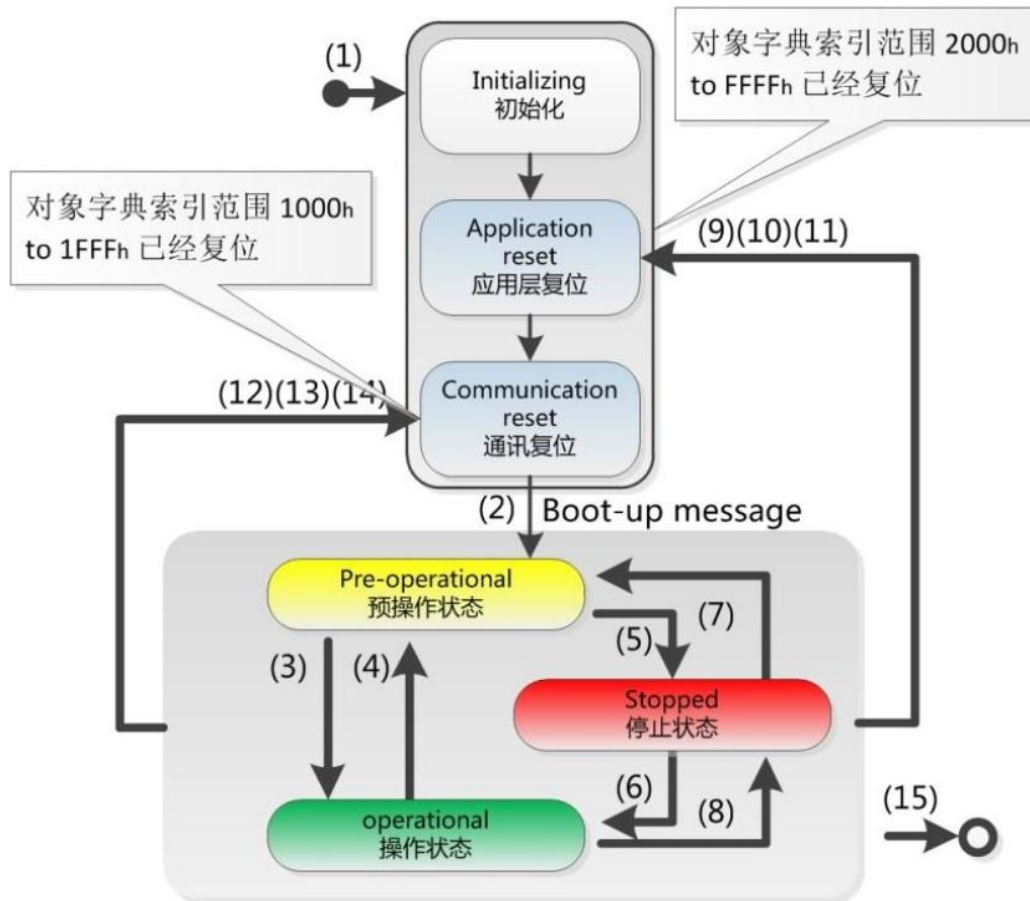
预操作状态（**Pre-operational**）：节点的 CANopen 通讯处于操作就绪状态，此时此节点不能进行 PDO 通信，而可以进行 SDO 进行参数配置和 NMT 网络管理的操作；

操作状态（**Operational**）：节点收到 NMT 主机发来的启动命令后，CANopen 通讯被激活，PDO 通信启动后，按照对象字典里面规定的规则进行传输，同样 SDO 也可以对节点进行数据传输和参数修改；

停止状态（**Stopped**）：节点收到 NMT 主机发来的停止命令后，节点的 PDO 通信被停止，但 SDO 和 NMT 网络管理依然可以对节点进行操作；

除了初始化状态，NMT 主机通过 NMT 命令可以让网络中任意一个的 CANopen 节点进行其他 5 种状态的切换。如图 3-2 所示。

当然 CANopen 节点也可以自动完成这些状态的切换。



- (1)Power on上电
 (2)Automatic switch to Pre-operational
 自动切换到预操作状态
 (3)and(6)NMT Switch to Operational
 网络管理切换到操作状态
 (4)and(7)NMT Switch to Pre-Operational
 网络管理切换到预操作状态
 (5)and(8)NMT Switch to Stopped
 网络管理切换到停止状态
 (9),(10)and(11)NMT Switch to Application reset
 网络管理切换到应用层复位状态
 (12),(13)and(14)NMT Switch to Communication reset
 网络管理切换到通讯复位状态
 (15)Power-off or hardware reset
 掉电或者硬件复位

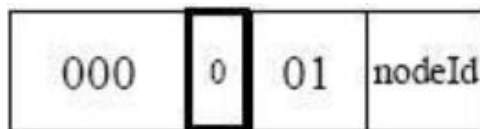
图 3-2 状态转换图

2.3 CANopen 子协议

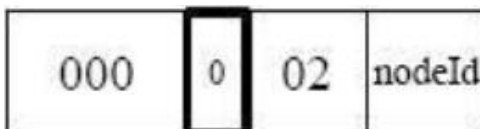
以下子协议中涉及的数字均为十六进制；RTR bit=0（不是远程帧），如黑框所示。

2.3.1 NMT 协议

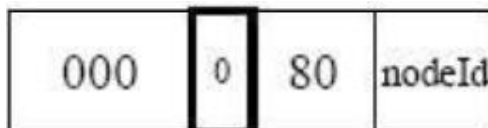
使节点进入 Operational 状态，发送命令：



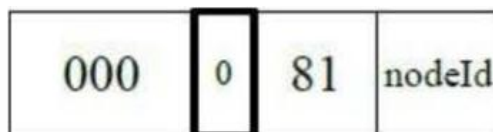
使节点进入 Stop 状态，发送命令：



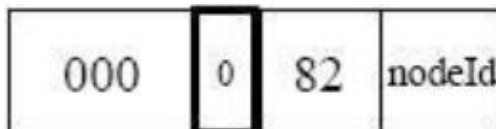
使节点进入 Pre-operational 状态，发送命令：



使节点进入 Application Reset 状态，发送命令：



使节点进入 Communication reset 状态，发送命令：



如果对所有节点发送命令，则 nodeId=0；

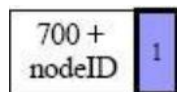
例：

如果使节点 0x06 进入 Operational 状态：000 01 06

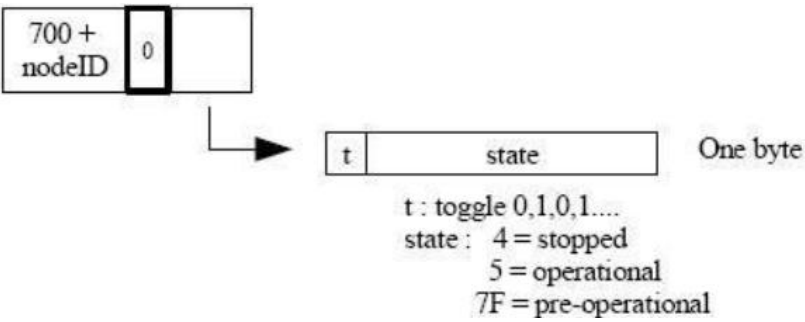
如果使所有节点进入 Pre-operational 状态：000 80 00

2.3.2 Node guard 协议

查询 CANopen 从站的状态，主站发送标准远程帧（无数据）如下：



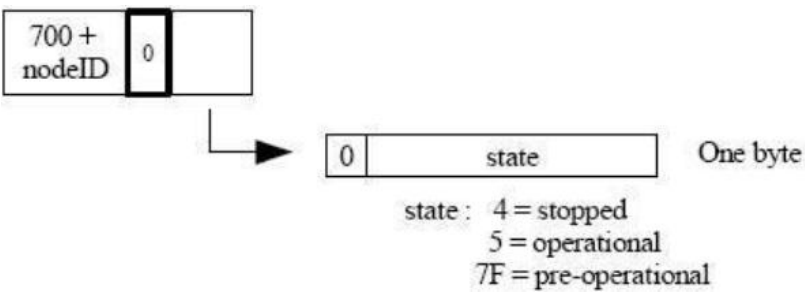
从站响应：



注：其中位 t（最高位）的值 0，1 交替变化。

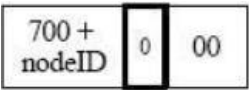
2.3.3 Heartbeat 协议

不需要主站发送请求命令，CANopen 从站周期性的发送其状态帧：



2.3.4 Bootup 协议

CANopen 从站节点初始化后进入 Pre-operational 状态时发送：



2.3.5 SDO 协议

命令格式：

Identifier	Command	Index Low Byte	Index High Byte	Subindex	Data 1	Data 2	Data 3	Data 4
------------	---------	----------------	-----------------	----------	--------	--------	--------	--------

响应格式：

Identifier	Command	Index Low Byte	Index High Byte	Subindex	Data 1	Data 2	Data 3	Data 4
------------	---------	----------------	-----------------	----------	--------	--------	--------	--------

读命令

主站发送命令：

600 + Serv NodeId	0	40	Index	Sub index	00	00	00	00
----------------------	---	----	-------	--------------	----	----	----	----

从站响应：

The server responds (if success) :

Data length = 1 byte

580 + Serv NodeId	0	4F	Index	Sub index	d1	x	x	x
----------------------	---	----	-------	--------------	----	---	---	---

X : undefined. Should be 0

The server responds (if success) :

Data length = 2 bytes

580 + Serv NodeId	0	4B	Index	Sub index	d1	d0	x	x
----------------------	---	----	-------	--------------	----	----	---	---

X : undefined. Should be 0

The server responds (if success) :

Data length = 3 bytes

580 + Serv NodeId	0	47	Index	Sub index	d2	d1	d0	x
----------------------	---	----	-------	--------------	----	----	----	---

X : undefined. Should be 0

The server responds (if success) :

Data length = 4 bytes

580 + Serv NodeId	0	43	Index	Sub index	d3	d2	d1	d0
----------------------	---	----	-------	--------------	----	----	----	----

The server responds (if failure) :

580 + Serv NodeId	0	80	Index	Sub index	SDO abort code error			
----------------------	---	----	-------	--------------	----------------------	--	--	--

写命令

主站发送命令：

The client request :

Data length = 1 byte

600 + Serv NodeId	0	2F	Index	Sub index	d0	x	x	x
----------------------	---	----	-------	--------------	----	---	---	---

X : undefined. Put 0

The client request :

Data length = 2 bytes

600 + Serv NodeId	0	2B	Index	Sub index	d1	d0	x	x
----------------------	---	----	-------	--------------	----	----	---	---

X : undefined. Put 0

The client request :

Data length = 3 bytes

600 + Serv NodeId	0	27	Index	Sub index	d2	d1	d0	x
----------------------	---	----	-------	--------------	----	----	----	---

X : undefined. Put 0

The client request :

Data length = 4 bytes

600 + Serv NodeId	0	23	Index	Sub index	d3	d2	d1	d0
----------------------	---	----	-------	--------------	----	----	----	----

从站响应：

The server responds (if success) :

580 + Serv NodeId	0	60	Index	Sub index	00	00	00	00
----------------------	---	----	-------	--------------	----	----	----	----

The server responds (if failure) :

580 + Serv NodeId	0	80	Index	Sub index	SDO abort code error			
----------------------	---	----	-------	--------------	----------------------	--	--	--

2.3.6 PDO 协议

PDO 数据的传输可以通过 SYNC、RTR，或者基于事件进行传输：

Identifier	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7	Data 8
------------	--------	--------	--------	--------	--------	--------	--------	--------

其中：Identifier 为表 3-1 中对应的 ID 值。

2.3.7 SYNC 协议

80	0
----	---