

目录

一、概述 ..... 2

二、LED 指示灯状态灯定义 ..... 2

三、产品技术参数 ..... 3

四、模块参数配置 ..... 3

五、使用 CODESYS 和模块通信 ..... 6

六、使用欧姆龙 Sysmac Studio 和模块通信 ..... 12



## 免责声明条款

感谢您购买高迈德产品。在使用之前，请仔细阅读本声明，一旦使用，即被视为对本声明全部内容的认可和接受。请严格遵守手册、产品说明和相关的法律法规、政策、准则安装和使用该产品。在使用产品过程中，用户承诺对自己的行为及因此而产生的所有后果负责。因用户不当使用、安装、改装造成的任何损失，高迈德将不承担法律责任。

本产品及手册为高迈德版权所有。未经许可，不得以任何形式复制翻印。关于免责声明的最终解释权，高迈德所有。

## 一、概述

高迈德 ECAT-01CAN 是一款 CAN 自由口 转 EtherCAT 网关，用于将 CAN 总线数据映射到 EtherCAT。使用 ESI 文件进行集成，免代码读取、读写 CAN 总线设备，可以快速设置数据并将其传输到 PLC 中。网关提供一路 CAN 接口，最大支持 30 个从站。

## 二、LED 指示灯状态灯定义

◆PWR 指示灯;电源指示灯

◆CAN 指示灯:灭(网关 CAN 口无接收到数据)，闪亮(网关 CAN 口接收到数据)，CAN 总线数据传送速率越高，越接近常亮。

◆RUN 指示灯:

1. 开机时 RUN 灯短闪一次
2. RUN 灭: 无 EtherCAT 网络信号
3. RUN 闪亮 ( 2 秒周期): 检测到 EtherCAT 通信，组态尚未完成
4. RUN 闪亮 ( 0.5 秒周期): 组态软件正在控制 LED 闪灯
5. RUN 常亮: 正常数据通信时 RUN (绿) 常亮
6. 没有接入 EtherCAT 主站时，RUN 常亮: 网关处于配置模式。
7. 正常工作状态下: RUN 灯常亮，CAN 灯亮。

### 三、产品技术参数

|               |                                                           |
|---------------|-----------------------------------------------------------|
| 现场总线          | EtherCAT                                                  |
| 最快循环周期        | 根据主站                                                      |
| 发送字节          | 最大 500 字节                                                 |
| 接收字节          | 最大 500 字节                                                 |
| 扫描周期          | <3ms                                                      |
| CAN           | 波特率(bps):<br>10K, 20K, 50K, 100K, 125K, 250K, 500K, 1000K |
| CAN 通信方式      | PDO, 需要 SDO 可以定制                                          |
| 最多支持设备数量      | 30                                                        |
| 最多支持 CAN 信令数量 | 80                                                        |
| 数据传输介质        | Ethernet CAT5 电缆                                          |
| 传输速率          | 100Mb/s                                                   |
| TYPE--C 接口    | 升级固件或配置模块参数时使用                                            |
| 工作电压          | 12-24V DC                                                 |
| 外形尺寸          | 100.8*80*35mm                                             |
| 工作温度          | -40 到 85℃                                                 |
| 防护等级          | IP20                                                      |

### 四、模块参数配置

4.1、首先将 type-c 口和电脑 USB 口连接好，打开“高迈德 CAN 组态软件”，选择对应 COM 端口，点击打开，此时模块 RUN 灯常亮表示处于配置模式，如果没有进入配置模块，可以在打开 COM 口后，把网关断电重启。



4.2、【波特率】：根据 CAN 总线的波特率选择相应波特率

4.3、【发送周期】：网关循环往 CAN 总线上写值的间隔，如下图

| CAN1 Receive/Transmit |                  |    |     | CAN2 Receive/Transmit |          |     |          |
|-----------------------|------------------|----|-----|-----------------------|----------|-----|----------|
| 保存数据                  |                  |    |     | 实时保存                  |          |     |          |
| 继续显示                  |                  |    |     | 显示模式                  |          |     |          |
| 清除                    |                  |    |     | 滤波设置                  |          |     |          |
| 高级屏蔽                  |                  |    |     |                       |          |     |          |
| 序号                    | 帧连续时间us          | 名称 | 帧ID | 帧类型                   | 帧格式      | DLC | 数据       |
| 00069345              | 14:13:16.726.965 | 接收 | 0B5 | DATA                  | STANDARD | 3   | 01 02 03 |
| 00069346              | 14:13:16.826.723 | 接收 | 0B5 | DATA                  | STANDARD | 3   | 01 02 03 |
| 00069347              | 14:13:16.926.478 | 接收 | 0B5 | DATA                  | STANDARD | 3   | 01 02 03 |
| 00069348              | 14:13:17.026.256 | 接收 | 0B5 | DATA                  | STANDARD | 3   | 01 02 03 |
| 00069349              | 14:13:17.126.006 | 接收 | 0B5 | DATA                  | STANDARD | 3   | 01 02 03 |
| 00069350              | 14:13:17.225.755 | 接收 | 0B5 | DATA                  | STANDARD | 3   | 01 02 03 |

4.4、【status Bytes】：在一个循环没有收到数据相应状态位置 0，通讯正常为 1，状态字节最低位对应第一个槽，以此类推。

4.5、【Enable Bytes】：控制字字节最低位对应第一个槽，以此类推，为 1 使能收发，为 0 不使能。如下图示例：

控制字字节为 QB2，最低位 QB2.0 置 1，第一条地址为 2 的命令正常读。QB2.1 置 1，第二条地址为 1 的命令正常写。

高迈德CAN组态软件

COM3

打开

读取文件

保存文件

读取芯片

写入芯片

波特率

500k

发送周期(ms)

100

起始地址: I区

Q区

检查

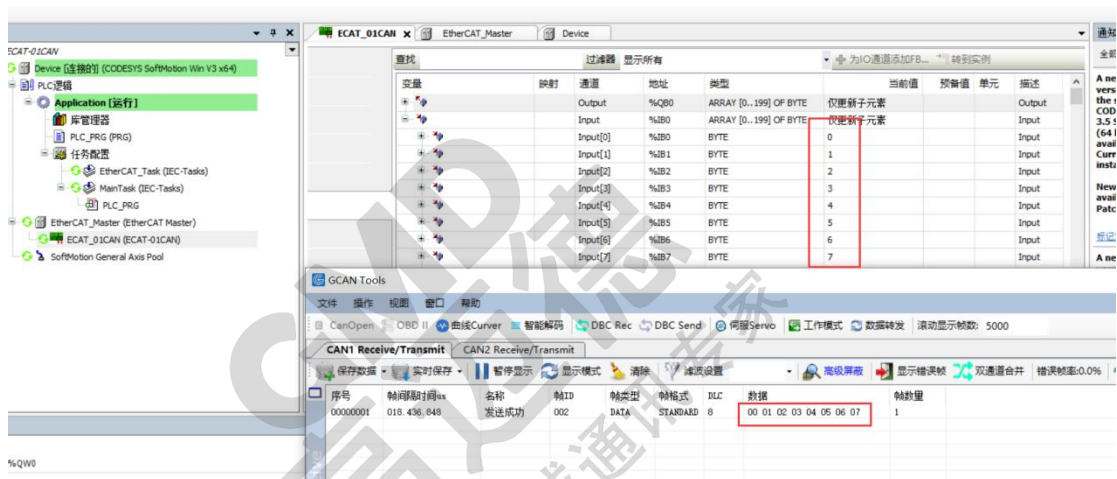
0

0

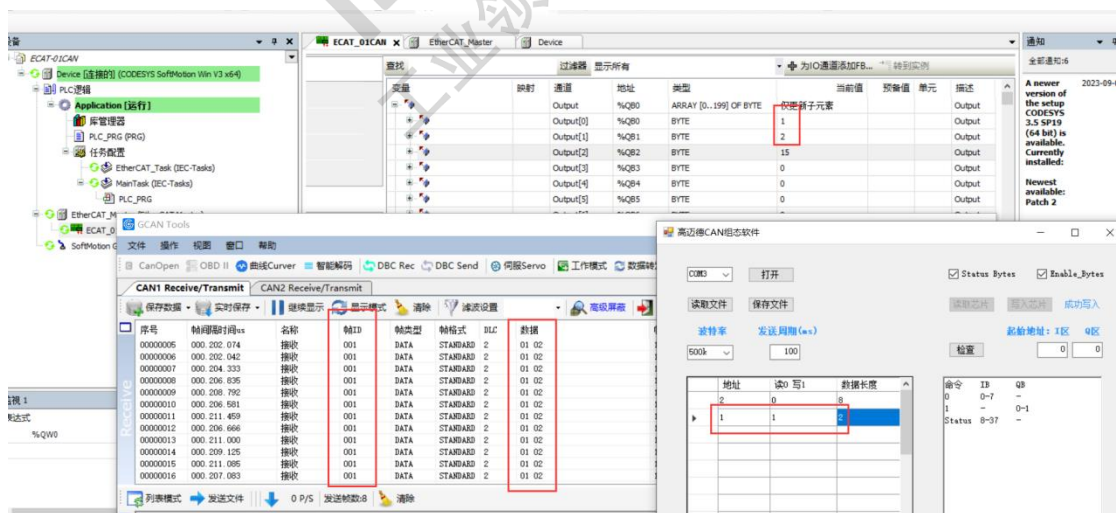
|  | 地址 | 读0 写1 | 数据长度 |
|--|----|-------|------|
|  | 2  | 0     | 8    |
|  | 1  | 1     | 2    |
|  |    |       |      |
|  |    |       |      |
|  |    |       |      |
|  |    |       |      |
|  |    |       |      |

| 命令     | IB   | QB   |
|--------|------|------|
| 0      | 0-7  | -    |
| 1      | -    | 0-1  |
| Status | 8-37 | -    |
| Enable | -    | 2-31 |

4.6、【命令栏】：地址为 CAN ID，读命令为 0，写命令为 1，数据长度最大不超过 8 字节。如上图，ID 2 是读 8 个字节数据放到 IB0-IB7。如下图：



ID1 是循环写，把 QB0-QB1 的值按 100ms 的周期往 CAN 总线发，如下图：



#### 4.7、【起始地址】：I 区 Q 区

网关分配的内存起始地址与 Ethercat 主站分配的起始地址一致。

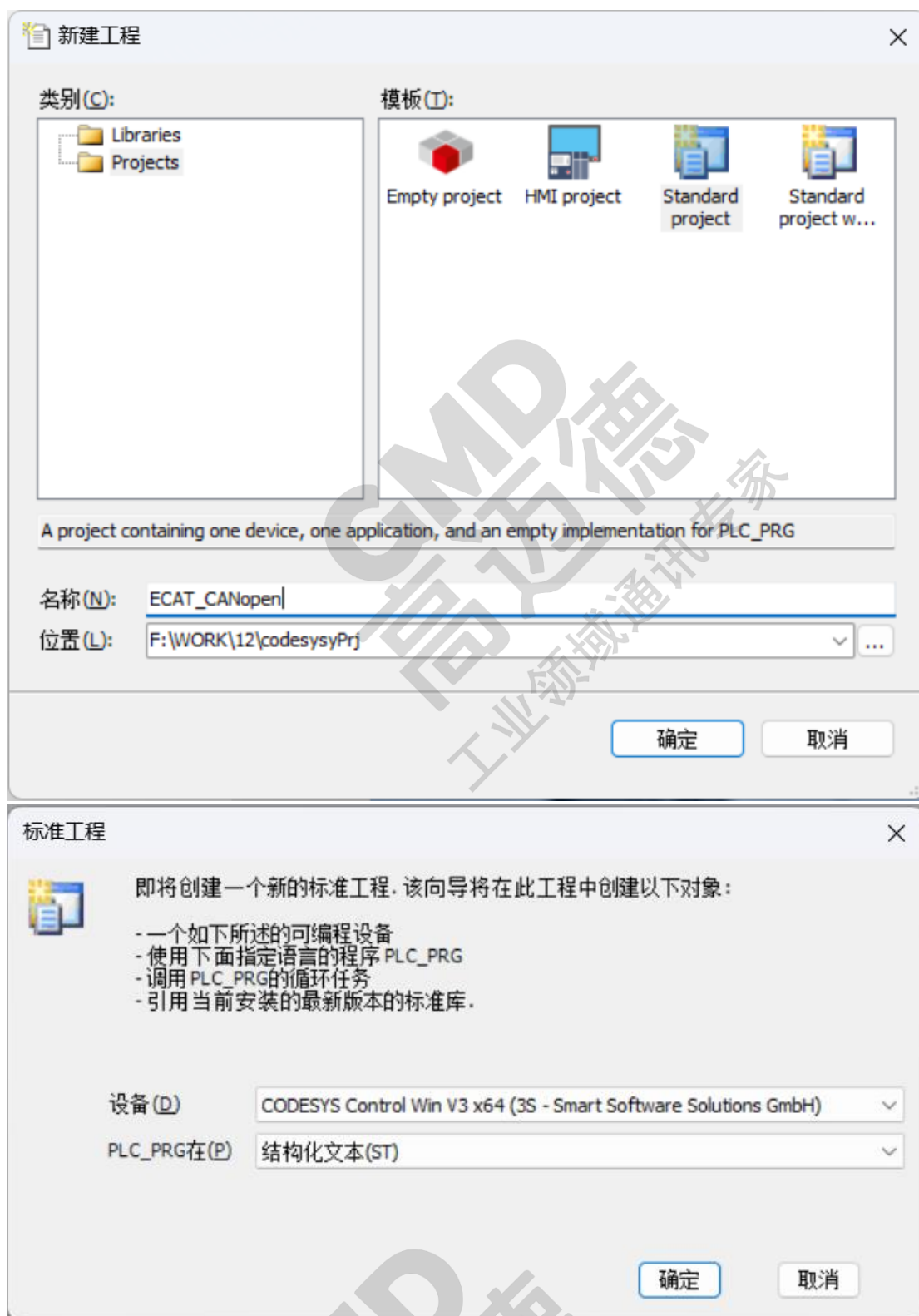


4.8、【写入芯片】：在连接串口后，RUN 灯常亮表示进入配置模式，设置完成点击“写入芯片”按钮，等待软件将参数写入模块，提示写入成功，**模块配置完成，此时必须关闭串口**，关闭串口后 RUN 灯灭，恢复工作模式。

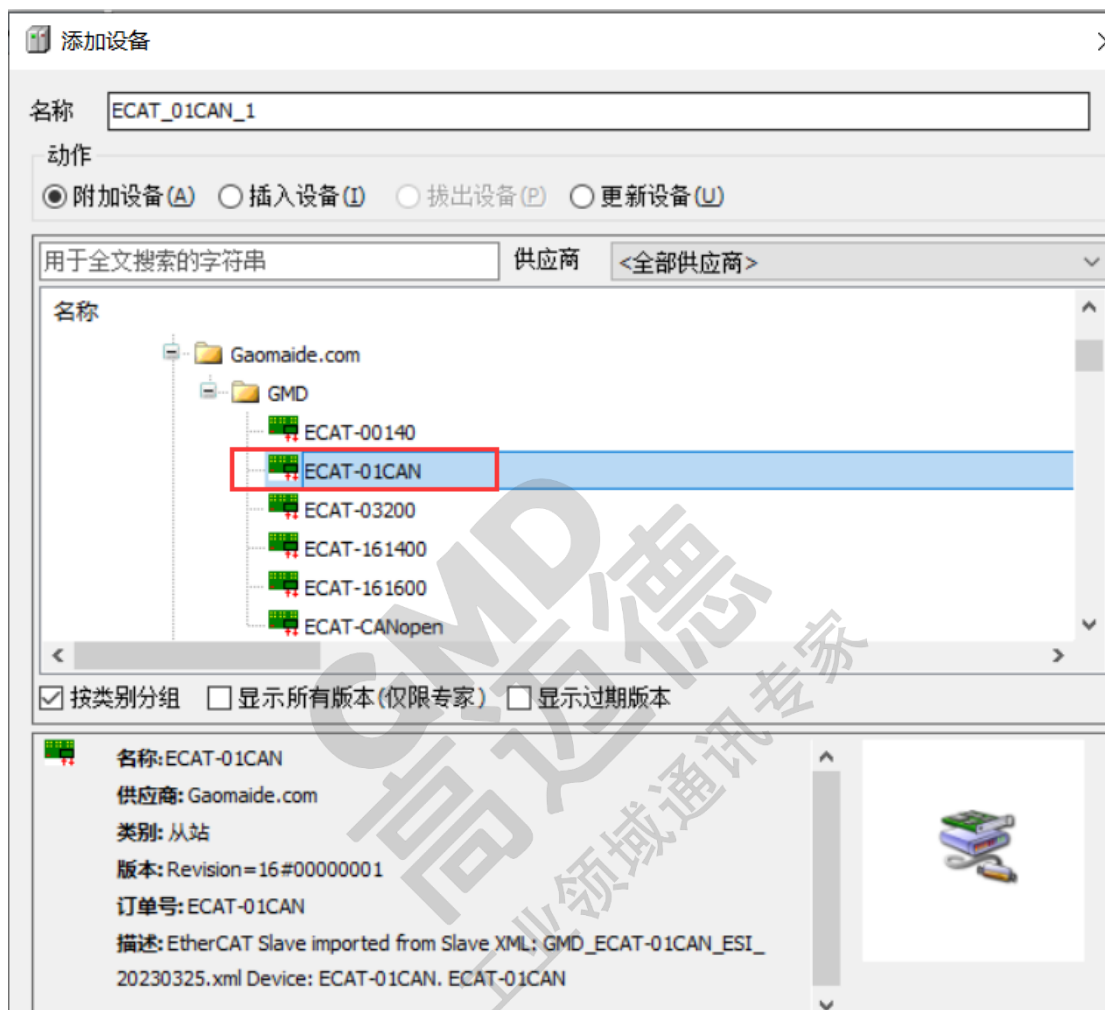


## 五、使用 CODESYS 和模块通信

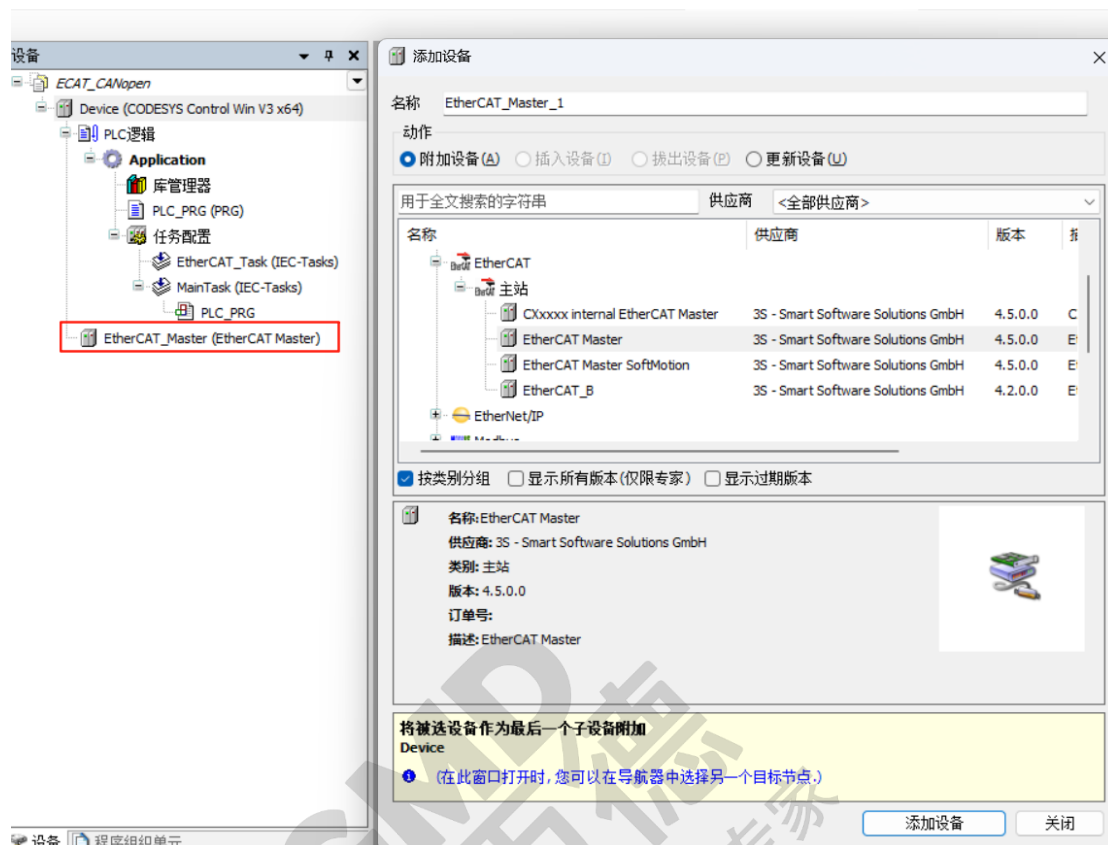
5.1、打开 codesys 软件（实例 codesys 版本：3.5 SP19），新建工程（以电脑当做 PLC 为例）



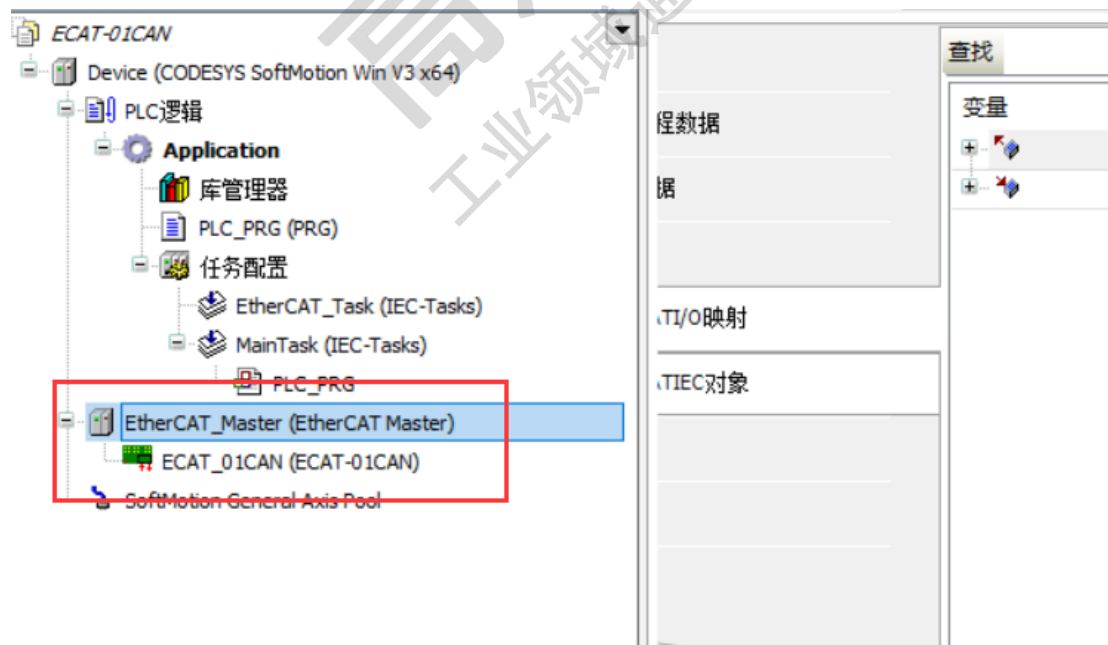
5.2、导入 ECAT-01CAN 模块的 ESI 文件，点击 codesys 菜单栏-工具-设备存储库-安装；找到我们提供的 ESI 文件安装，安装完成如下图



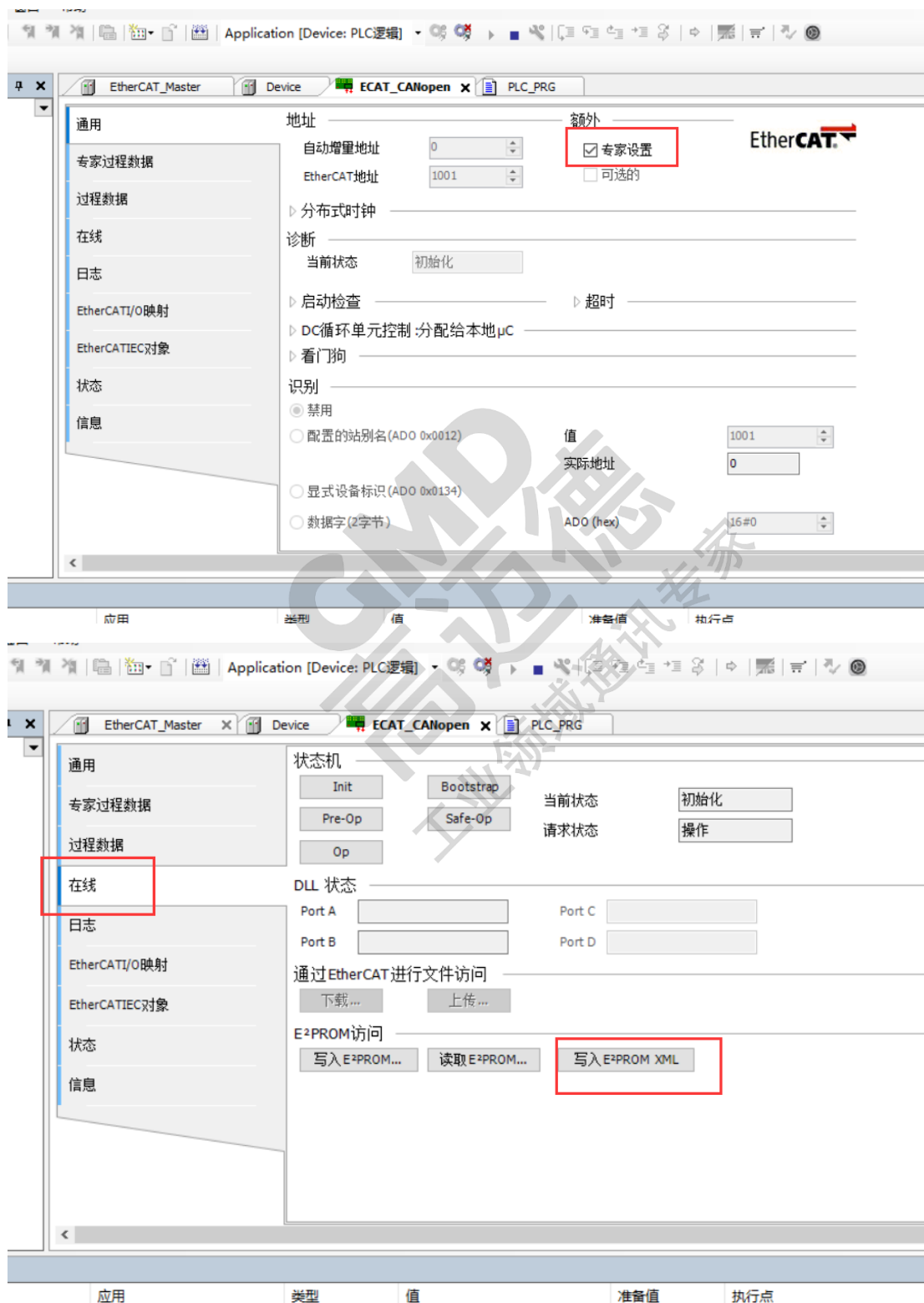
5.3、添加 EtherCAT 主站，右键 Device-添加设备-现场总线-EtherCAT-主站-EtherCAT Master，点击添加设备



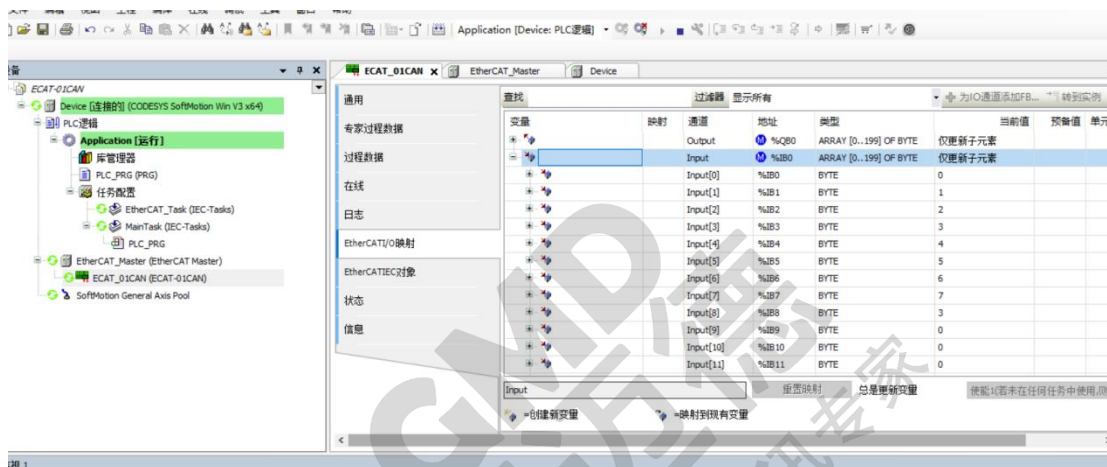
5.4、添加模块，点击刚添加的 EtherCAT\_Master-现场总线-从站-Gaomaide.com-GMD-ECAT-01CAN，添加设备



在 ECAT-01CAN 设备中点通用，打开专家设置，连接在线，写入 XML，网关断电重启，重启完 run 灯不闪，表示已经写入 XML，**写入后网关断电重启。**

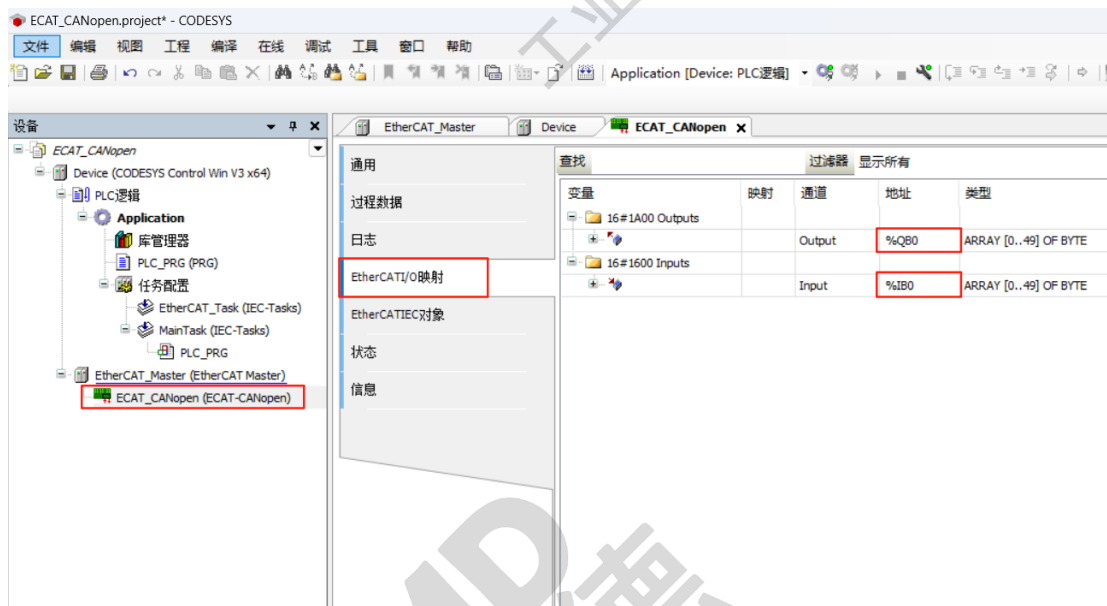


5.5、双击 EtherCAT\_Master，通用-选择以太网，双击 ECAT-01CAN，EtherCAT I/O 映射-使用父设备设置改成使能。



5.6、点击生成代码，再点击登录-运行，模块和主站通讯正常，可读取和写入数据到 01CAN 从站。

5.7、对象字典和 PLC 地址对应关系，首先查看主站 Q 区和 I 区起始地址如下图 Q 区和 I 区起始地址都是 0

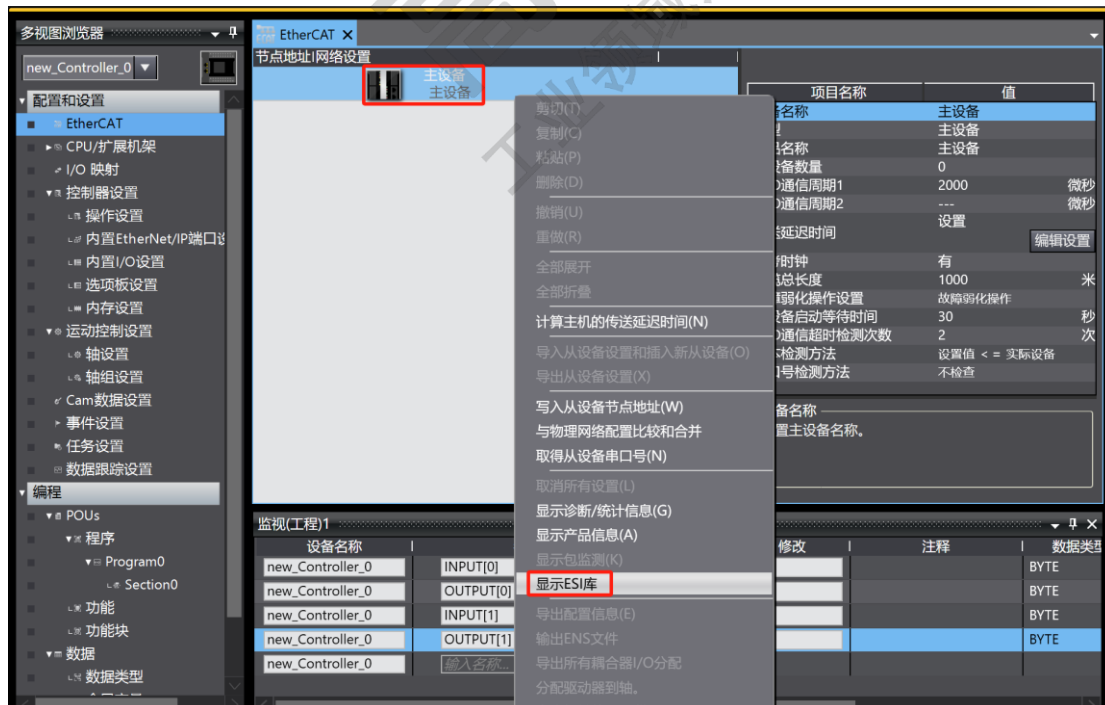


打开“高迈德 CAN 组态软件”，修改 Q 区和 I 区起始地址和主站一致，点击“检查”ECAT-01CAN 和主站的地址就对齐了

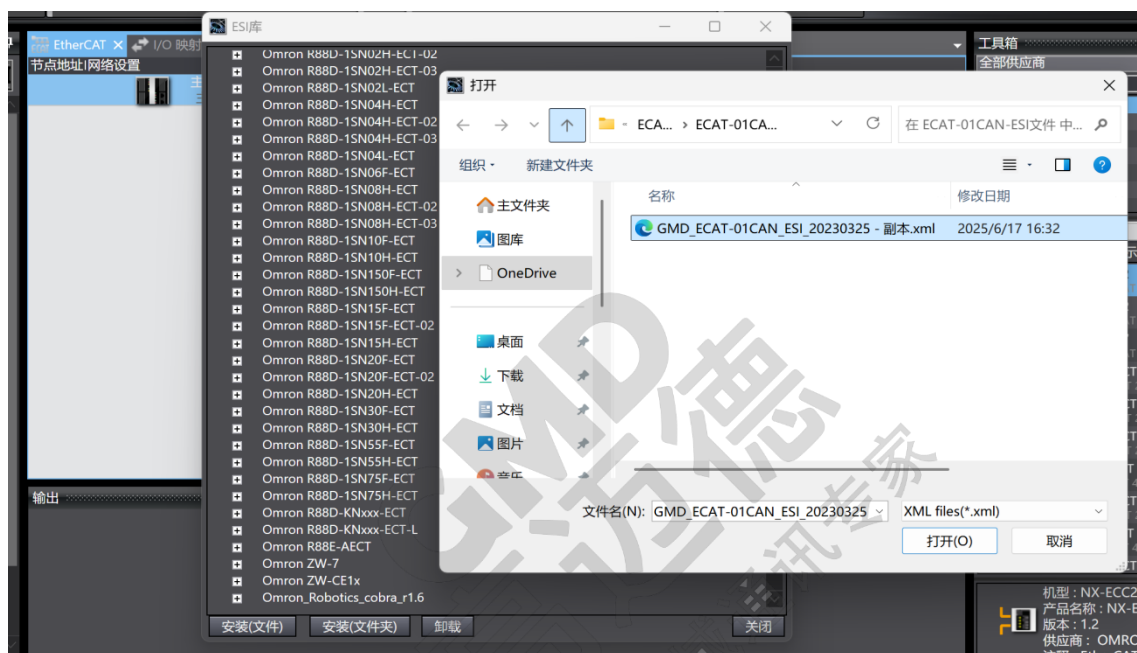


## 六、使用欧姆龙 Sysmac Studio 和模块通信

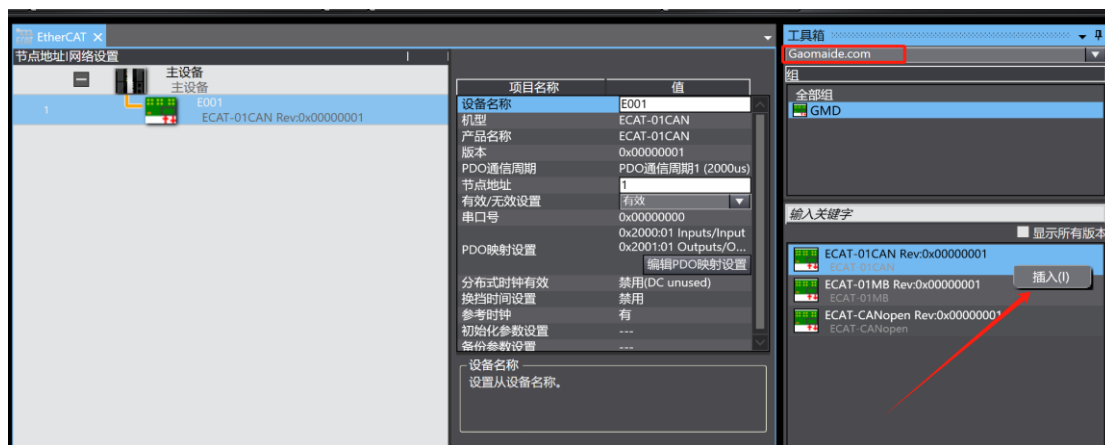
1、打开设备树点击 EtherECAT 组态页面，右键主设备显示 ESI 库安装 ECAT-CAN 设备 XML 设备描述文件



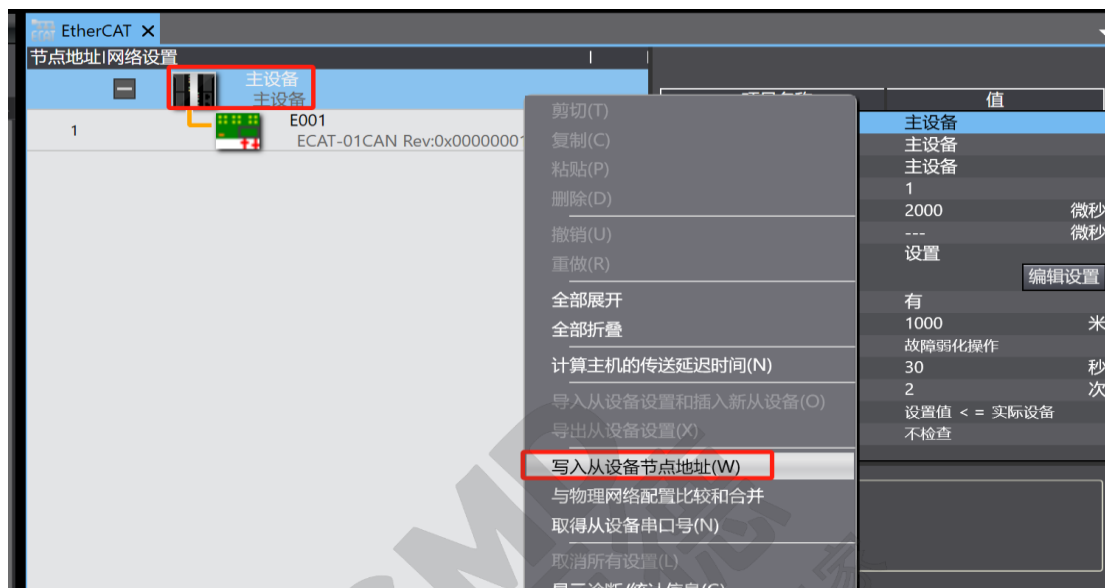
2、打开 ESI 库后下方安装文件找到 ECAT-CAN 的 xml 文件路径，选择打开安装后显示设备信息



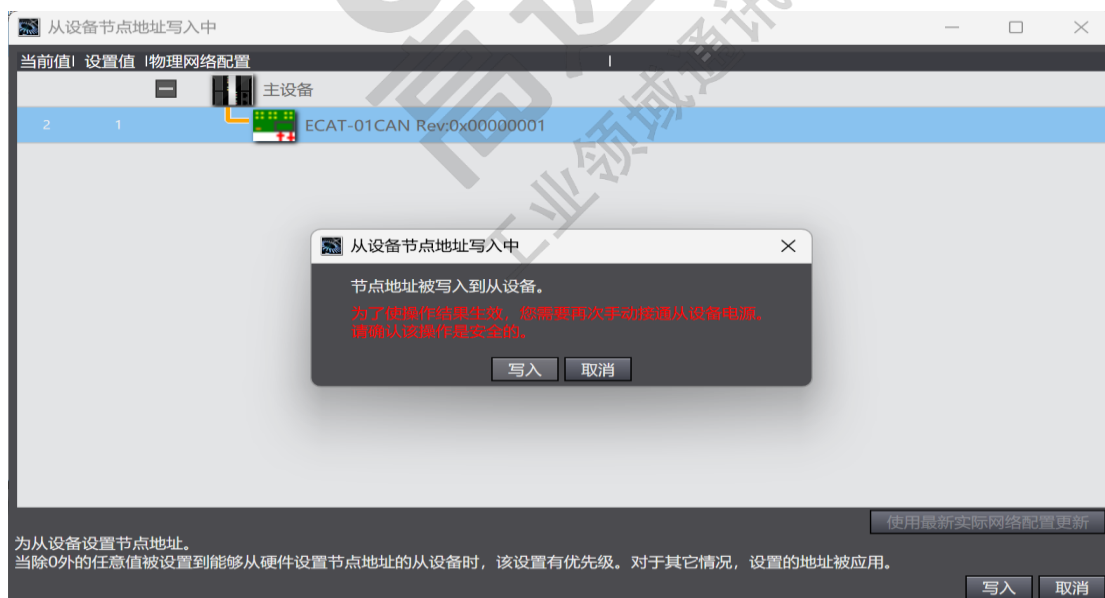
3、右边工具箱下方供应商选择 Gaomaide.com，全部组下方 GMD，找到 ECAT-CAN 设备右键  
插入到主设备下方



4、在线后右键主设备选择写入从设备节点地址



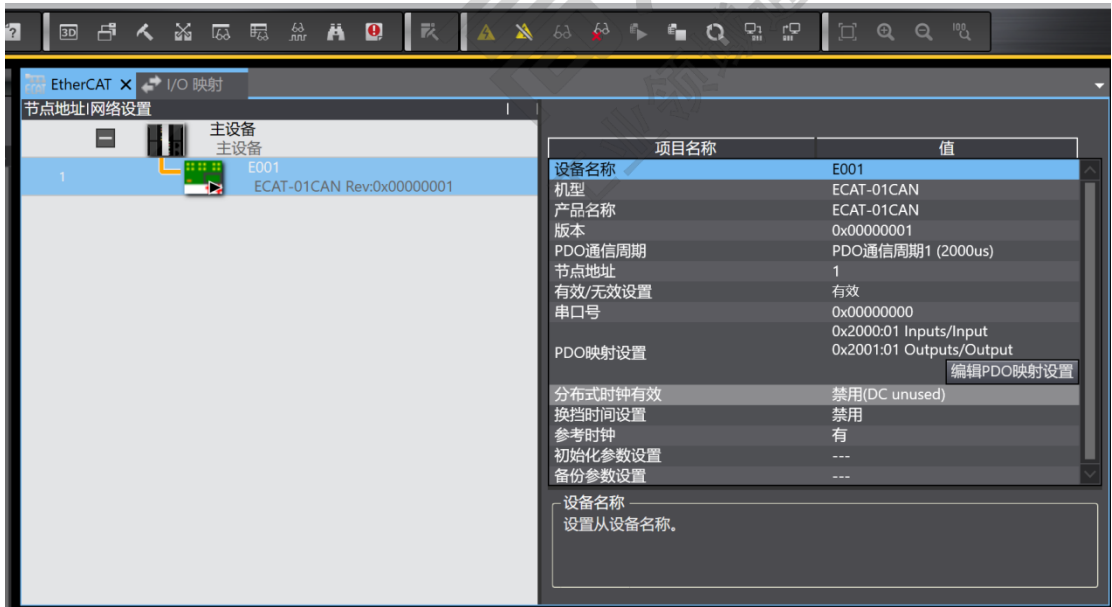
5、设置值与组态时的节点地址一致，修改写入后按照提示给从设备断电重启



6、随后在编程模式下点击控制器栏中选择同步，传送到控制器



7、切换到运行模式，ECAT-CAN 从站设备呈现三角运行符号表示 PLC 与 ECAT-CAN 设备正常通信  
此时可观察 ECAT-CAN 设备 RTU 灯常亮，PLC 控制器的 ECAT 状态灯无异常



8、在左边设备树中 IO 映射选项找到 ECAT-CAN 设备映射的 IO 地址，根据 ECAT-CAN 组态工具读写指令对其进行数据交互

The screenshot displays the GCAN Tools software interface. The main window shows the 'EtherCAT I/O 映射' (EtherCAT I/O Mapping) configuration. The left sidebar contains a tree view with 'new\_Controller\_0' selected, and 'I/O 映射' (I/O Mapping) is highlighted under '配置和设置' (Configuration and Settings). The main area shows the 'EtherCAT网络配置' (EtherCAT Network Configuration) table, which lists the mapping for '节点1' (Node 1) under 'ECAT-01CAN'. The table shows 'Inputs\_Input\_2000\_01' and 'Outputs\_Output\_2001\_01' with their respective R/W, data types, and values.

Overlaid on the main window is the 'GCAN Tools' dialog box, which is currently showing the 'CAN1 Receive/Transmit' data table. This table lists received CAN messages with columns for '序号' (Serial Number), '帧间隔时间ms' (Frame Interval Time in ms), '名称' (Name), '帧ID' (Frame ID), '帧类型' (Frame Type), '帧格式' (Frame Format), 'DLC' (Data Length Code), and '数据' (Data). The table shows three received frames with IDs 00000891, 00000892, and 00000893, all of type '接收' (Receive) and format 'STANDARD'.

Below the main window, a detailed view of the 'EtherCAT I/O 映射' table is shown, focusing on the 'Inputs\_Input\_2000\_01' mapping. This table lists the mapping for '节点1' (Node 1) under 'ECAT-01CAN'. The table shows the mapping for 'Inputs\_Input\_2000\_01' with its R/W, data type, and value.

| 位置  | 端口         | 说明                   | R/W | 数据类型         | 值    | 变量 | 变量注 |
|-----|------------|----------------------|-----|--------------|------|----|-----|
| 节点1 | ECAT-01CAN | Inputs_Input_2000_01 | R   | ARRAY[0..49] |      |    |     |
|     | [0]        |                      | R   | BYTE         | 16#0 |    |     |
|     | [1]        |                      | R   | BYTE         | 16#1 |    |     |
|     | [2]        |                      | R   | BYTE         | 16#2 |    |     |
|     | [3]        |                      | R   | BYTE         | 16#3 |    |     |
|     | [4]        |                      | R   | BYTE         | 16#4 |    |     |
|     | [5]        |                      | R   | BYTE         | 16#5 |    |     |
|     | [6]        |                      | R   | BYTE         | 16#6 |    |     |
|     | [7]        |                      | R   | BYTE         | 16#7 |    |     |
|     | [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 |    |     |