EtherCAT Slave to

CANopen Master 网关产品手册

技术支持热线：010-85958895　邮箱：cn-sales@jiyuansys.com
地址：北京市朝阳区朝阳门北大街乙 12 号天辰大厦 8 层 808 室

目　录

版本说明：

版本	更新时间	更新内容	更新者
Ver1.0.0	20211225	初版	EnTalk

1 引言

1.1 关于说明书

本说明书描述了网关 EnTalk EtherCAT Slave to CANopen Master（以下简称 ECAT – CANopen）的各项参数，具体使用方法和注意事项，为方便工程人员的操作使用。在使用网关之前，请仔细阅读本说明书。

1.2 版权说明

1.3 术语

CANopen：CIA 推出基于 CAN 的一种通讯规范。

EtherCAT：是一项高性能、低成本、应用简易、拓扑灵活的工业以太网技术，可用于工业现场级的超高速 I/O 网络。

2 产品概述

2.1 产品功能

本产品实现 EtherCAT 网络与 CANopen 网络之间的数据通讯，可连接 CANopen 网络到 EtherCAT 网络。即将 CANopen 设备连接到 EtherCAT 网络。

2.2 产品特点

- 应用广泛：应用于支持 CANopen 接口的变频器、步进电机、伺服驱动器、仪表、PLC、DCS、FCS 等等。
- 配置简单：用户不必了解 CANopen 和 EtherCAT 细节，只需要参考手册，根据要求就能配置网关，不需要复杂编程，即可在短时间内实现连接功能。

2.3 技术指标

ECAT – CANopen 在 EtherCAT 一侧为 EtherCAT 从站，在 CANopen 一侧可以作为 CANopen 主站。

- ESI 文件由配置工具自动生成
- 支持最大的输入字节数为 512 字节，最大的输出字节为 512 字节
- CANopen 支持波特率：10kbit/s, 20kbit/s, 50kbit/s, 100kbit/s, 125kbit/s, 250kbit/s, 500kbit/s, 800kbit/s, 1Mbit/s，其它波特率可以定制
- 最大连接数：126 个从站
- 接收 PDO 数量：512
- 发送 PDO 数量：512
- 供电：24VDC(±5%)，最大功率 3.5W
- 工作环境温度：-25～55℃，湿度≤95%
- 外形尺寸：34mm（宽）×110mm（高）×70mm（厚）
- 安装方式：35mm 导轨
- 防护等级：IP20

3 产品外观

3.1 产品外观

3.2 指示灯定义

指示灯定义如下：

状态\灯	PWR	RUN
亮	电源接通	配置下载成功
灭	电源故障	无配置工程
闪烁		

CANopen 指示灯定义如下：

指示灯	状态	含义
RUN	绿灯常亮	节点处于运行状态
	绿灯周期性亮 200ms、灭 1000ms	节点处于停止状态
	绿灯周期性亮 200ms、灭 200ms	节点处于预运行状态
ERR	绿灯灭	CANopen 网络正常
	绿灯周期性亮　200ms、灭 1000ms	CANopen 控制器的错误计数器达到或超过警戒值
	绿灯常亮	BusOff

EtherCAT 网络指示灯（第 2 排）定义如下：

绿色 RUN LED 指示灯指示以下控制器状态

LED 指示灯行为	状态	先决条件
永久熄灭	INIT	从站处于初始化状态
闪烁	预操作	从站处于预操作状态
单次闪烁	安全操作	从站处于安全操作状态
快闪	初始化或启动	从站处于初始化阶段，尚未进入 INIT 状态或从站处于引导加载程序模式，且正在加载固件
常绿	操作	从站处于操作状态

绿色 ERR LED 指示灯指示以下控制器状态

LED 指示灯行为	状态	先决条件
永久熄灭	无错误	没有错误
闪烁	无效配置	一半配置错误
单次闪烁	本地错误	从站由于本地错误而自动更改其 EtherCAT 状态
两次闪烁	进程数据监视器超时/EtherCAT 监视器超时	监视器超时
三次闪烁	应用程序错误	从站的电源状态机出现错误
快闪	启动错误	加载固件时出错
常绿	应用控制器故障	从站不再响应

3.3 通讯端口

3.3.1 电源端口

引脚	功能
1	24V+，直流 24V 电源正，范围 9-30V
2	0V，直流 24V 电源负
3	PE，地

3.3.2 RS485端口

引脚	功能
1	CAN-H
2	CAN-L
3	GND,保护地

红色拨码开关用于设置终端匹配功能，当开关拨到“ON”时，启用终端匹配功能，当开关拨到“OFF”时，禁止终端匹配功能。终端匹配电阻为 220 欧姆。

CAN 传输技术特征：

网络拓扑：线性总线，两端有有源的总线终端电阻；

传输速率：10kbit/s~1Mbit/s；

介质：屏蔽双绞电缆，也可取消屏蔽，取决于环境条件（EMC）；

站点数：每分段 32 个站（不带中继），可多到 127 个站（带中继）；

插头连接：3 针可插拔端子；

CAN 传输设备安装要点：

本模块 CAN 可在配置软件单独配置；

总线的最远两端各有一个总线终端电阻，确保网络可靠运行；

4 使用方法

4.1 配置模块

1. 正确连接电源，通过配置口将 ECAT – CANopen 与 PC 相连，给 ECAT – CANopen 上电；
2. 打开配置软件，根据需求在配置软件中进行配置（请参考配置软件的使用方法）；
3. 将配置下载到 ECAT – CANopen 中；
4. 连接主站控制器与网关模块；
5. 等待 ECAT – CANopen 会与 PLC 之间建立连接，此时 ECT 绿色指示灯常亮；
6. 网关模块 CANopen 侧通讯正常后，RUN 绿色指示灯常亮；

4.2 PLC 模块参数设置步骤

以 TwinCAT3 为例展开如何添加 ECAT- CANOPEN 模块；

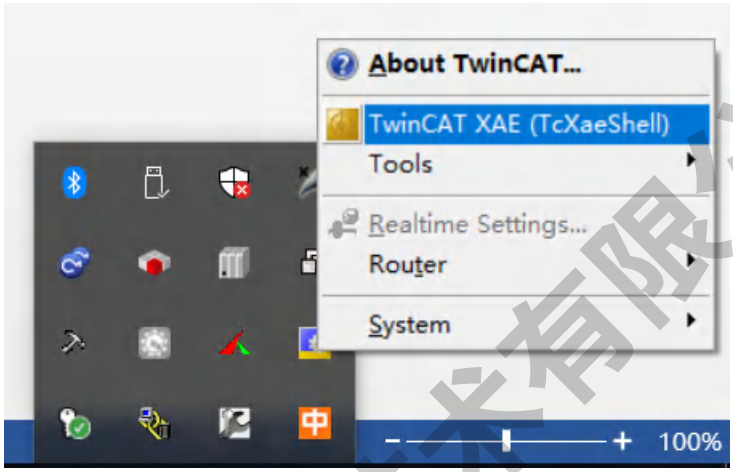
4.2.1 导入XML文件

将 ECAT-CANopen 模块从站的 XML 文件复制粘贴至 TwinCAT3 安装目录（默认安装路径）：C:\TwinCAT\3.1\Config\Io\EtherCAT；

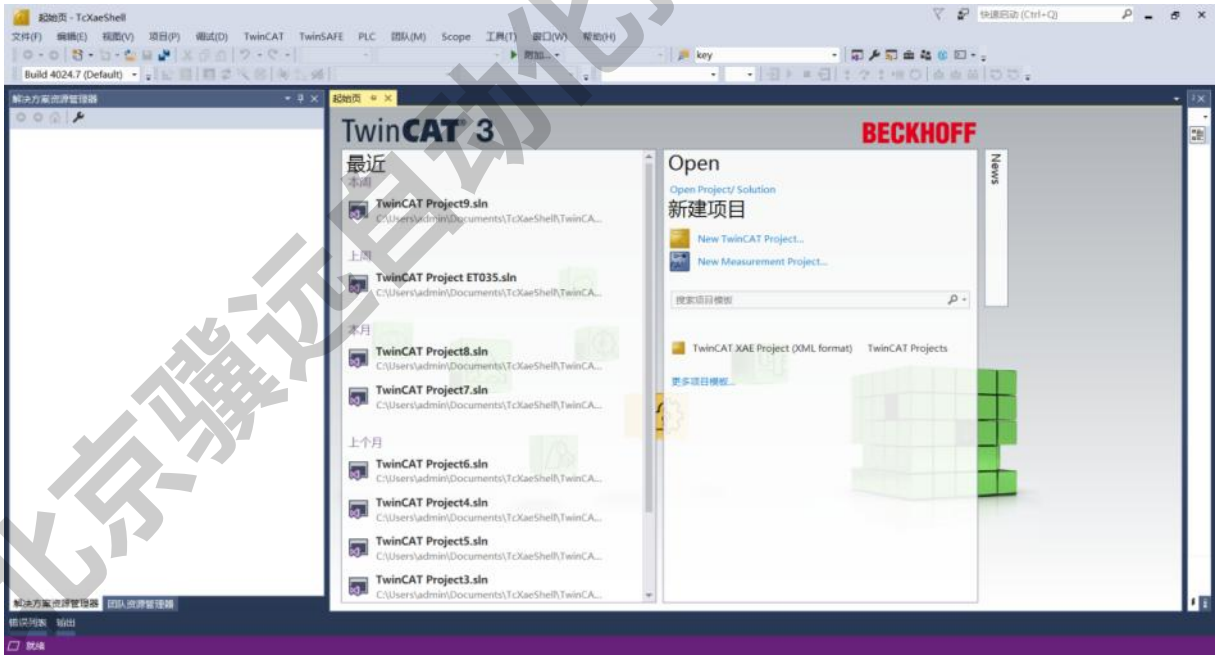
4.2.2 配置ECAT - CANopen模块

4.2.2.1 新建工程

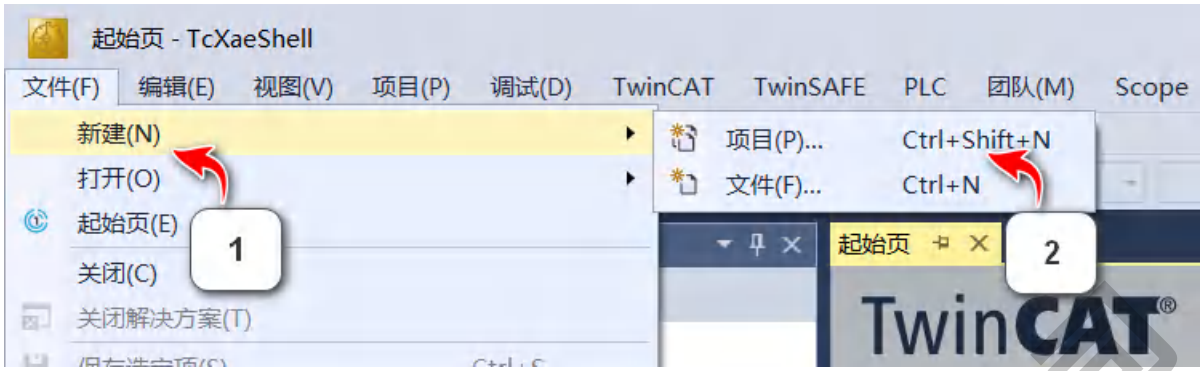
打开安装好的 TC3 软件，在电脑右下角右键点击 TC3 图标，选择“TwninCAT XAE(TcXaeShell)”



进入 VS 开发环境；

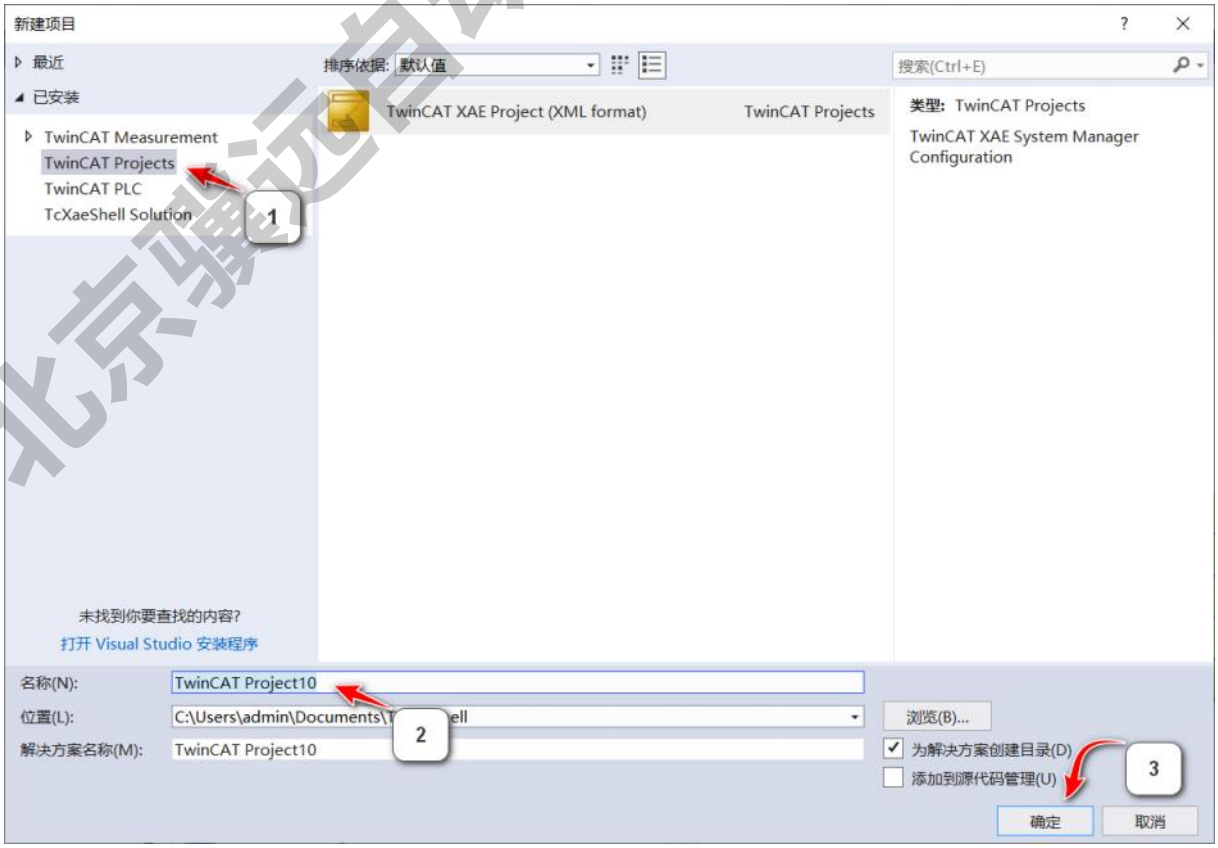


在 TC3 主菜单栏执行“文件”-“新建（N）”-“项目（P）...”；

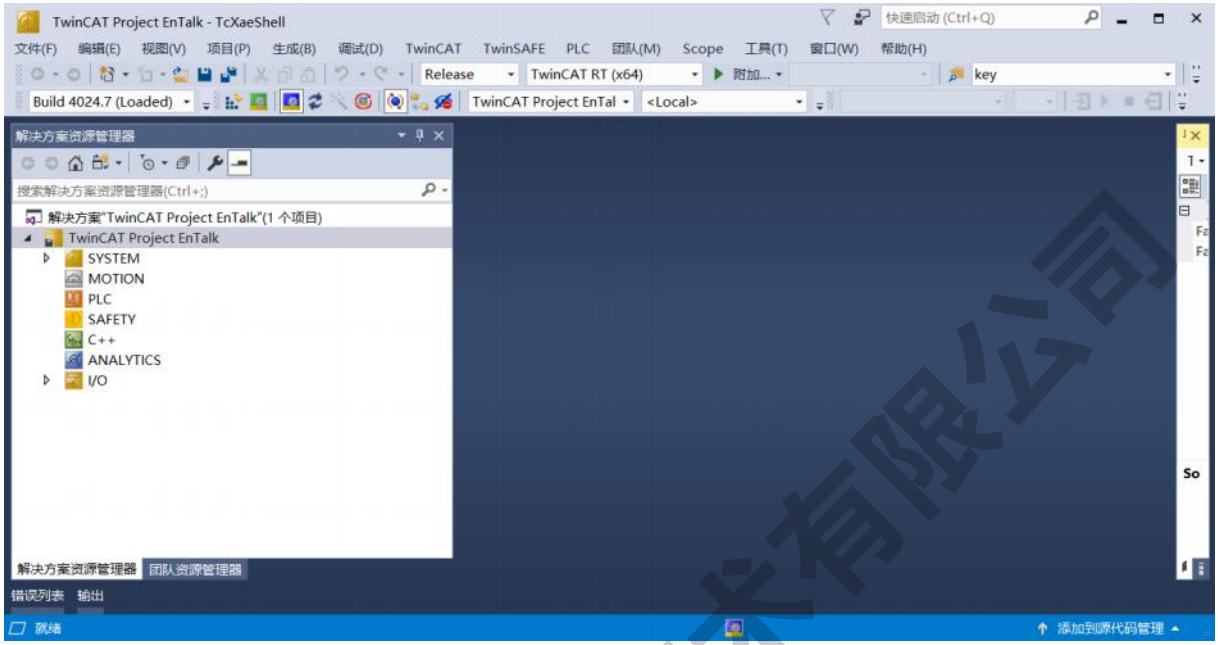


或者在 Open 界面下执行“New TwinCAT Project...”

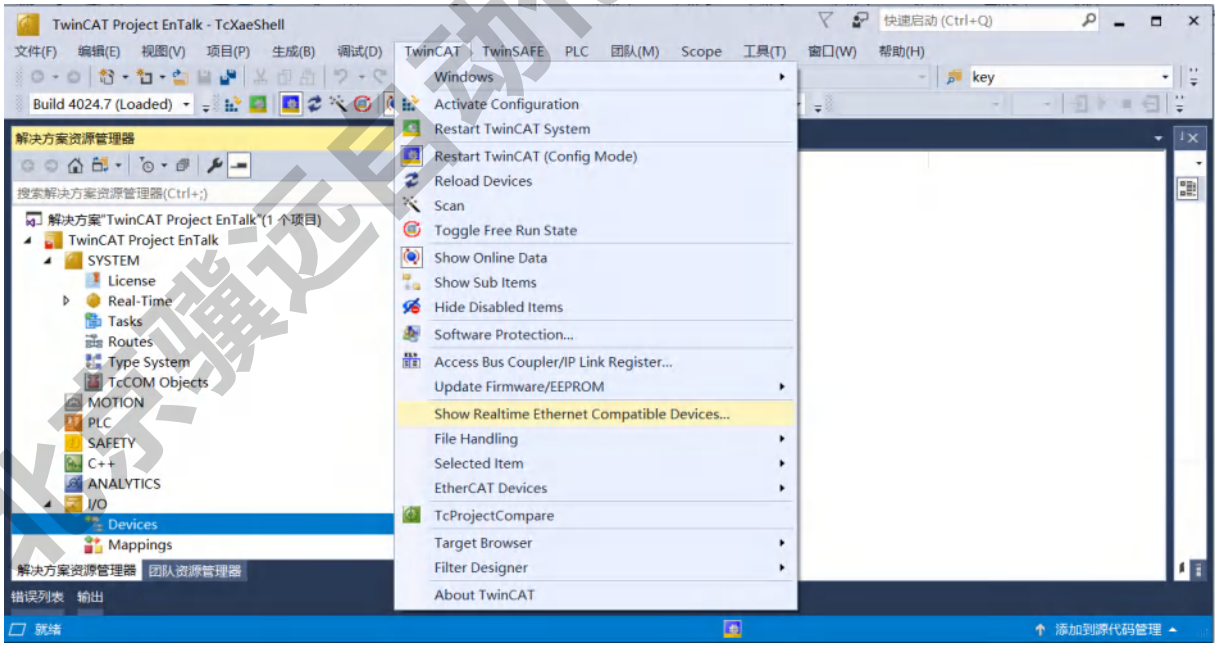
选择 TwinCAT Project，修改工程名称，点击“确定”；



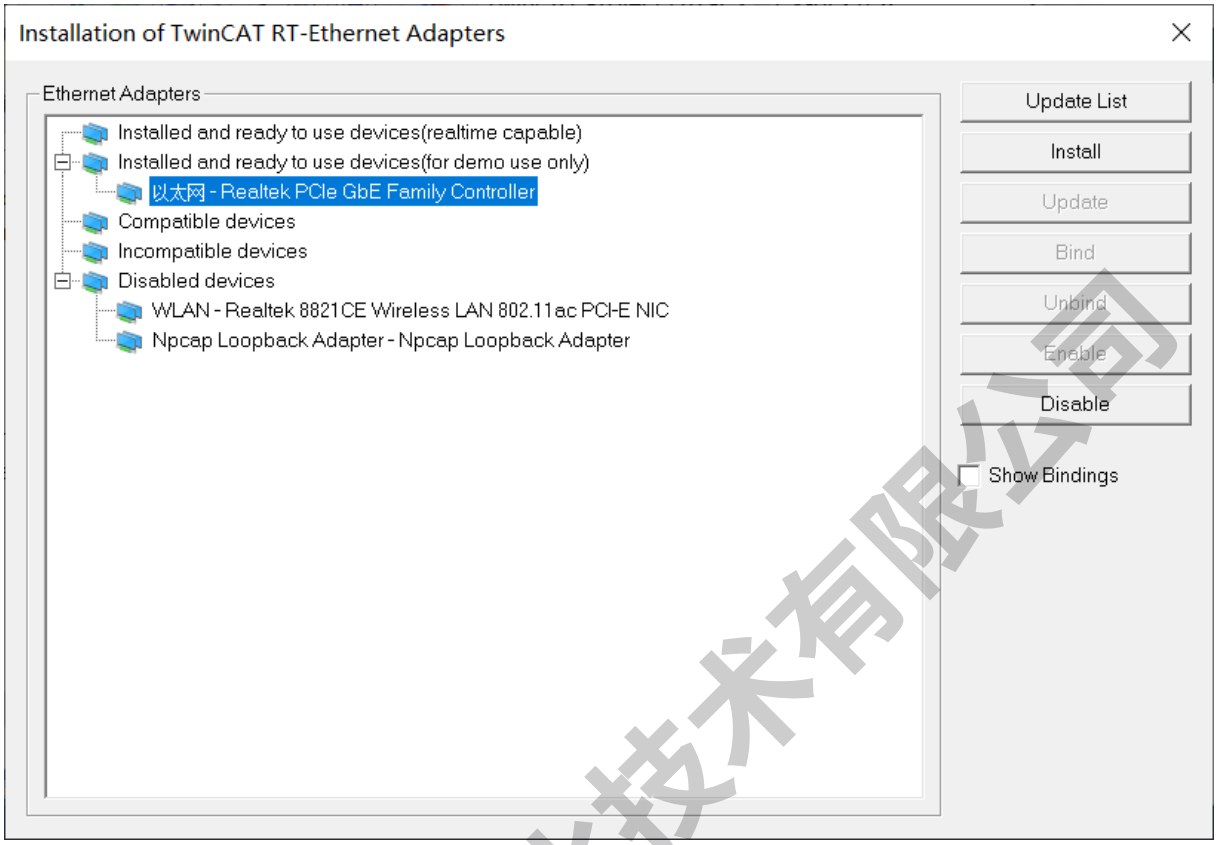
之后显示如下界面：



在 TC3 内安装 EtherCAT 主站网卡驱动，点击主菜单栏“TwinCAT”下的“Show Realtime Ethernet Compatible Devices…”，选择本机网卡，点击“Install”；

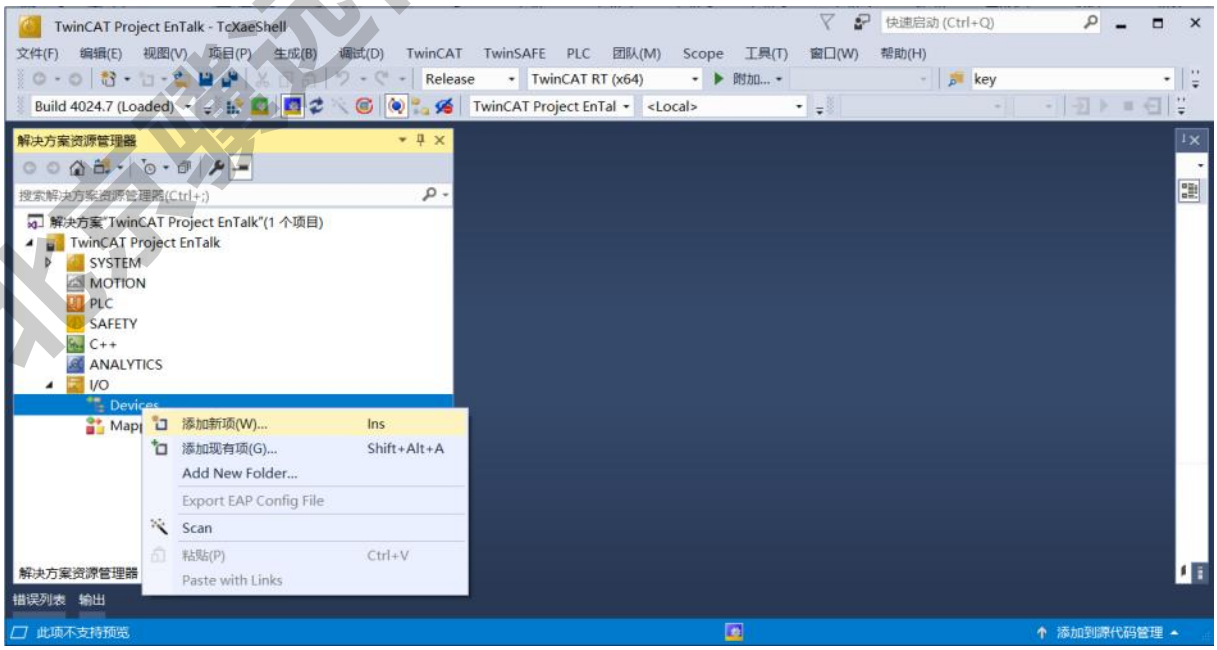


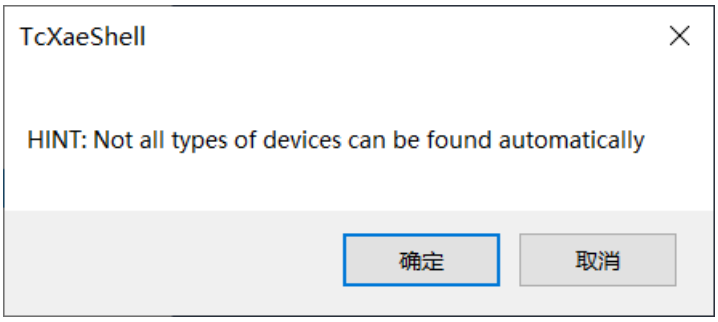
若成功安装网卡后显示如下界面：



4.2.2.2 扫描从站设备

在上图中“I/O”展开“Devices”的位置，点击鼠标右键选择“Scan”扫描连接的从站或者选中“Devices”后点击扫描按钮“ ”；

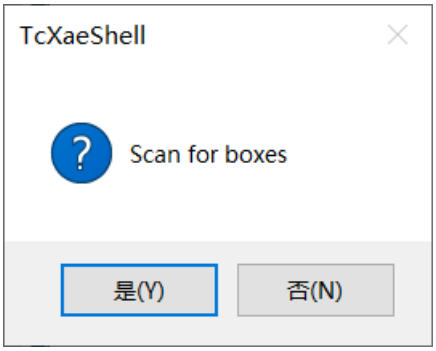




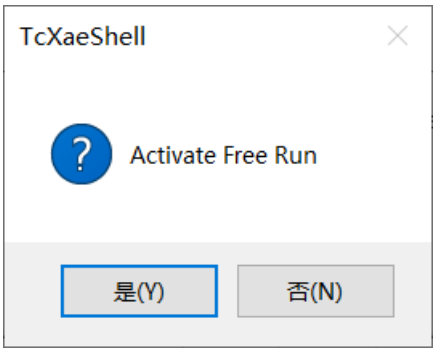
点击“确定”按钮，弹出“1 new I/O devices found”对话框，选择所需要的 Ethernet 接口，点击“OK”；

如下图所示：

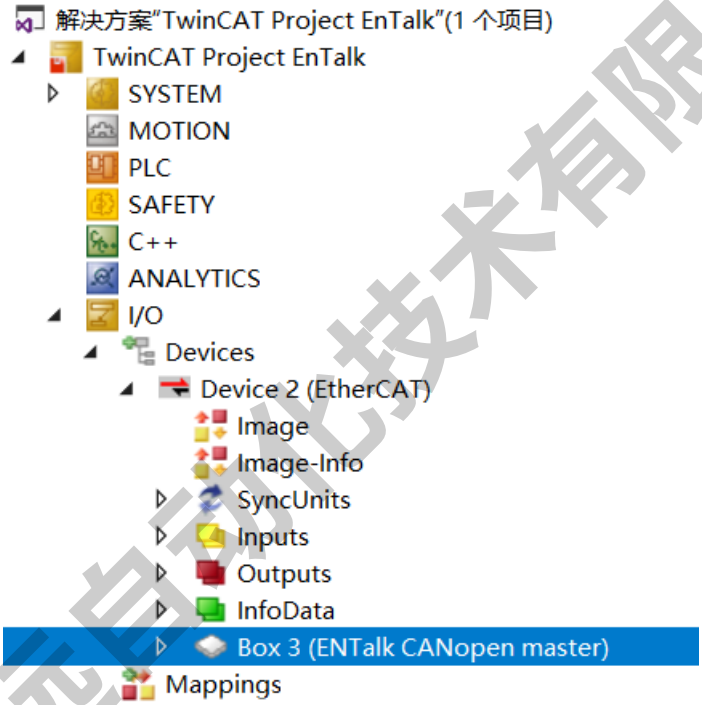
点击“是（Y）”将扫描从站设备；



选择是否进入“Activate Free Run”,点击“否”；

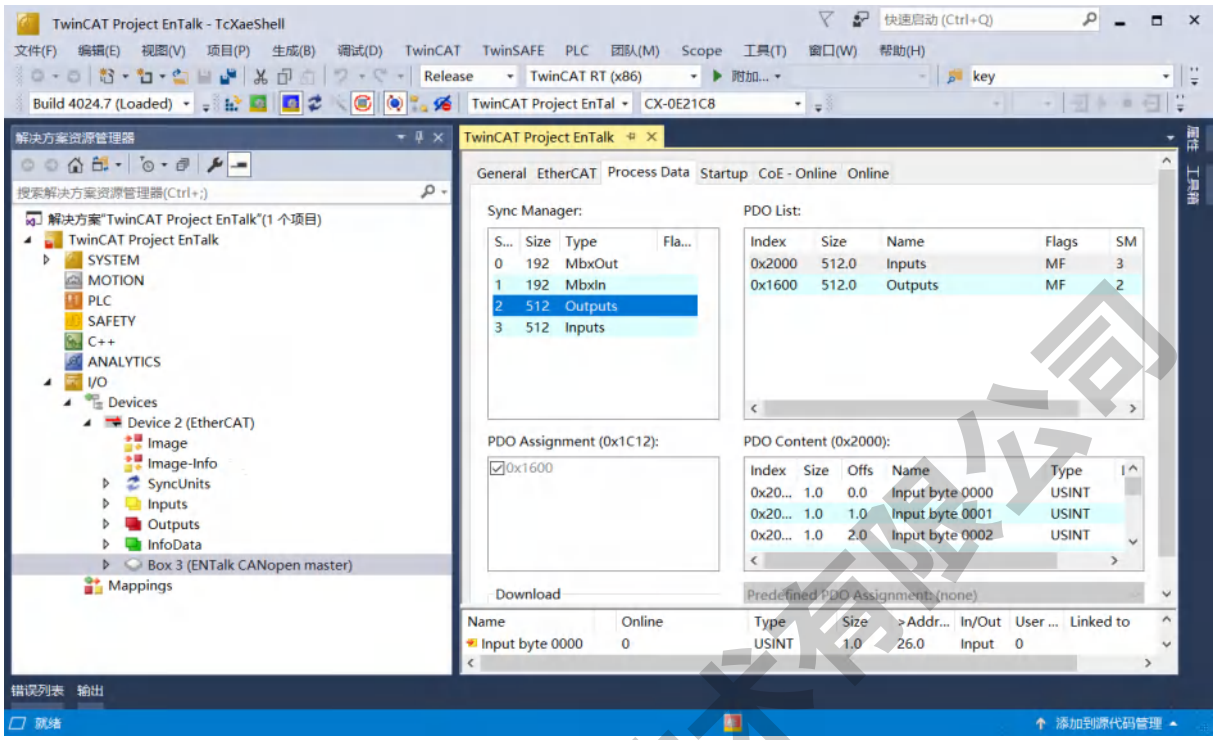


正常扫描到如下设备：

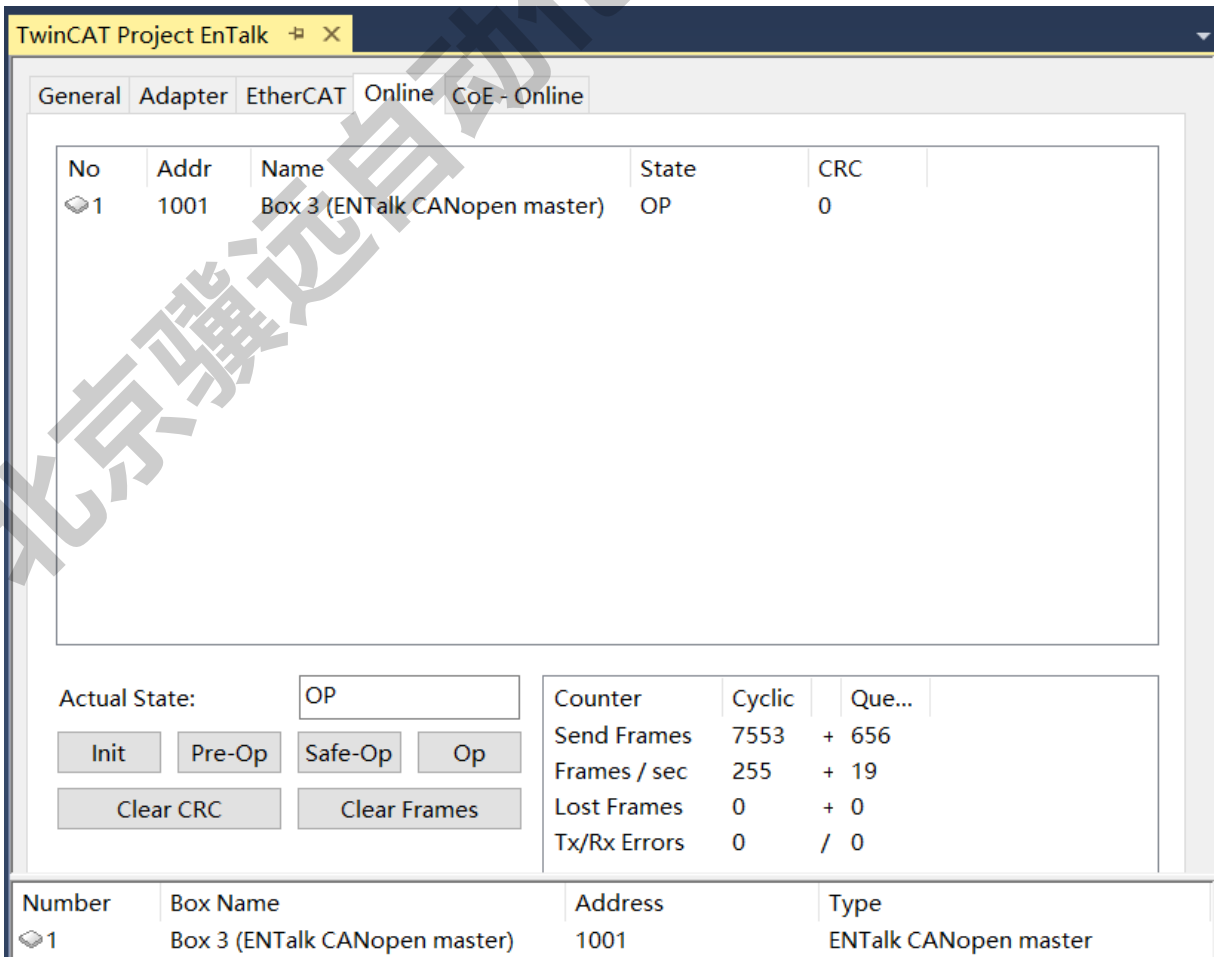


4.2.2.3 过程数据

在如下窗口“Process Data”选项页中，将清楚的看到 TC3 已经分配好 I/O 数据给 ECAT- CANopen 模块；MbxOut 和 MbxIn 的大小不需要修改；“Size”表示数量用于访问数据的命令字节长度；即 CANopen 网络数据访问的输入输出数据长度；

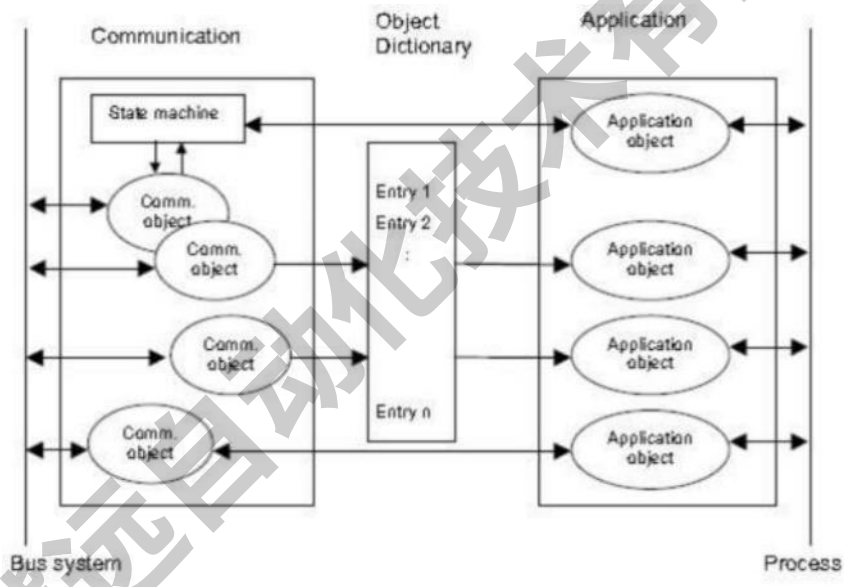


运行 TC3 软件，主站控制器与 ECAT- CANopen 模块建立通讯后，可看到从站设备已经进入“OP”状态；说明 EtherCAT 连接建立成功；



4.3 CANopen 介绍

CANopen 协议是由 CiA 协会针对 CAN 协议的不完整性而定义出来的一个更高层次的协议——应用层协议。通信接口和协议软件提供在总线上收发通信对象的服务。不同 CANopen 设备间的通信都是通过交换通信对象完成的。这一部分直接面向 CAN 控制器进行操作。对象字典描述了设备使用的所有的数据类型，通信对象和应用对象。对象字典位于通信程序和应用程序之间，向应用程序提供接口，应用程序对对象字典进行操作就可以实现 CANopen 通信。应用程序包括功能部分和通信部分，通信部分通过对对象字典进行操作实现 CANopen 通信，而功能部分则根据应用要求实现。



4.3.1服务资料对象(SDO)

服务资料对象(SDO)：可用来存取远端节点的对象字典，读取或设定其中的资料。提供对象字典的节点称为 SDO server，存取对象字典的节点称为 SDO client。SDO 通讯一定由 SDO client 开始，并提供初始化相关的参数。

CANopen 的术语中，上传是指由 SDO server 中读取资料，而下载是指设定 server 的资料。

SDO 通过使用索引和子索引（在 CAN 报文的前几个字节），SDO 使客户机能够访问设备（服务器）对象字典中的项（对象）。

SDO 通过 CAL 中多元域的 CMS 对象来实现，允许传送任何长度的数据（当数据超过 4 个字节时分拆成几个报文）。

协议是确认服务类型：为每个消息生成一个应答（一个 SDO 需要两个 ID）。SDO 请求和应答报文总是包含 8 个字节（没有意义的数据长度在第一个字节中表示，第一个字节携带协议信息）。SDO 通讯有较多的协议规定。

各种传输帧格式：

SDO 用来访问一个设备的对象字典。访问者被称作客户(client)，对象字典被访问且提供所请求服务的 CANopen 设备别称作服务器(server)。客户的 CAN 报文和服务器的应答 CAN 报文总是包含 8 字节数据（尽管不是所有的数据字节都一定有意义）。一个客户的请求一定有来自服务器的应答。

4.3.2过程数据对象(PDO)

PDO 用来传输实时数据，数据从一个生产者传到一个或多个消费者。数据传送限制在 1 到 8 个字节（例如，一个 PDO 可以传输最多 64 个数字 I/O 值，或者 4 个 16 位的 AD 值）。PDO 通讯没有协议规定。PDO 数据内容只由它的 CAN ID 定义，假定生产者和消费者知道这个 PDO 的数据内容。

每个 PDO 在对象字典中用 2 个对象描述：

PDO 通讯参数：包含哪个 COB-ID 将被 PDO 使用，传输类型，禁止时间和定时器周期。

PDO 映射参数：包含一个对象字典中对象的列表，这些对象映射到 PDO 里，包括它们的数据长度(bits)。生产者和消费者必须知道这个映射，以解释 PDO 内容。

PDO 消息的内容是预定义的（或者在网络启动时配置的）：

映射应用对象到 PDO 中是在设备对象字典中描述的。如果设备(生产者和消费者)支持可变 PDO 映射，那么使用 SDO 报文可以配置 PDO 映射参数。

PDO 可以有多种传送方式：

同步非周期：由远程帧预触发传送，或者由设备子协议中规定的对象特定事件预触发传送。

同步周期：传送在每 1 到 240 个 SYNC 消息后触发。

异步远程帧触发传送。

异步由设备子协议中规定的对象特定事件触发传送。

一个 PDO 可以指定一个禁止时间，即定义两个连续 PDO 传输的最小间隔时间，避免由于高优先级信息的数据量太大，始终占据总线，而使其它优先级较低的数据无力竞争总线的问题。禁止时间由 16 位无符号整数定义，单位 100us。

一个 PDO 可以指定一个事件定时周期，当超过定时时间后，一个 PDO 传输可以被触发（不需要触发位)。事件定时周期由 16 位无符号整数定义，单位 1ms。

4.3.3 NMT模块控制

只有 NMT-Master 节点能够传送 NMT Module Control 报文。所有从设备必须支持 NMT 模块控制服务。

NMT Module Control 消息不需要应答。NMT 消息格式如下：

NMT-Master →NMT-Slave(s)

COB -ID	Byte0	Byte1
0x00	CS	Node-ID

当 Node-ID=0，则所有的 NMT 从设备被寻址。CS 是命令字，可以取如下值：

命令字	NMT 服务
1	Start Remote Node
2	Stop Remote Node
128	Enter Pre-operational State
129	Reset Node
130	Reset Communication

4.3.4NMT节点保护（NMT Node Guarding）

通过节点保护服务，MNT 主节点可以检查每个节点的当前状态，当这些节点没有数据传送时这种服务尤其有意义。

NMT-Master 节点发送远程帧（无数据）如下：

NMT-Master →NMT-Slave

COB -ID
0x700 + Node-ID

NMT-Slave 节点发送如下报文应答：

NMT-Master ←NMT-Slave

COB -ID	Byte0
0x700 + Node-ID	Bit7：toggle Bit6-0：状态

数据部分包括一个触发位（bit7），触发位必须在每次节点保护应答中交替置“0”或者“1”。触发位在第一次节点保护请求时置为“0”。位 0 到位 6（bits0～6）表示节点状态，可为下表中的数值。

Value	状态
0	Initialising
1	Disconnected
2	Connecting
3	Preparing
4	Stopped
5	Operational
127	Pre-operational

4.3.5心跳报文（Heartbeat）

Heartbeat Producer → Consumer(s)

COB -ID	Byte0
0x700 + Node-ID	状态

状态可为下表中的数值：

状态	意义
0	Boot-up
4	Stopped
5	Operational
127	Pre-operational

当一个 Heartbeat 节点启动后它的启动报文是其第一个 Heartbeat 报文。Heartbeat 消费者通常是 NMT-Master 节点，它为每个 Heartbeat 节点设定一个超时值，当超时发生时采取相应动作。

一个节点不能够同时支持 NodeGuarding 和 Heartbeat 协议。

4.3.6NMT Boot-up

NMT-slave 节点发布 Boot-up 报文通知 NMT-Master 节点它已经从 initialising 状态进入 pre-operational 状态。

NMT-Master ←NMT-Slave

COB -ID	Byte0
0x700 + Node-ID	0

4.3.7应急指示对象：

应急指示报文由设备内部出现的致命错误触发，由相关应用设备已最高优先级发送到其它设备。

适用于中断类型的错误报警信号。

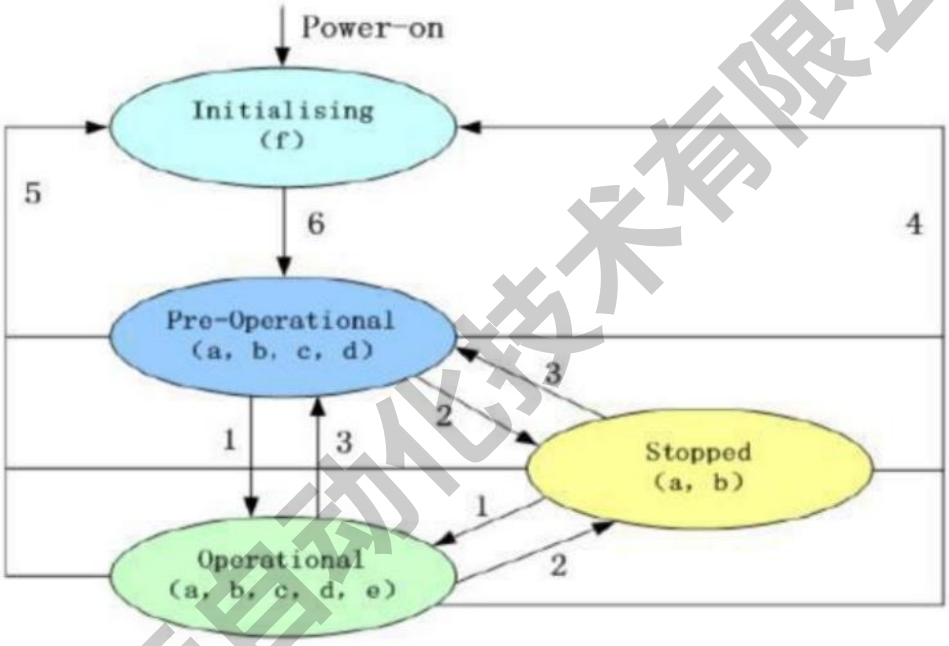
一个应急报文由 8 字节组成，格式如下：

sender → receiver(s)

COB -ID	Byte0-1	Byte2	Byte3-7
0x800 + Node-ID	应急错误代码	错误寄存器 （对象 0x1001）	制造商特定的错误区域

4.3.8NMT状态控制过程：

CANopen 启动过程，可以用节点状态转换图表示这两种初始化过程，如下图所示。



注意：

图中括号内的字母表示处于不同状态那些通讯对象可以使用。

a.NMT，b.NodeGuard，c.SDO，d.Emergency，e.PDO，f.Boot-up

状态转移（1 - 5 由 NMT 服务发起），NMT 命令字（在括号中）：

1：Start_Remote_Node (0x01)

2：Stop_Remote_Node (0x02)

3：Enter_Pre-Operational_State (0x80)

4：Reset_Node (0x81)

5：Reset_Communication (0x82)

6：设备初始化结束，自动进入 Pre_Operational 状态，发送启动消息

在任何时候NMT服务都可使所有或者部分节点进入不同的工作状态。NMT服务的CAN报文由CAN头(COB-ID=0)和两字节数据组成；第一个字节表示请求的服务类型(‘NMTcommandspecifier’)，第二个字节是节点 ID，或者 0（此时寻址所有节点)。

设备进入准备状态后，除了 NMT 服务和节点保护服务（如果支持并且激活的话）外，将停止通讯。

4.4 配置软件

配置模块需要使用配置软件，用户可以从光盘或者网站上获取并安装，用户使用网关配置软件可以轻松完成 PN - CANopen 的配置，从上到下分别分标题栏、菜单栏、工具栏、设备树，以及配置区，下面详细说明这 5 部分的功能，主界面如下图：

4.5 运行

4.5.1数据交换

ECAT - CANopen 的 EtherCAT 网络和 CANopen 网络之间的数据转换是通过“映射”关系来建立的。在 ECAT - CANopen 中有两块数据缓冲区，一块是输入缓冲区（1.5K 字节），地址范围为 0x000-0x5DC；另一块是输出缓冲区（1.5K 字节），地址范围同样为 0x000-0x5DC。

4.5.2 EtherCAT从站

假定用户配置的输入数据的长度为 L1，输出数据的长度为 L2。ECAT - CANopen 会把[0x000,L1]地址范围内的数据发送到 EtherCAT 网络中，当从 EtherCAT 网络接收到数据是，ECAT- CANopen 会将数据写到[0x000,L2]地址范围内。

4.6 软件安装

在安装 CANopen Configuration Tools（以下简称 CCT）软件时，推荐使用的计算机配置如表所示。

环境	类型	型号
硬件环境	显示器	彩色 CRT
	输入输出	标准键盘，鼠标
	USB 接口	至少一个 2.0 接口
	显卡	分辨率支持 1280×1024
	CPU	Intel Pentium 2.4GHz 以上
	内存	512M 以上
	硬盘	10G 以上
软件环境	操作系统	Windows7
	应用软件	CANopen 配置套件 V2.3.2.8

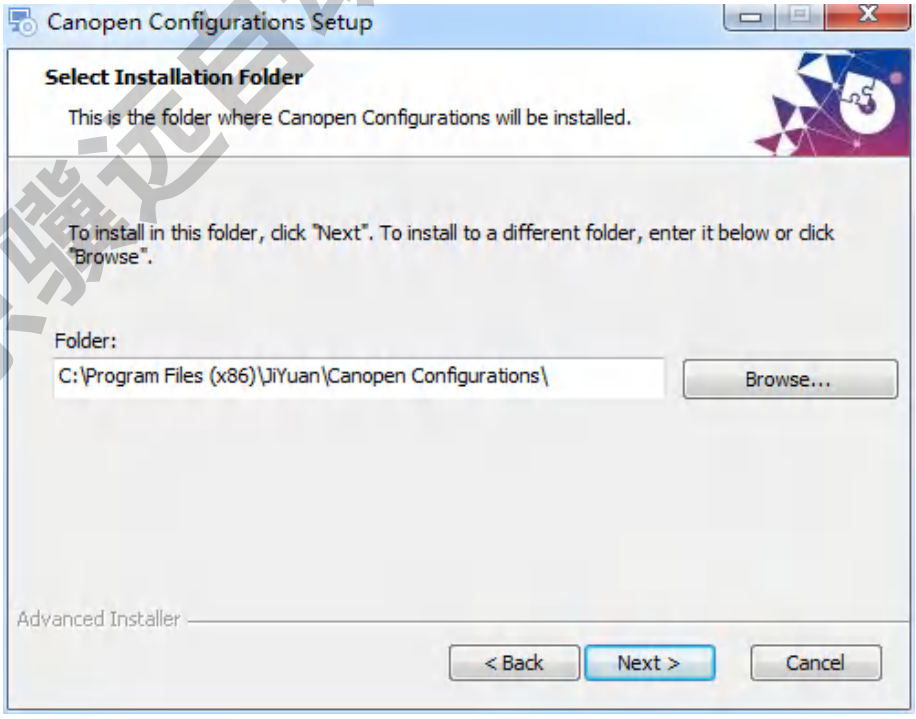
安装 CCT 软件的主要步骤如下所述。

第 1 步 启动安装向导

双击安装包，弹出如下图，点击下一步：

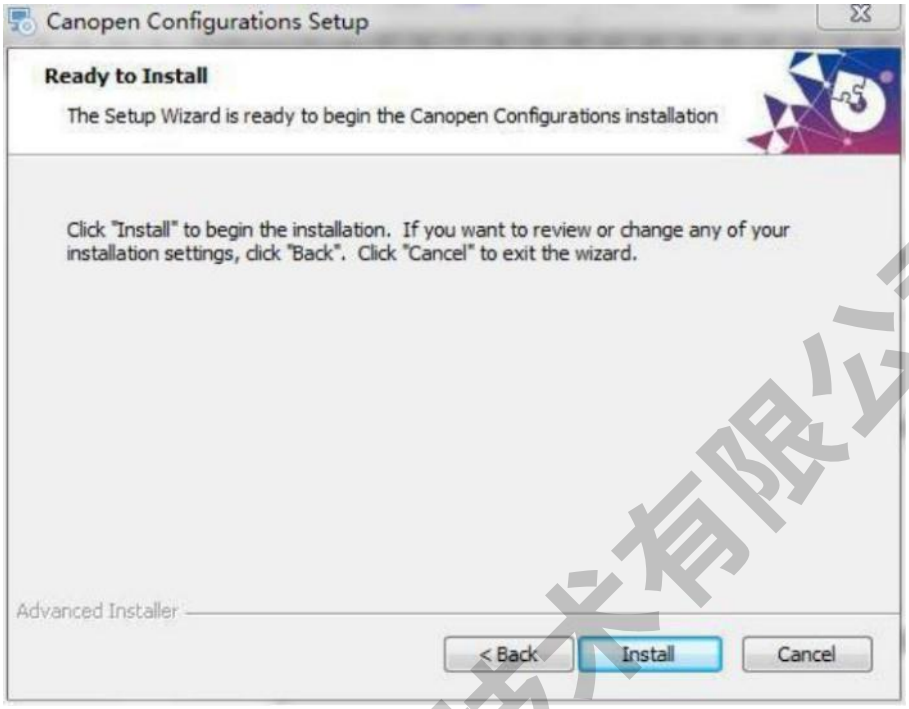
第 2 步 选择安装位置

选择安装位置，点击下一步：



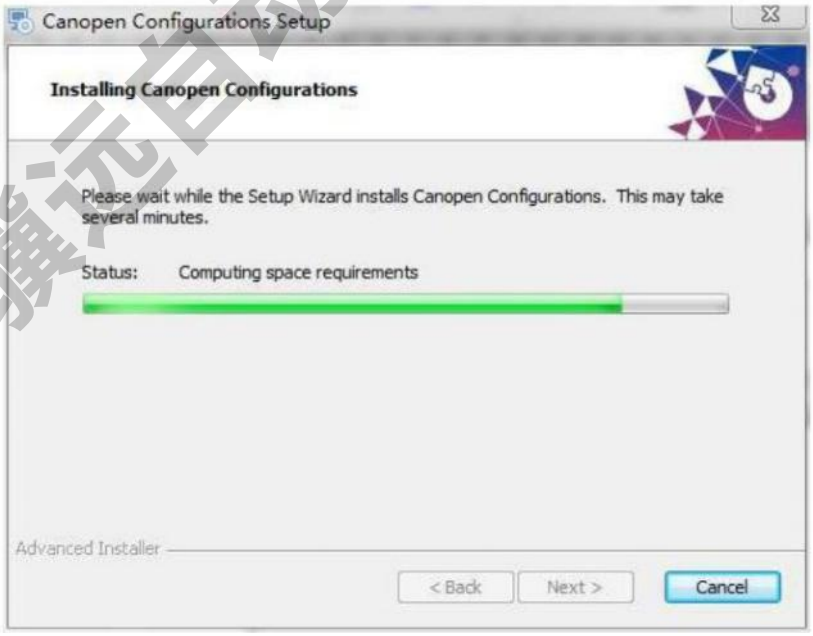
第 3 步 开始安装

选择安装，点击开始安装：



第 4 步 正在安装

安装过程需要几分钟：



第 5 步 安装完成提示

安装完毕，弹出“CCT 安装完成向导”窗口。鼠标左键单击“完成”立即运行 CCT，如下图所示。

4.7 用户界面介绍

用户界面主要由以下部分构成，如图

4.8 标题栏

用于显示软件的名称，如果打开或保存了一个 CANopen 配置工程，也将显示其所在的路径。

> CanOpen Configuration - C:\Users\Administrator\Desktop\TEST.canproj

4.8.1菜单栏

提供软件所支持的工程文件操作，比如打开、保存；以及工程的下装设置等。

文件(F)　设备(D)　工程(P)　语言(L)　帮助(H)

4.8.2工具栏

用于快速访问菜单栏中的各个功能。

4.8.3设备树

用于组态需要的硬件设备，一共分为两个区域，分别是：

主站设备：列出组态到工程中的主要设备，主设备只能添加一个。

从站设备：列出组态到工程中的从属设备，从属设备可以添加多个。

4.8.4配置区

用于详细配置主设备与从设备的信息，只有当选择了对应的设备时，才会在配置区显示其信息，该区域包含如下 5 个页面：

通用属性：设备的概述与总体信息，比如设备名称、厂商信息。

显示主从站的逻辑名称，添加不同主从站类型模块，显示节点名称不一样；下列表格为主站逻辑名称对应的协议转换模块；该逻辑名称可在“设备参数”选项页进行修改；

协议转换类型	逻辑名称
PROFINET Slave to CANopen Master	ET005-PN2CM

PROFIBUS DP Slave to CANopen Master	ET025-PN2CM
EtherCAT Slave to CANopen Master	ET035-PN2CM
Modbus TCP Server to CANopen Master	ET045-PN2CM
Modbus RTU Slave to CANopen Master	ET065-PN2CM
CCLink IE Field Basic Slave to CANopen Master	ET075-CCIEFB2CM

节点 ID：显示设备的节点 ID，主站默认 0x7F；

波特率：指定 CANopen 通讯的波特率参数；

同步 Cobid：按照协议值为 0x80000080;

通讯周期：设置主站与从站同步周期时间，微秒；

错误控制：用于配置设备是工作在心跳模式，还是监听模式。

若采用心跳机制，CANopen 设备将根据“生产者时间”参数所设置的周期来发送心跳报文，该周期通常以 ms 为单位。用户还可在“消费者时间”参数设置被监视设备的节点 ID 和相应的事件周期。

生产者时间：

消费者时间：默认为 750ms；

如果采用节点/寿命保护，用户必须在主机中设置一个包含 CANopen 设备监视时间的表格。“守护时间”规定了两次查询之间的时间间隔，通常以 ms 为单位，“寿命因子”该系数与守护时间相乘所得到的时间，就是主机查询设备的最迟时间，例如下图守护时间=250ms，寿命因子 =4，250 x 4 =1000ms；有了这种机制，CANopen 设备识别 NMT 主机故障就有了保障。

通用属性 | 错误控制 | PDO 参数 | 设备参数 | SDO初始化 | 在线监视

节点Id	逻辑名称	生产者时间	消费者时间	守护时间	寿命因子
0x7F	ET035-ECAT2CM	0		0	0
1	SlaveLib 5.00	0	0	250	4
2	SlaveLib 5.00	0	0	250	4
3	SlaveLib 5.00	0	0	250	4
4	SlaveLib 5.00	0	0	250	4

PDO 参数：提供对 RPDO 与 TPDO 对象的参数控制。

设备参数：提供对设备的通信区域、制造商区域，以及标准化区域等内容的组态配置。

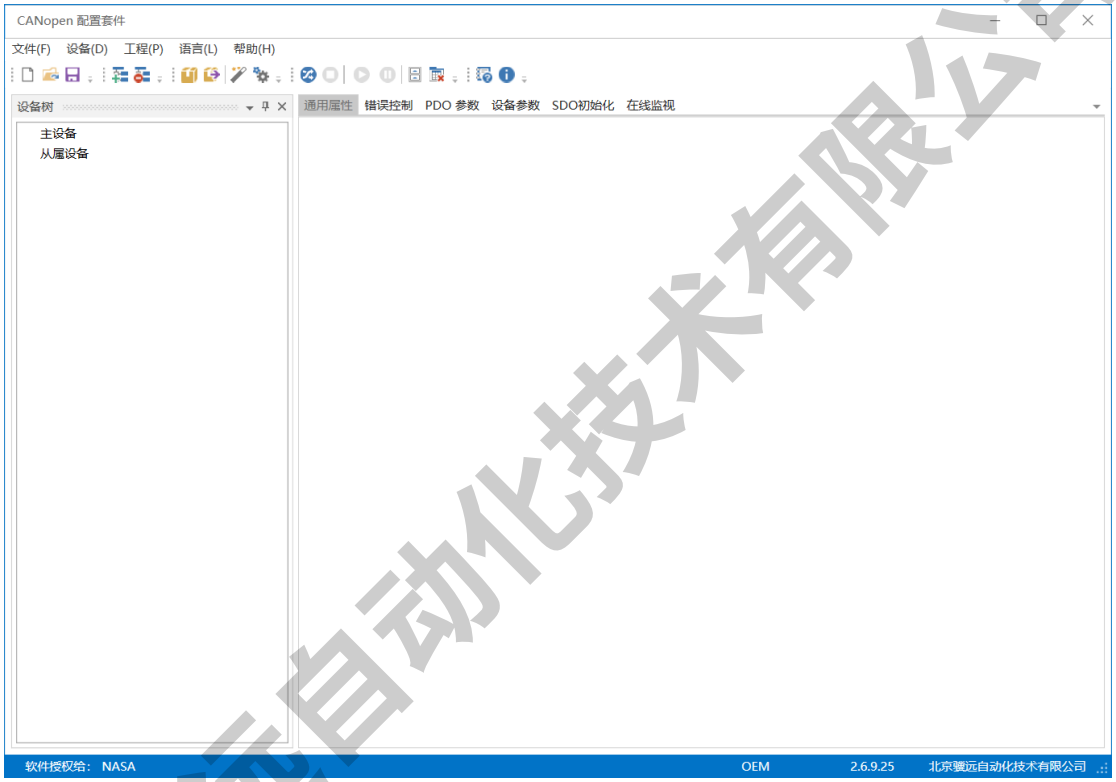
在线监视：监视数据。

5 开始使用

该部分将用一个最简单的流程来说明本软件的使用方法。

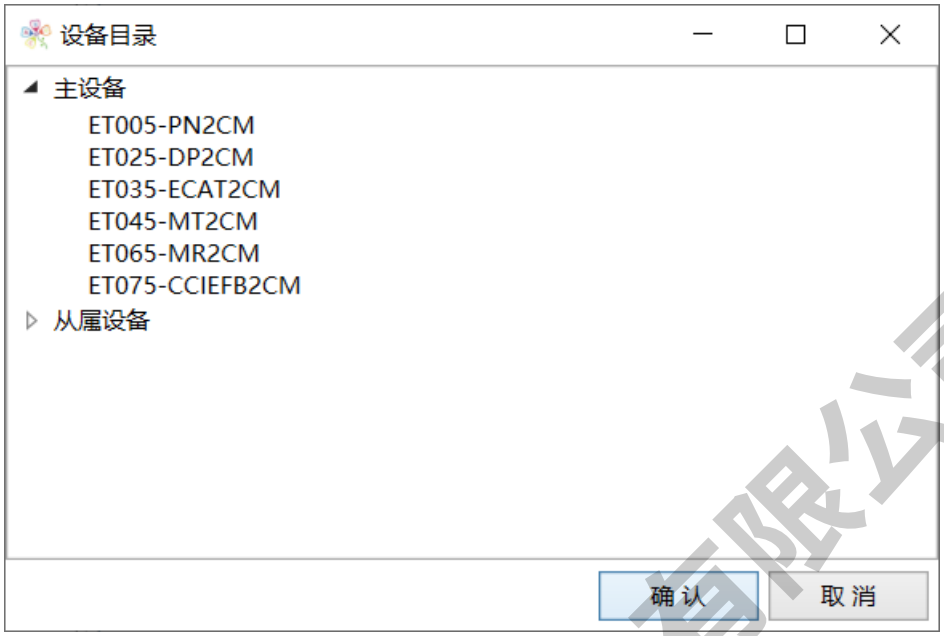
5.1.1创建新工程

启动 CANopen 配置套件，软件会自动创建一个空的工程，如图所示。



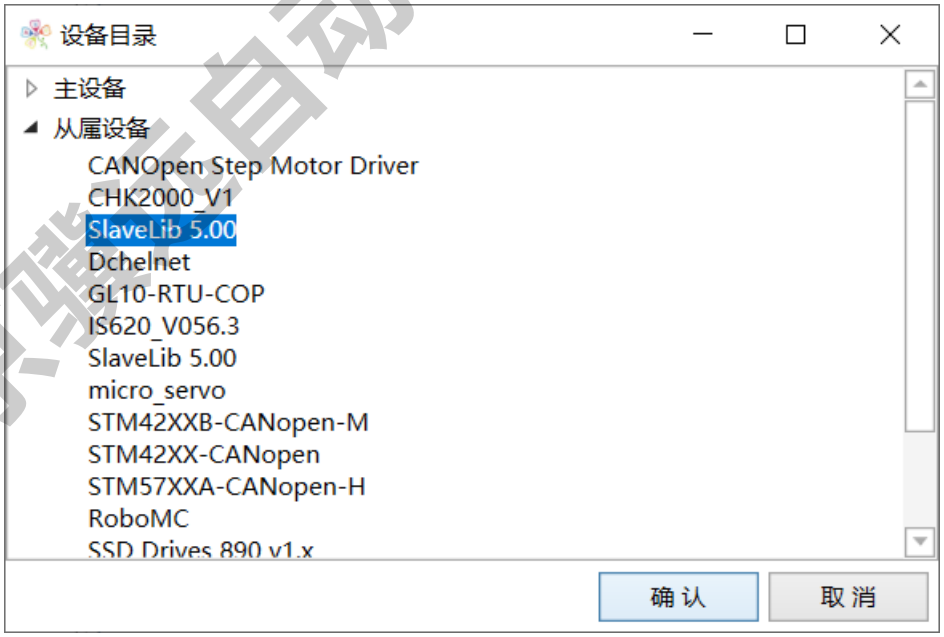
5.1.2添加主要设备

选中主设备，右击添加设备，弹出“设备目录”对话框，如图所示。然后从“主设备”列表中选择“ET035-ECAT2CM”作为主要设备，点击“确认”返回主界面。

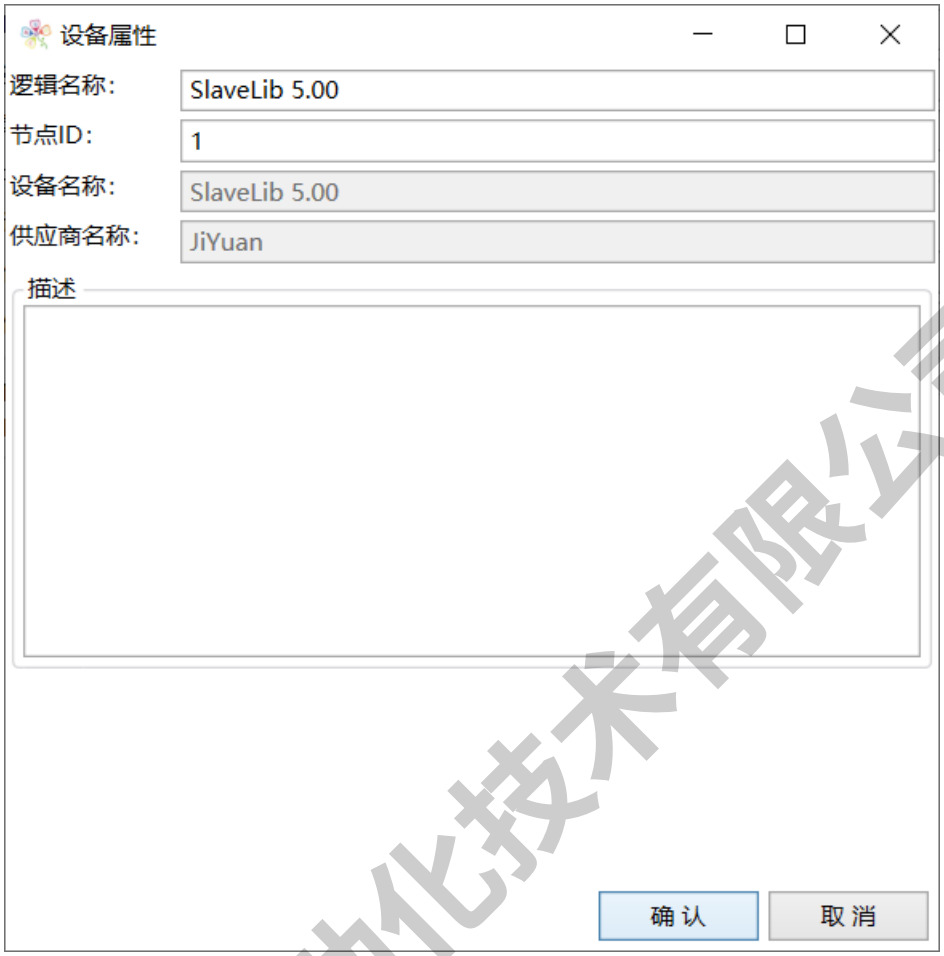


5.1.3添加从属设备

选中从属设备，右击添加设备，弹出“设备目录”对话框，如下图所示。例如添加 SlaveLib 5.00 作为从属设备。

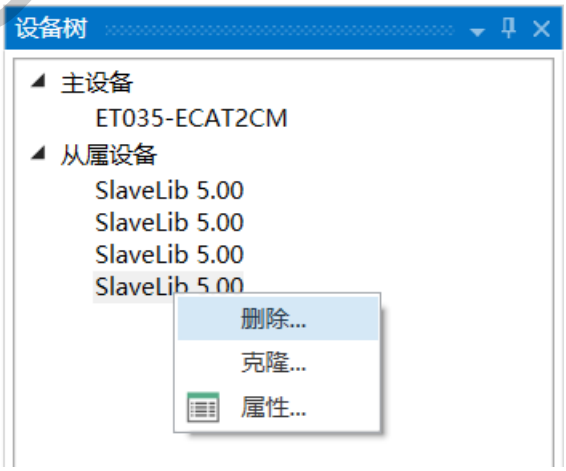


选择从属设备，点击“确认”后，弹出该从属设备的设备属性，可以修改节点 ID。



5.1.4删除从设备

单击左键选中设备，然后右击鼠标选中“删除”，如下图所示。删除 SlaveLib 5.00 从属设备。



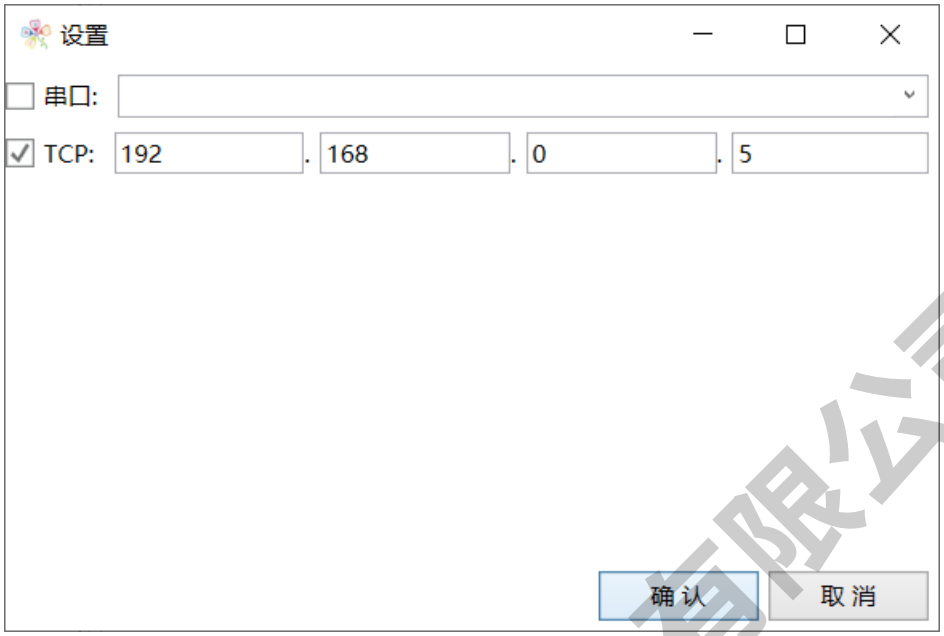
5.1.5克隆从设备

单击左键选中设备，然后右击鼠标选中“克隆”，如下图所示。

提示选择克隆从属设备的个数，每次最多支持克隆 10 个从站设备。点击“确认”快速完成多个从站组态；

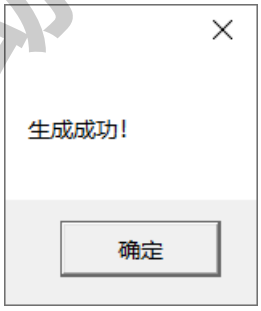
5.1.6配置下载口

单击工具栏上的“设置”按钮 ，弹出“设置”对话框，如图所示。选择要使用的下载方式，然后单击 OK 按钮返回主界面。



5.1.7生成下装文件

单击工具栏上的生成程序文件按钮 ，生成下装用的二进制文件。如果生成过程中没有错误，那么会弹出“生成成功!”确认框。



5.1.8下装

将生成的二进制文件，下载到硬件。如果下装时没有生成文件，那么会自动生成需要的文件。

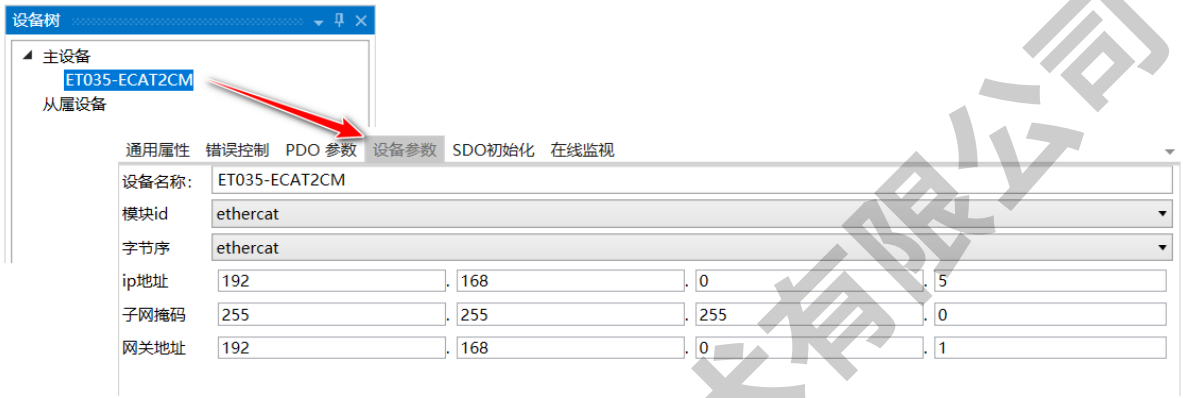
5.1.9保存工程

单击“文件”菜单中的“保存”，文件名为 xxxx，将当前工程保存到硬盘上，以便后续可以继续使用。这里默认的后缀名为.canproj，代表是一个 CANopen 的工程文件。

5.2 配置视图操作

5.2.1主设备配置参数

点击主站设备 ET035-ECAT2CM 的设备参数：



上述参数描述如下：

字节序：ethercat、canopen 可选

选择 ethercat：0x12，0x34，0x56，0x78 → 0x78，0x56，0x34，0x12

IP 地址：设备 IP 地址；

子网掩码：设备子网掩码；

网关地址：在局域网的网关地址；

注意：IP 地址、子网掩码、网关地址参数仅为以太网下载提供参数而配置。

5.2.2导入新的EDS文件

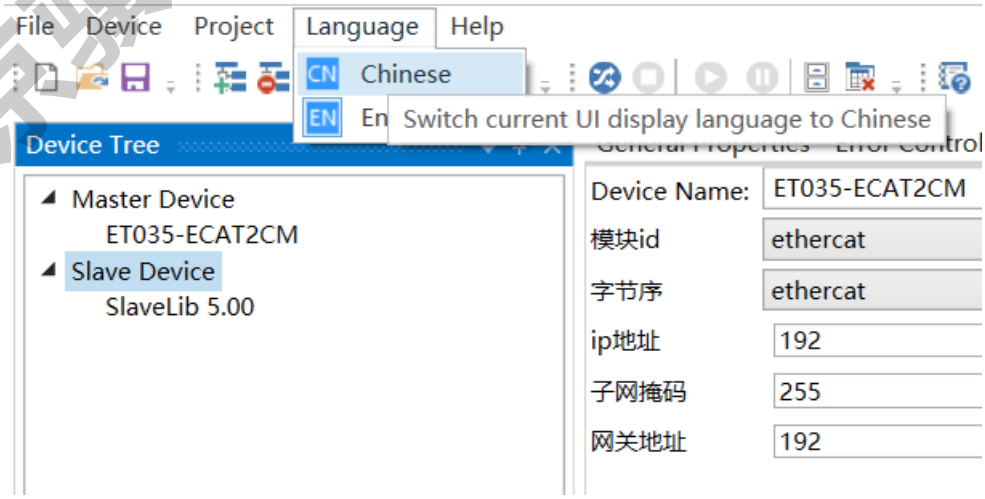
将 EDS 文件复制至安装目录下的“CANopen Configuration Studio\Devices\EDS”，比如采用默认的安装目录则复制至“C:\Program Files (x86)\JiYuan\CANopen Configuration Studio\Devices\EDS”目录下即可，重新打开软件。

或者通过主菜单“设备 – 导入 EDS”的方式导入新的 EDS 文件；

5.2.3中英文切换

此软件支持中文和英文两种语言，在“语言”中选择“英文（E)”，可以将软件切换为英文模式，如下图所示：

在“Language”中选择“Chinese”，可以将软件切换为中文模式；



5.2.4PDO参数

用来传输实时数据，数据从一个生产者传到一个或多个消费者。数据传输限制在 1 到 8 字节；

PDO 通讯没有协议规定。

每个 PDO 在对象字典中用到 2 个对象描述：

1） PDO 通讯参数：包含哪个 COB-ID 将被 PDO 使用，传输类型、传输速率、抑制时间和事件计时器；

COB-ID：Communication Object Identifier ,CAN ID;

传输类型:

同步：通过接收 SYNC 对象实现同步

非周期：由远程帧预触发传送，或者由设备子协议中规定的对象特定事件预触发传送。

周期：传送在每 1 到 240 个 SYNC 消息触发。

异步：

由远程帧触发传送。

由设备子协议中规定的对象特定事件触发传送。

PDO 传输类型与 PDO 触发模式对应表

传输类型	触发 PDO 的条件 B= both needed O=one or both			PDO 传输
	SYNC	RTR	Event	
Synchronous（acycilc）	B		B	同步，非周期
Synchronous（cycilc）	O			同步，循环周期
RTR-only（synchronous）	B	B		同步，在 RTR 之后
RTR-only（event-driven）		O		异步，在 RTR 之后
Event-driven（profile）		O	O	异步，设备子协议特定事件
Event-driven（manufacturer）		O	O	异步，制造商特定事件
SYNC：接收到 SYNC – object(同步对象) RTR：接收到远程帧 Event：数值改变或者定时器中断				

B 代表两个触发条件均满足时触发 PDO 传输， O 代表一个或者两个触发条件满足时均可触发 PDO 传输

传输速率：1-240，该数字代表两个 PDO 之间的 SYNC 对象的数目；

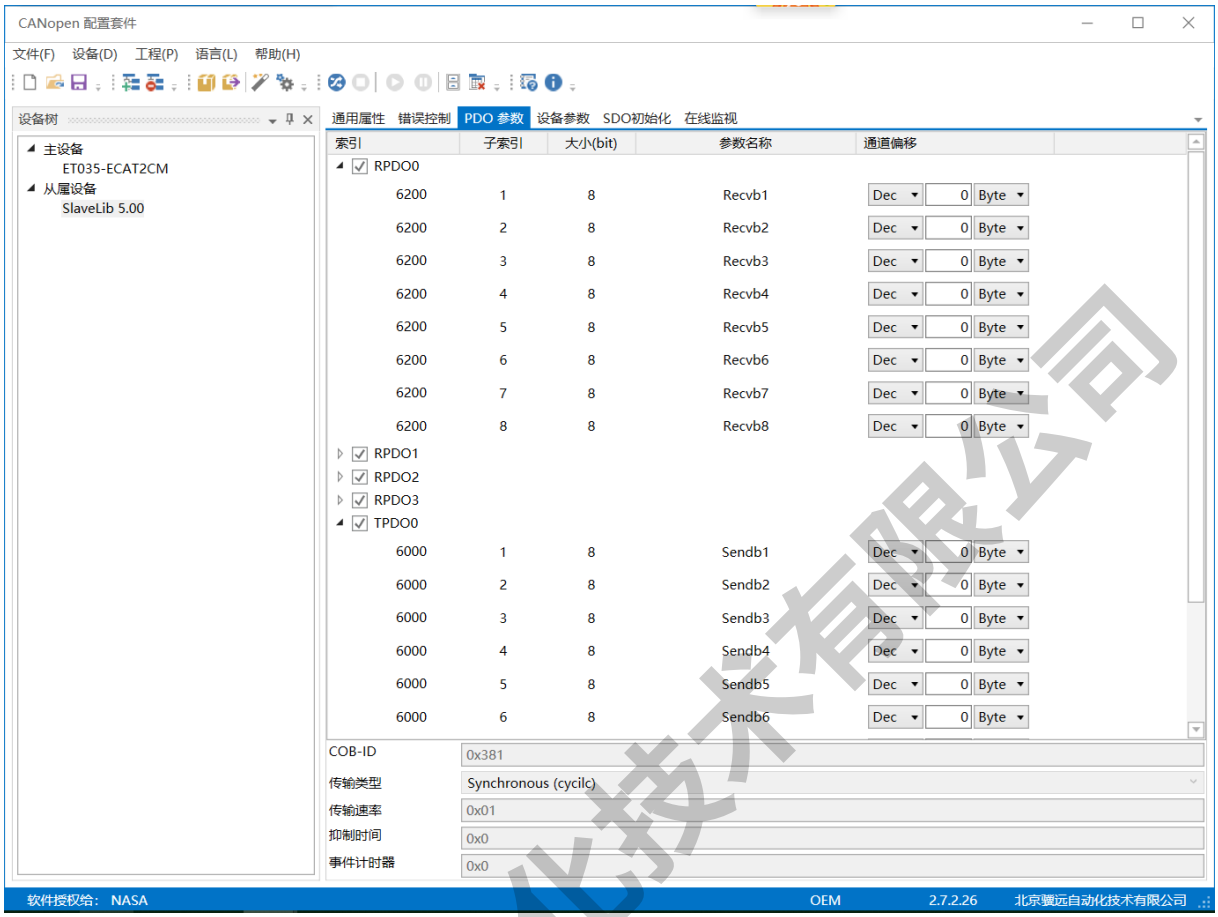
抑制事时间：一个 PDO 可以指定一个禁止时间，即定义两个连续 PDO 传输的最小间隔时间，避免由于高优先级信息的数量太大，始终占据总线，而使其它优先级较低的数据无力竞争总线的问题，单位 100 微秒；

事件计时器：一个 PDO 可以指定一个事件定时周期，当超过定时时间后，一个 PDO 传输可以被触发，单位为 1ms；

2） PDO 映射参数：包含一个对象字典中对象的列表，这些对象映射到 PDO 里，包括数据长度，生产者和消费者必须知道这个映射，用来解释 PDO 内容；

例如 RPDO1 的映射参数的对象字典索引为 6200h，子索引 01h 为映射到该 PDO 中待传输的数据所在的索引和子索引及数据长度。

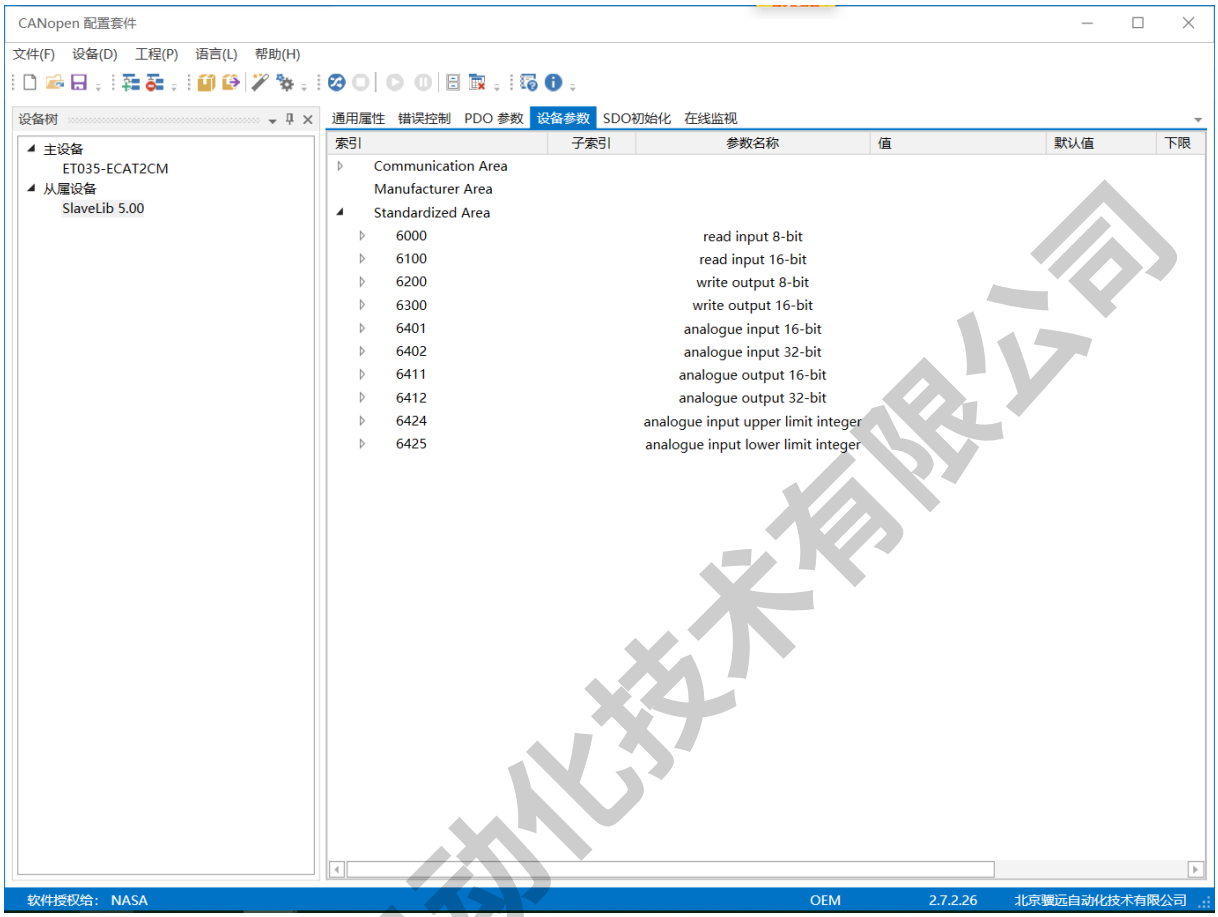
无需配置的 RPDO 或 TPDO 选项去掉前端的勾选项即可；



右击对应的 RPDO 或 TPDO 添加对象；

选择相应的索引，点击添加，添加完成后关闭对话框；

5.2.5设备参数



5.2.6SDO初始化

将用户组态的 PDO 参数进行 SDO 初始化各个索引参数；

通用属性 | 错误控制 | PDO 参数 | 设备参数 | SDO初始化 | 在线监视

设备名称	站地址	索引	子索引	原始值	重写值	大小(bit)	使能
SlaveLib 5.00	0x1						
		1017	0	0x1F4	0x	16	☑
		1014	0	0x80000081	0x	32	☑
		1014	0	0x81	0x	32	☑
		1016	80	0x102EE	0x	32	☑
		1400	1	0x80000201	0x	32	☑
		1600	0	0x0	0x	8	☑
		1600	1	0x62000108	0x	32	☑
		1600	0	0x1	0x	8	☑
		1400	2	0x1	0x	8	☑
		1400	3	0x0	0x	16	☑
		1400	1	0x201	0x	32	☑
		1800	1	0x80000181	0x	32	☑
		1A00	0	0x0	0x	8	☑
		1A00	1	0x60000108	0x	32	☑
		1A00	0	0x1	0x	8	☑
		1800	2	0x1	0x	8	☑
		1800	3	0x0	0x	16	☑
		1800	1	0x181	0x	32	☑

5.2.7错误控制

通用属性 | 错误控制 | PDO 参数 | 设备参数 | SDO初始化 | 在线监视

节点Id	逻辑名称	生产者时间	消费者时间	守护时间	寿命因子
0x7F	ET005-PN2CM	0	...	0	0
1	SlaveLib 5.00	500	0	0	0

心跳（生产者时间，消费者时间）

节点保护（守护时间，寿命因子）

心跳和节点保护为互斥使用，

心跳一般从站作为生产者，主站作为消费者，主站监视从站

节点保护：主站请求从站状态，从站上报状态给主站，用于相互监视。

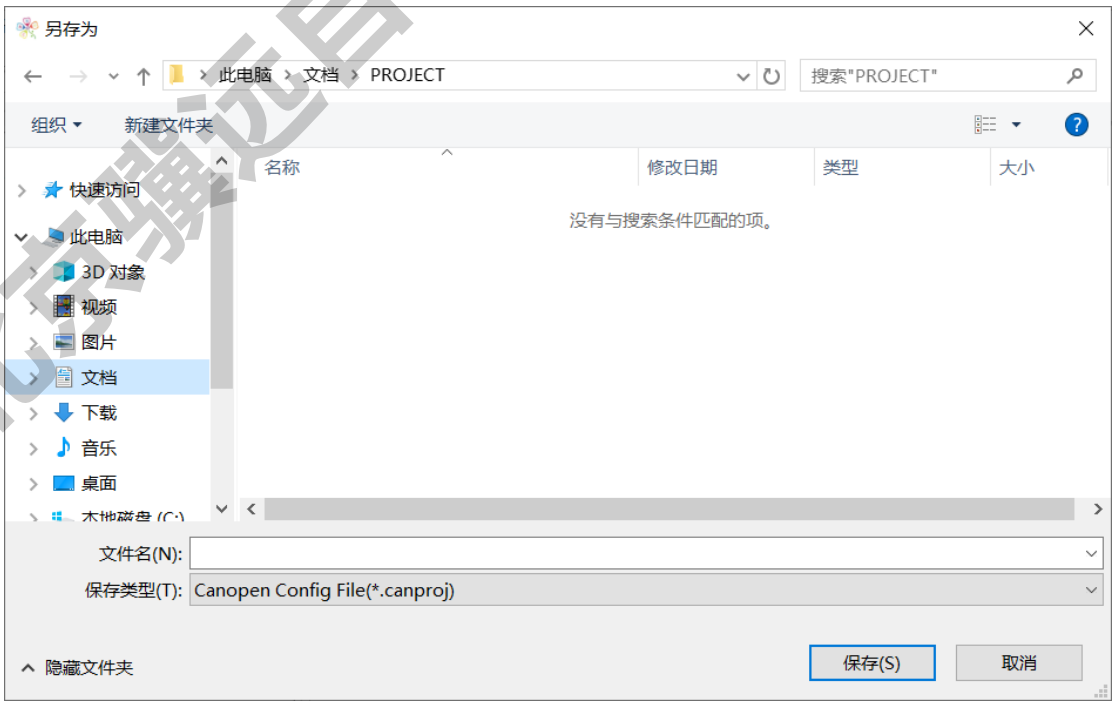
5.3 数据映射

通过工具栏的“ ”自动偏移按钮或者菜单栏“工程（P）”下 “自动偏移”选项完成通道偏移，也可自定义偏移；

5.4 加载和保存配置

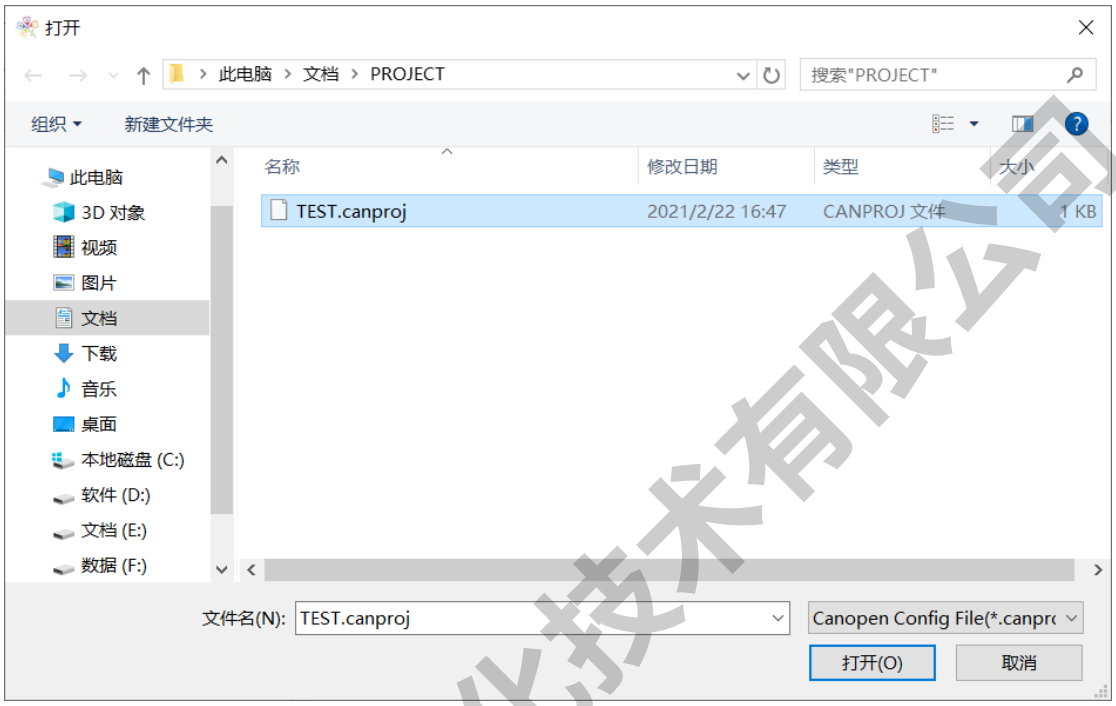
5.4.1保存配置工程

在“文件”中选择“保存”，可以将配置好的工程以.canproj 文件保存，如下图所示：



5.4.2加载配置工程

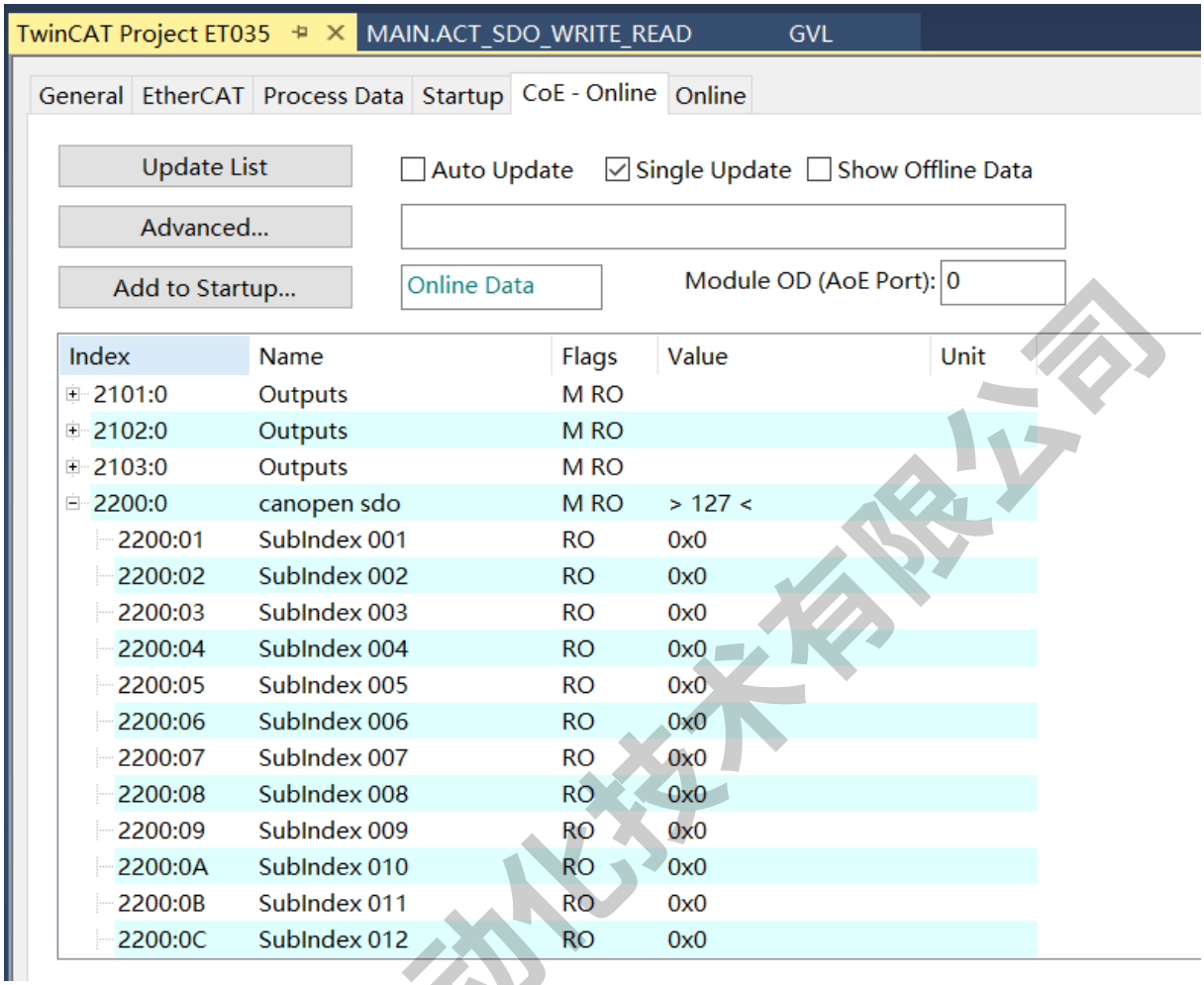
在“文件”中选择“打开”，可以将保存的.canproj 文件打开。



5.5 CANopen 之 SDO 读写说明

通过 CoeSDO 通讯的功能块实现 SDO 的读写操作，功能块属于库文件 TcEtherCAT.Lib；

ECAT-CANopen 网关模块的 SDO 索引地址为 0x2200，子索引地址代表从站地址，如下图所示：



在上图中的“Value”为 64 bits 参数；下表介绍“Value”参数各个字节的含义：

SDO 发送

参数	功能	描述
字节 0	Bit0-bit1 功能（此时功能码为 1） Bit2-bit4 操作数据长度	Bit0-bit1 功能码 1：写 SDO :2#0000 0001 2：读 SDO :2#0000 0010 3：读状态 :2#0000 0011 Bit2-bit4 操作数据长度 4bytes： 2#100:data0 , data1, data2, data3 3bytes： 2#011:data0, data1, data2 2bytes： 2#010:data0 , data1 1byte 2#001:data0 综上所述，例如操作写 SDO 不同数据长度，字节 0 的参数值如下： 4bytes ： 0x11

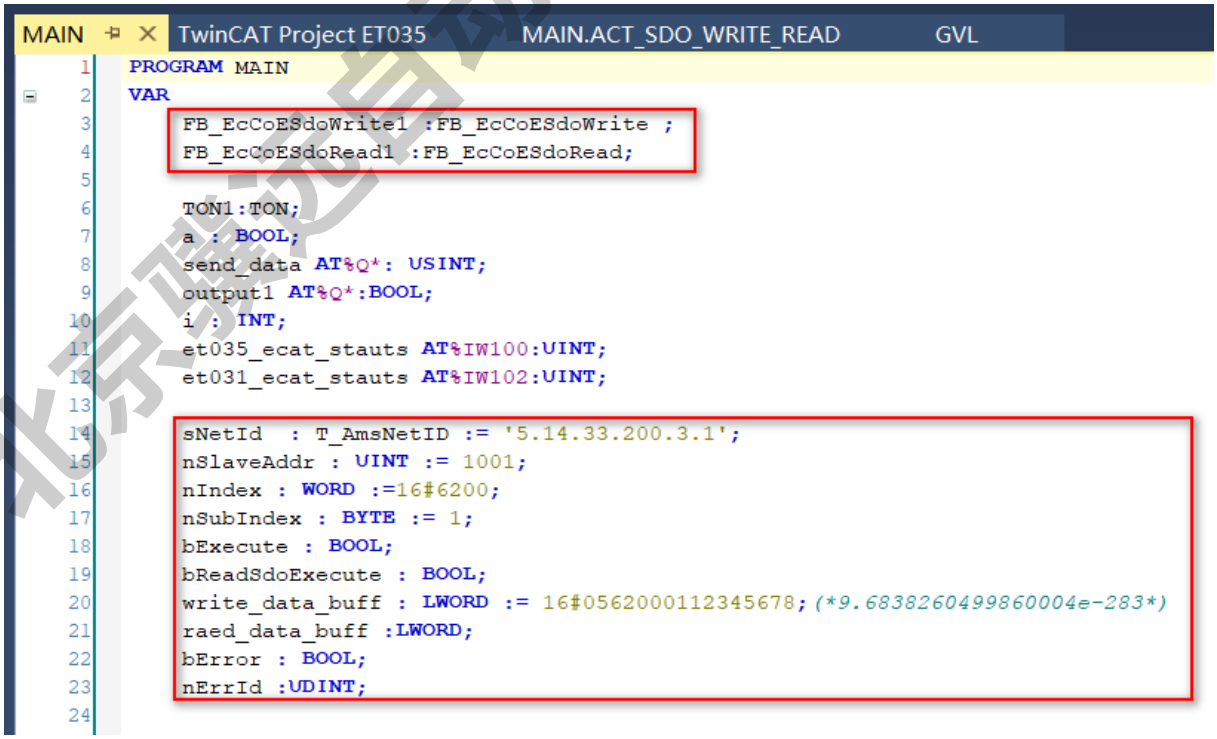
		3bytes ： 0x0D 2bytes ： 0x09 1bytes ： 0x05
字节 1	索引高字节	Index High byte
字节 2	索引低字节	Index Low byte
字节 3	子索引	SubIndex
字节 4-7	数据	data0 , data1, data2, data3 （与字节 0 相关）

SDO 应答：

参数	功能	描述
字节 0	0x40+功能码（此时功能码为 1） Bit2-bit4 操作数据长度 Bit6：代表完成 Bit7：最高位代表有错误码	例如： 0x51：写 4 字节 SDO 回复 0x4A：读 2 字节 SDO 回复 0x43：读从站状态回复 0xCX：代表有错误码，数据部分
字节 1	索引高字节	Index High byte
字节 2	索引低字节	Index Low byte
字节 3	子索引	SubIndex
字节 4-7	数据	data0 , data1, data2, data3 （与字节 0 相关）

第一步：创建 SDO 读写功能，并配置好参数；如下图调用 FB_EcCoeSdoWrite 和 FB_EcCoeSdoRead

功能块；



第二步：写 SDO，直接调用 FB_EcCoeSdoWrite 功能块；

在 FB_EcCoeSdoWrite 块中对 1 号从站索引 16#6200，子索引 16#01 写 1 字节值为 16#12；

nSubIndex : 16#01 （对应 1 号从站）

nIndex : 16#2200 （对应 CANopen SDO 操作索引）

write_data_buff：16#0562000112000000 （参照 SDO 发送表格说明）

MAIN [Online] TwinCAT Project ET035 MAIN.ACT_SDO_W...E_READ [Online] GVL [Online]

TwinCAT_Project_ET035.ET035Demo.MAIN.ACT_SDO_WRITE_READ

```
FB_EcCoESdoWrite1(
    sNetId:= '5.14.33.200.3.1',
    nSlaveAddr:= nSlaveAddr,
    nSubIndex:= nSubIndex,
    nIndex:= nIndex,
    pSrcBuf:= ADR(write_data_buff),
    cbBufLen:= SIZEOF(write_data_buff),
    bExecute:= bExecute,
    tTimeout:= ,
    bBusy=> ,
    bError=> bError,
    nErrId=> nErrId);

FB_EcCoESdoRead1(
    sNetId:= sNetId,
    nSlaveAddr:= nSlaveAddr,
    nSubIndex:= nSubIndex,
    nIndex:= nIndex,
    pDstBuf:= ADR(raed_data_buff),
    cbBufLen:= SIZEOF(raed_data_buff),
    bExecute:= bReadSdoExecute,
    tTimeout:= ,
    bBusy=> ,
    bError=> ,
    nErrId=> );RETURN
```

bExecute = TRUE 后，CANopen 主站侧发送的报文如下：

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	接收	16:04:13.491			0x00000601	数据帧	标准帧	0x08	2F 00 62 01 12 00 00 00
1	接收	16:04:13.493			0x00000581	数据帧	标准帧	0x08	60 00 62 01 00 00 00 00

第三步：读 SDO，先调用 FB_EcCoeSdoWrite 功能块通知主站要读取从站设备的对象字典，再调用 FB_EcCoeSdoRead 功能块读取该对象字典；

MAIN [Online] TwinCAT Project ET035 MAIN.ACT_SDO_W...E_READ [Online] GVL [Online]

TwinCAT_Project_ET035.ET035Demo.MAIN.ACT_SDO_WRITE_READ

```
FB_EcCoESdoWrite1(
    sNetId '5.14.33.20' := '5.14.33.200.3.1',
    nSlaveAddr 16#03E9 := nSlaveAddr 16#03E9,
    nSubIndex 16#01 := nSubIndex 16#01,
    nIndex 16#2200 := nIndex 16#2200,
    pSrcBuf 16#81B5C5B8 := ADR(write_data_buff 16#0662000100000000),
    cbBufLen 16#00000008 := SIZEOF(write_data_buff 16#0662000100000000),
    bExecute FALSE := bExecute FALSE,
    tTimeout := ,
    bBusy => ,
    bError FALSE => bError FALSE,
    nErrId 16#00000000 => nErrId 16#00000000);

FB_EcCoESdoRead1(
    sNetId '5.14.33.20' := sNetId '5.14.33.20',
    nSlaveAddr 16#03E9 := nSlaveAddr 16#03E9,
    nSubIndex 16#01 := nSubIndex 16#01,
    nIndex 16#2200 := nIndex 16#2200,
    pDstBuf 16#81B5C5C0 := ADR(raed_data_buff 16#0662000189000000),
    cbBufLen 16#00000008 := SIZEOF(raed_data_buff 16#0662000189000000),
    bExecute TRUE := bReadSdoExecute TRUE,
    tTimeout := ,
    bBusy => ,
    bError => ,
    nErrId => );RETURN
```

FB_EcCoeSdoWrite 功能块执行“write_data_buff” = 16#0662000100000000，即通知 CANopen 主站下发从站即将读取索引 16#6200，子索引 16#01 参数值；如上图 FB_EcCoeSdoWrite1 功能块；

执行该功能块后，CANopen 主站侧报文如下图序号 2 和 3 数据：

序号	传输方向	时间标识	状态	名称	帧ID	格式	类型	DLC	数据
0	接收	16:40:40.744			0x00000601	数据帧	标准帧	0x08	2F 00 62 01 89 00 00 00
1	接收	16:40:40.748			0x00000581	数据帧	标准帧	0x08	60 00 62 01 00 00 00 00
2	接收	16:42:01.370			0x00000601	数据帧	标准帧	0x08	40 00 62 01 00 00 00 00
3	接收	16:42:01.372			0x00000581	数据帧	标准帧	0x08	4F 00 62 01 89 00 00 00

接着执行执行 FB_EcCoeSdoRead 功能块操作“read_data_buff” = 16#0662000100000000，bExecute = TRUE 后，获取的参数值直接显示在 read_data_buff 中；

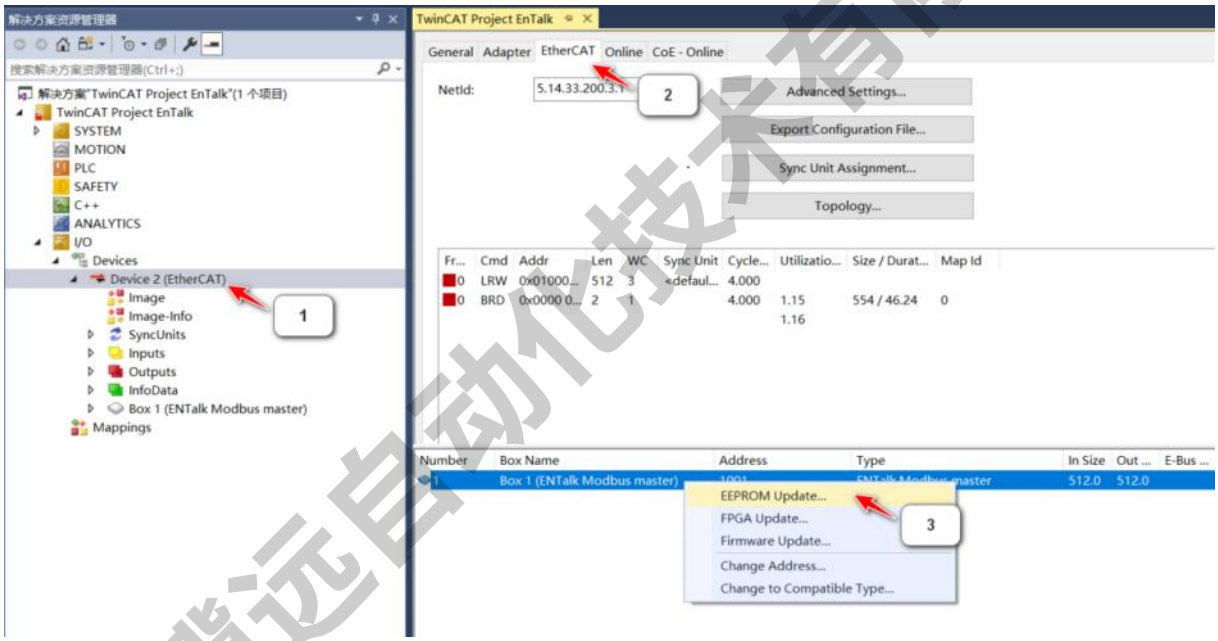
6 烧写 EEPROM

EEPROM 存放从站配置信息，即为 ESI 文件(.XML 文件)。

烧录方法可以有很多种，一般都是通过主站来烧写。下面介绍 TwinCAT3 直接烧写 XML 文件

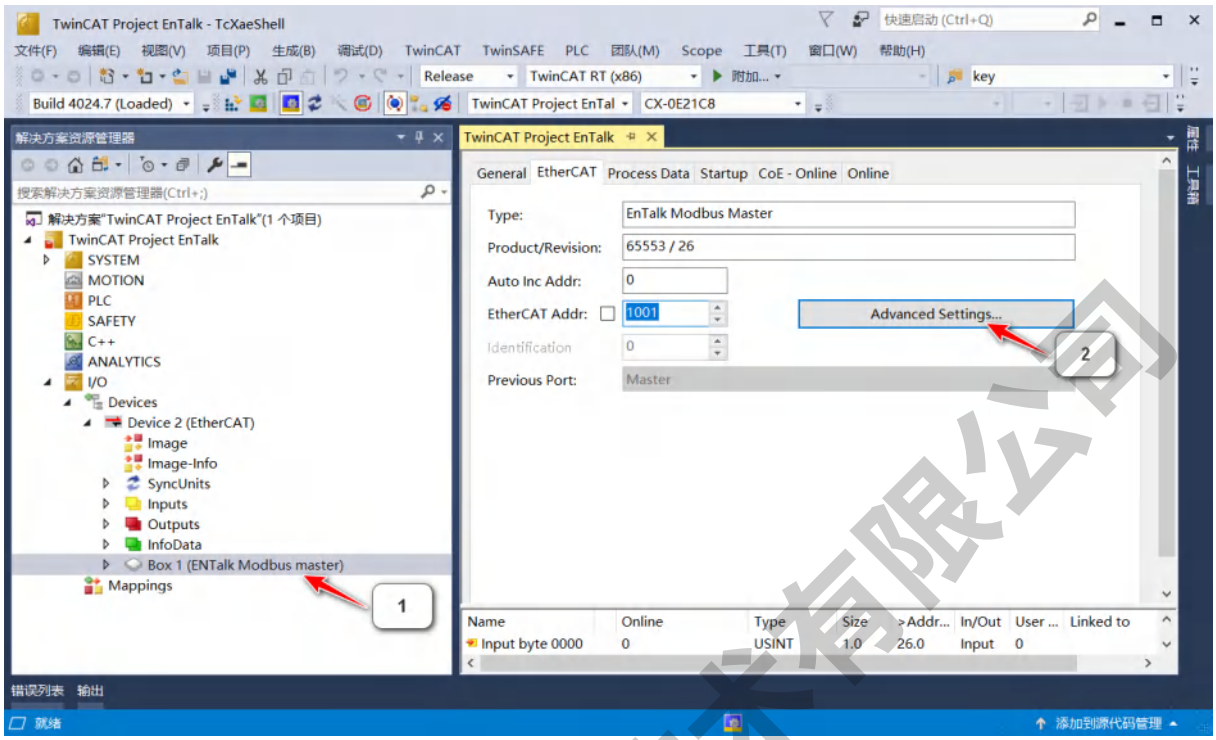
第一种方式：

1. 选择 EtherCAT 总线，
2. 在右侧窗口中选择“EtherCAT”选项，从站设备将显示；
3. 选择 ECAT-CANopen 从站，右击选择“EEPROM Update...”,弹出“Write EEPROM”对话框，选择烧写的 XML 文件：EnTalk_ECAT2ModbusMaster.XML；点击“OK”进行烧写，在窗口的右下角有状态进度条可以观察烧写进度；

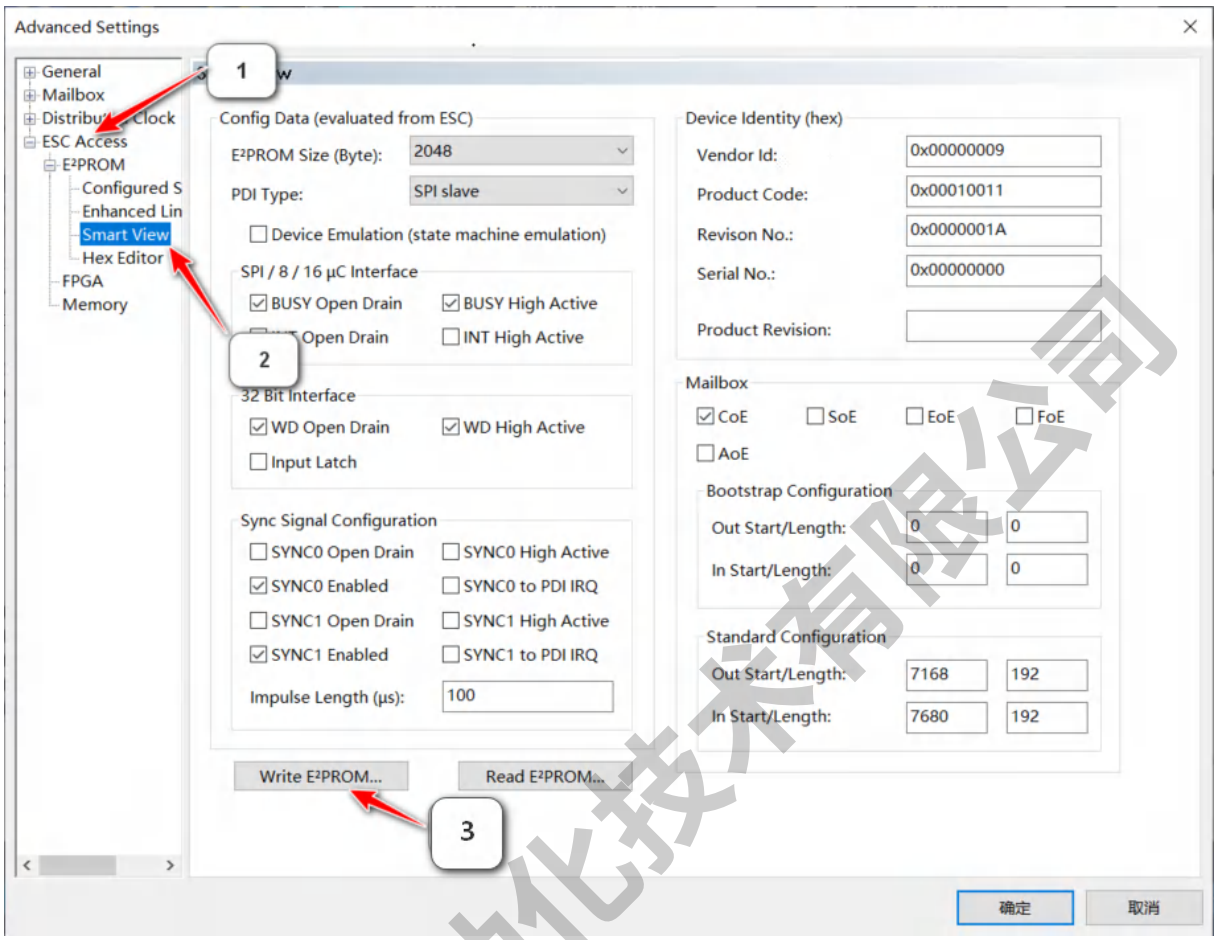


第二种方式：

1. 选择从站设备，
2. 在右侧窗口中选择“EtherCAT”选项，点击“Advanced Settings...”；



3. 弹出“Advanced Settings”对话框， 选择“SEC Access”- “EEPROM” - “Smart View”，点击“Write EEPROM”，选择烧写的 XML 文件：EnTalk_ECAT2CANopenMaster.XML；点击“OK”进行烧写，在窗口的右下角有状态进度条可以观察烧写进度；



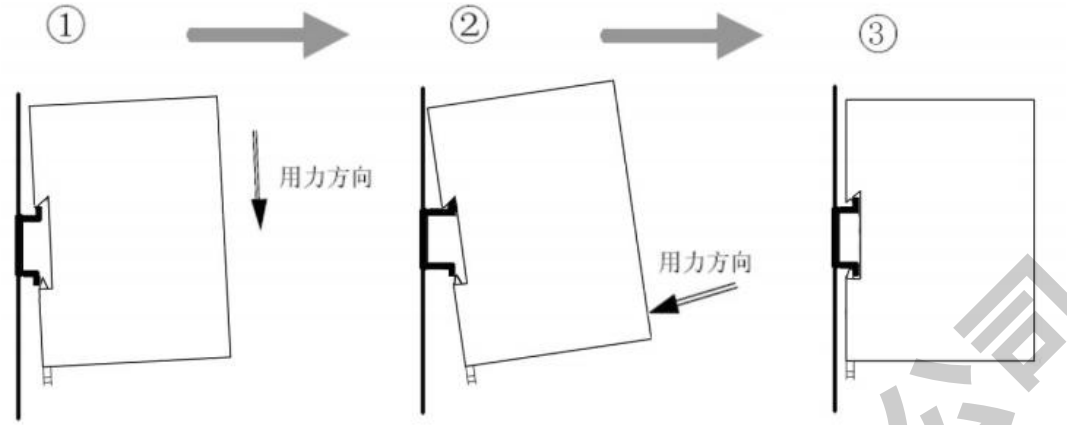
7 安装

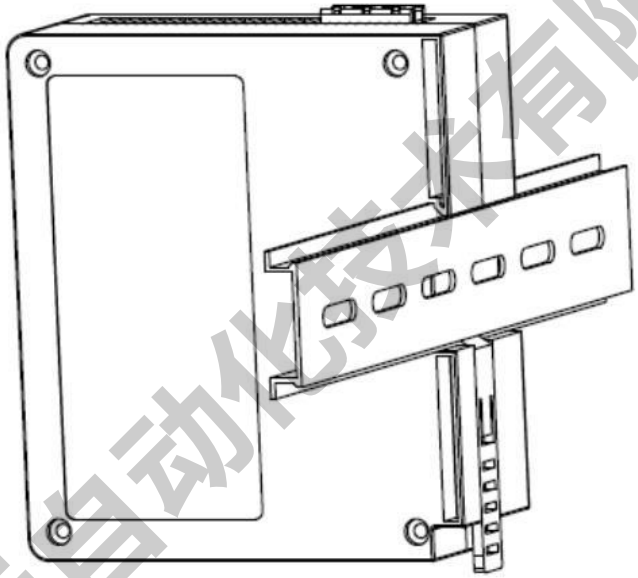
7.1 机械尺寸

尺寸：40mm（宽）×110mm（高）×74mm（深）

7.2 安装方法

35mm DIN 导轨安装





8 运行维护及注意事项

- 模块需防止重压，防止损坏；
- 模块需防止重击，以防器件损坏；
- 供电电压控制在说明书的要求范围内，防止内部器件烧坏；
- 模块防止进水，防止内部器件损坏；
- 上电前请检查接线，防止接错损坏模块。