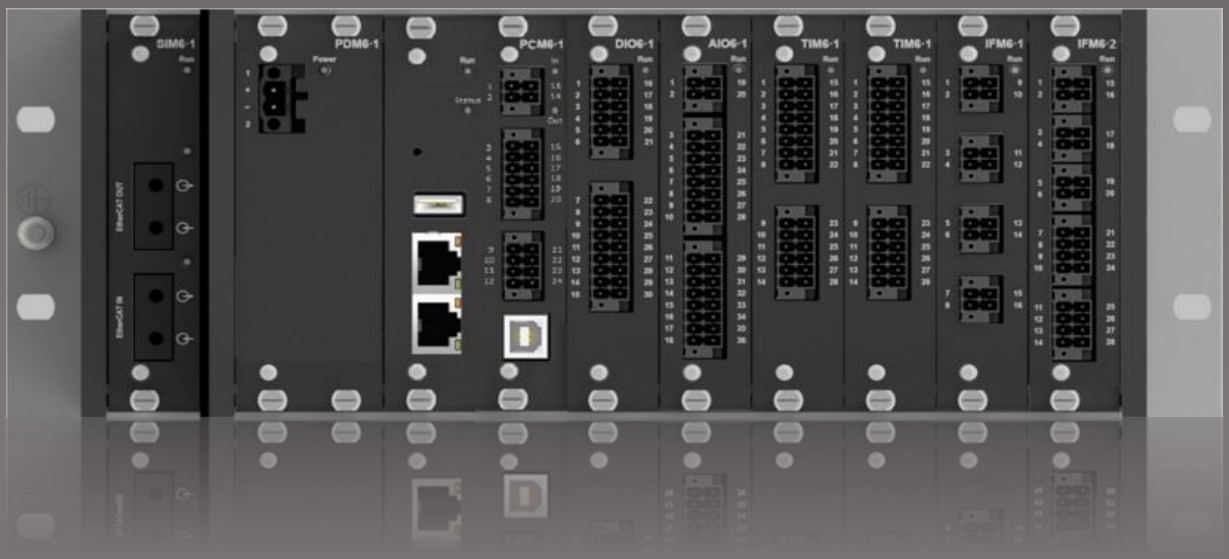
福氏技术
PRACTEK



控制平台 AWP100 基础编程手册

文件编号：4155000081
文件版本：1.0.0.0
发布日期：2022-12-30

福氏新能源技术（上海）有限公司
www.practek.cn

目录

1 版本

版本	作者	发布日期	描述
1.0.0.0	CHS	2022-12-30	首次发布；

2 前言

2.1 关于 AWP100

AWP100 是为满足严苛应用环境而设计开发的高级控制系统平台，模块化控制器和 I/O 模块具有高度的可靠性、稳健性和灵活性。AWP100 产品系列模块之间采用机架背板总线通信，机架之间采用扩展模块进行分布式连接。

2.2 安全提示

本文件所涵盖的所有操作活动中，操作人员应始终遵照相应国家、地区及厂商包括但不仅限于：高低压电器操作规范、安全规程、个人防护、环境保护等与安全和环境相关的法律法规进行规范操作。福氏新能源技术（上海）有限公司谢绝承担由于个人忽视相关法规条例引发人身安全和财产损失的责任。

2.3 免责声明

福氏新能源技术（上海）有限公司保留更改本文件任何内容的权利，恕不另行通知。

2.4 商标

PRACTEK®是福氏新能源技术（上海）有限公司注册商标。

EtherCAT®是 Beckhoff Automation GmbH 注册商标和专利技术。

Linux®是 Linus Torvalds 注册商标。

所有商标和专利技术均归属其各自所有者。

2.5 版权

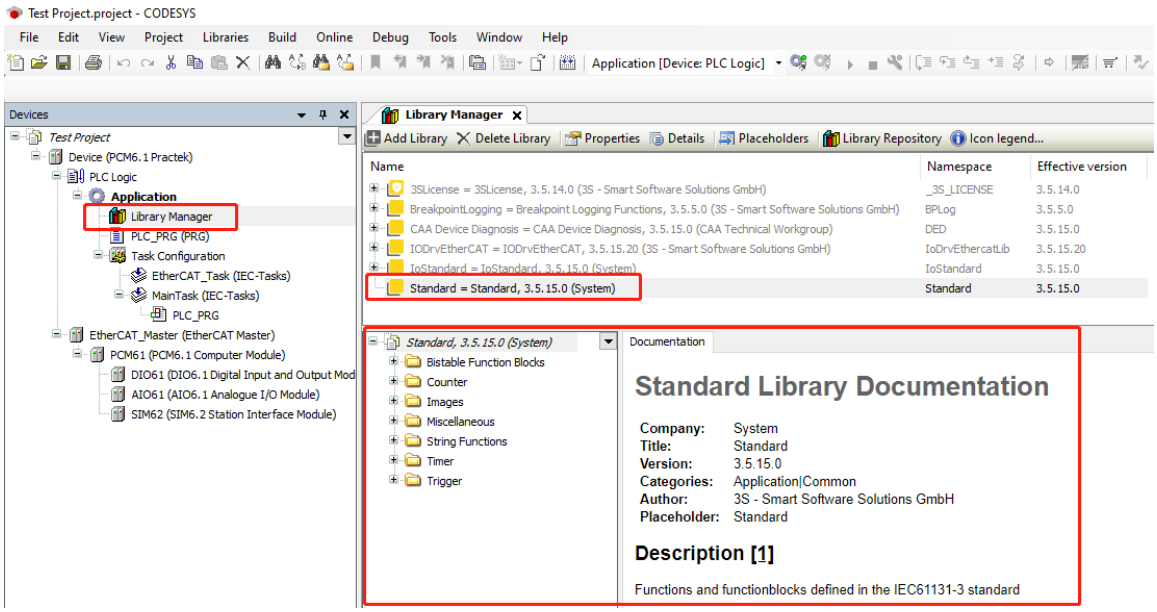
3 AWP100 编程环境

3.1 CODESYS 介绍

AWP100 编程环境由工业自动化领域广泛应用的 CODESYS IDE 提供，福氏技术基于 CODESYS 开发设计 AWP100 的相关功能。CODESYS 是一种功能强大的 PLC 软件编程工具，支持 IEC 61131-3 标准 IL 、ST、 FBD 、LD、 CFC、 SFC 六种 PLC 编程语言，用户可以在同一项目中选择不同的语言编辑子程序、功能模块等。

3.2 CODESYS 库管理

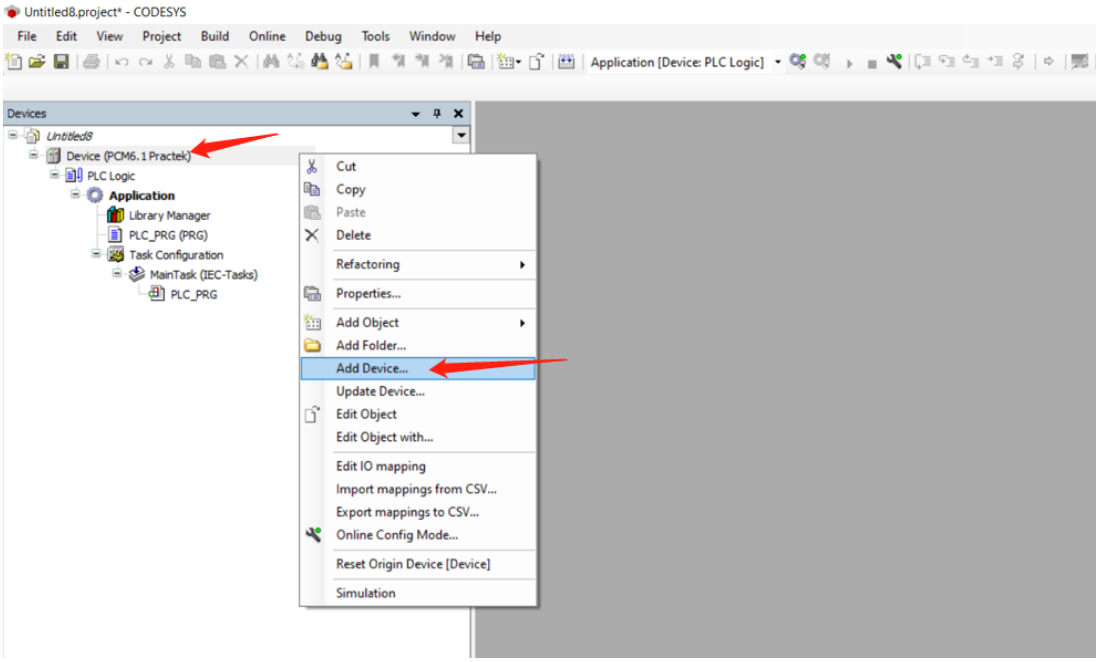
CODESYS 标准库和 AWP100 功能库可以在工程项目中的 Library Manager 库管理器中进行管理，以便在程序中调用。

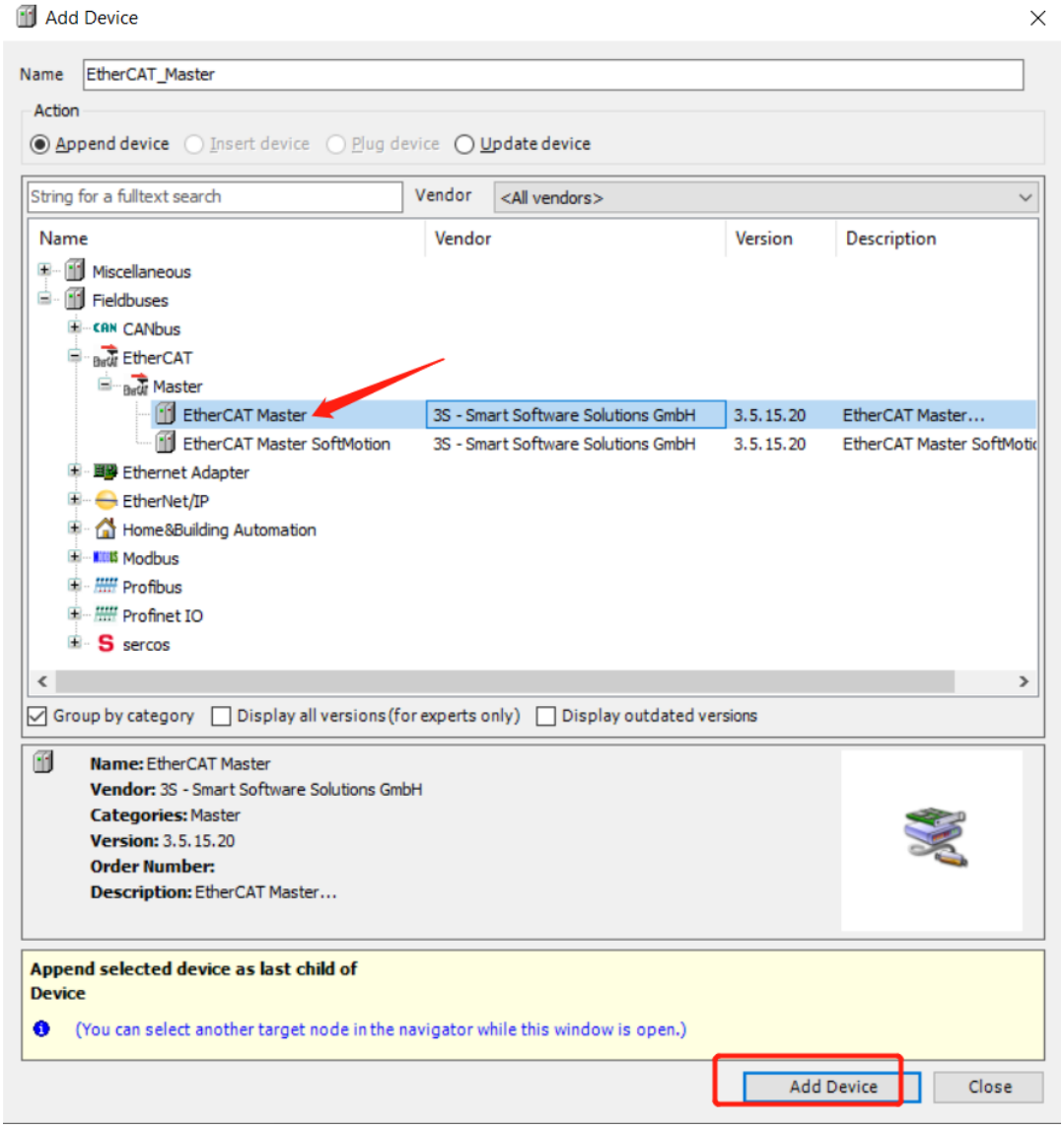


3.3 添加总线

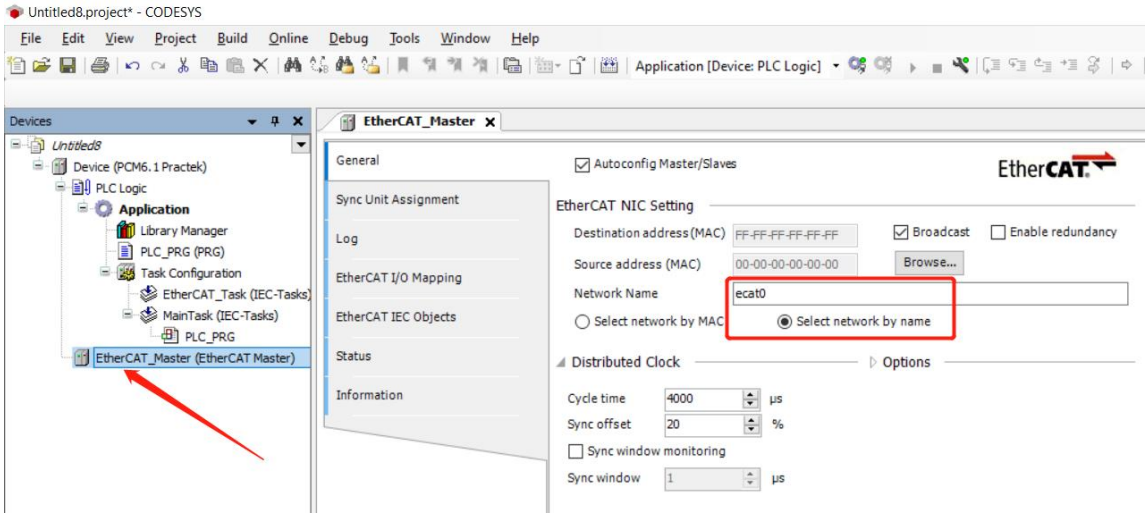
AWP100 产品系列模块之间采用 EtherCAT 总线通信，添加总线 EtherCAT Master 方法如下。

- 右键点击“Device”，选择“Add Device”。
- 弹出对话框选择“Fieldbuses / EtherCAT / Master / EtherCAT Master”。


Untitled8.project* - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Application [Device: PLC Logic]
Devices
Untitled8
Device (PCM6.1 Practek)
PLC Logic
Application
Library Manager
PLC_PRG (PRG)
Task Configuration
MainTask (IEC-Tasks)
PLC_PRG
Cut
Copy
Paste
Delete
Refactoring
Properties...
Add Object
Add Folder...
Add Device...
Update Device...
Edit Object
Edit Object with...
Edit IO mapping
Import mappings from CSV...
Export mappings to CSV...
Online Config Mode...
Reset Origin Device [Device]
Simulation


Add Device
Name EtherCAT_Master
Action
Append device
Insert device
Plug device
Update device
String for a fulltext search
Vendor
<All vendors>
Name
Vendor
Version
Description
Miscellaneous
Fieldbuses
CANbus
EtherCAT
Master
EtherCAT Master
3S - Smart Software Solutions GmbH
3.5.15.20
EtherCAT Master...
EtherCAT Master SoftMotion
3S - Smart Software Solutions GmbH
3.5.15.20
Ethernet Adapter
EtherNet/IP
Home&Building Automation
Modbus
Profibus
Profinet IO
sercos
Group by category
Display all versions (for experts only)
Display outdated versions
Name: EtherCAT Master
Vendor: 3S - Smart Software Solutions GmbH
Categories: Master
Version: 3.5.15.20
Order Number:
Description: EtherCAT Master...
Append selected device as last child of
Device
(You can select another target node in the navigator while this window is open.)
Add Device
Close

➢ 完成 EtherCAT Master 添加，同时“Network Name”处填写“ecat0”且点选“Select network by name”。



3.4 任务配置

Main Task 设置周期、优先级、执行方式、程序调用、看门狗等。

Priority：任务优先级，0~31，0 优先级最高，31 优先级最低。

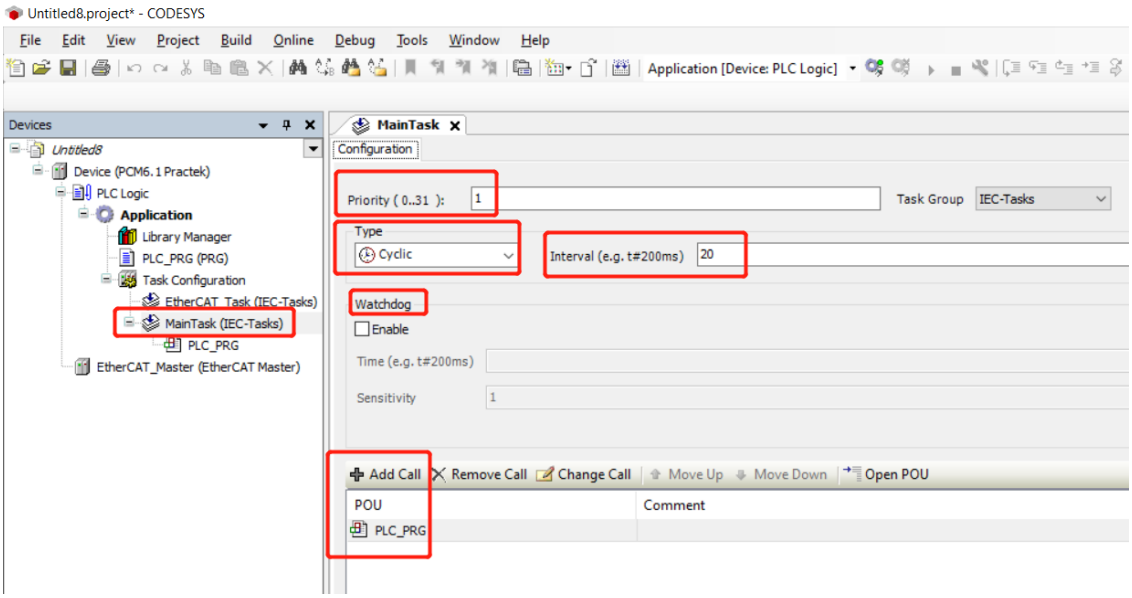
Interval：任务周期。

Type：任务执行方式，通常使用循环 Cyclic。

Watchdog：任务看门狗，监控任务执行情况。

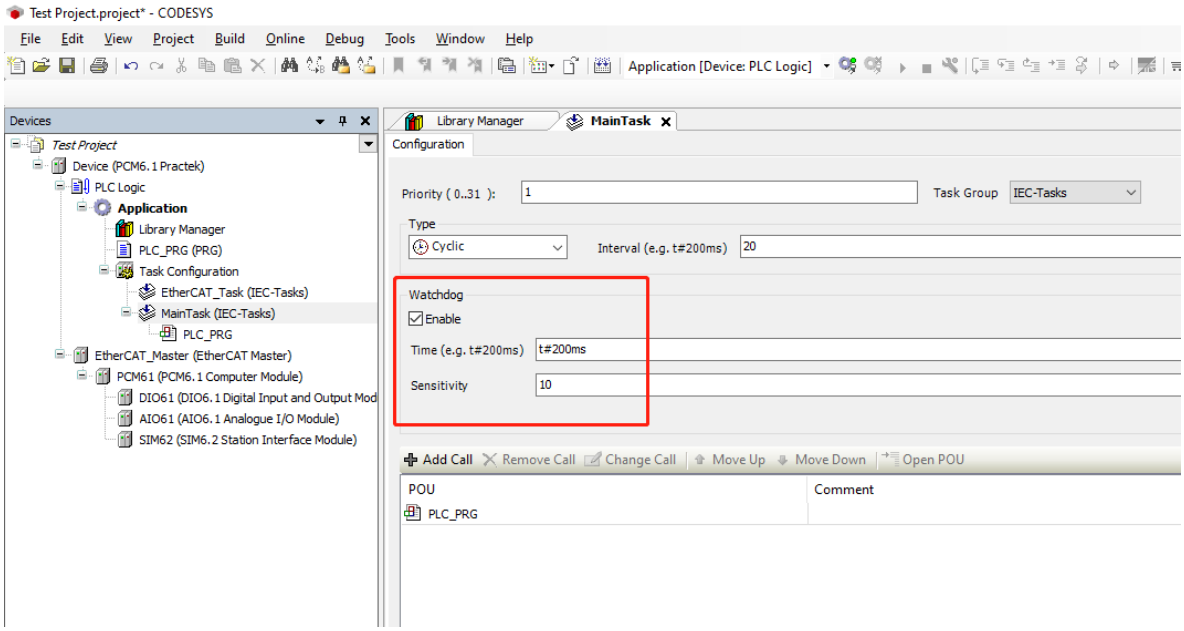
Add Call：任务调用程序，选择任务中执行的程序单元。

EtherCAT_Task 通常采用默认设置。



3.5 任务看门狗

CODESYS 工程项目中任务 Task 运行时间过长或 CPU 超载，可以通过设置任务看门狗功能检测和控制。参考例程 Demo3.5_Watchdog。



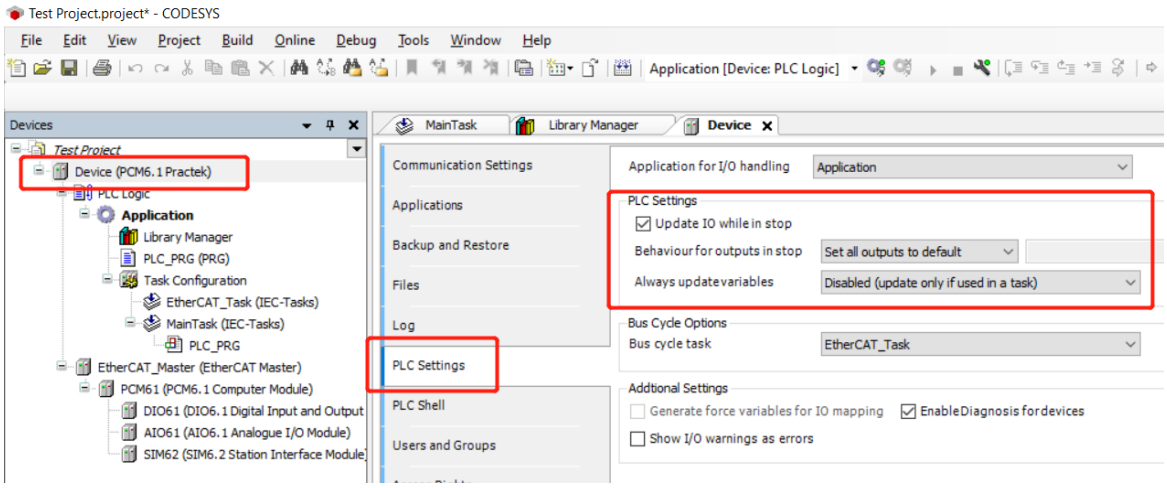
- 任务看门狗设置时间“Time”必须大于任务运行周期。
- 任务运行时间如果超过“Time”×“Sensitivity”，或者任务运行时间连续超过设置时间的次数大于灵敏度 Sensitivity，将导致看门狗触发。

- 看门狗的触发通常是由于代码中含有空指针的调用、死循环、除零等异常情况，当包括看门狗在内的系统故障触发时，实时系统将停止运行并触发 Exception 故障，此时程序内变量将保持故障前状态不变。
- 针对系统异常故障的处理，可以使用 CODESYS 提供的接口回调程序，并在回调程序中设置当出现系统异常故障时采用何种操作处理，例如重启控制器。
- 禁用或重新使能看门狗也可以通过接口函数的方式进行设置。

3.6 设置 I/O 默认状态

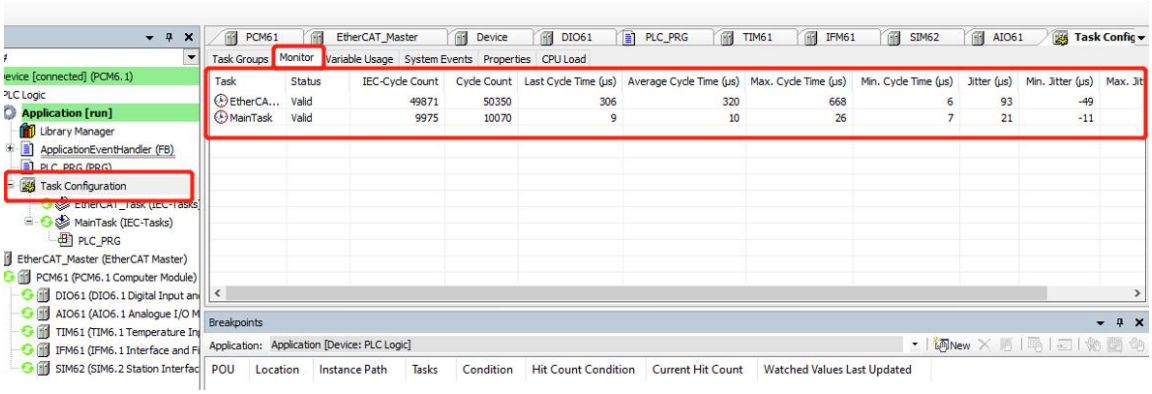
在“Device/PLC Settings”页面可以进行 I/O 默认状态设置：

- 勾选“Update IO while in stop”。
- 设置“Behaviour for outputs in Stop”为“Set all outputs to default”。



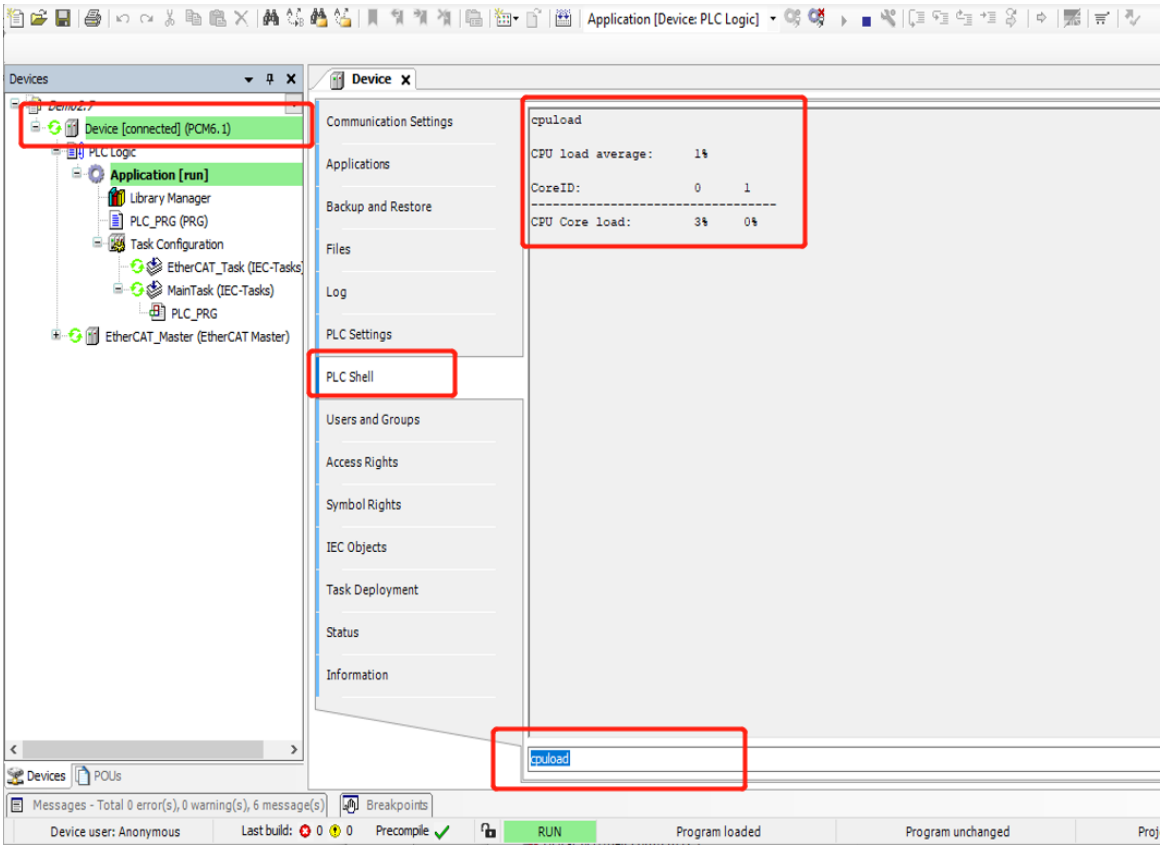
3.7 控制器负载监控

CODESYS 中可以通过 Task Configuration/Monitor 查看任务运行循环时间和统计数据，也可以测量每个子系统的执行时间。任务运行时间必须小于任务设定周期并且尽可能的短，避免因为任务超时导致系统故障。参考例程 Demo3.7_CPUload。

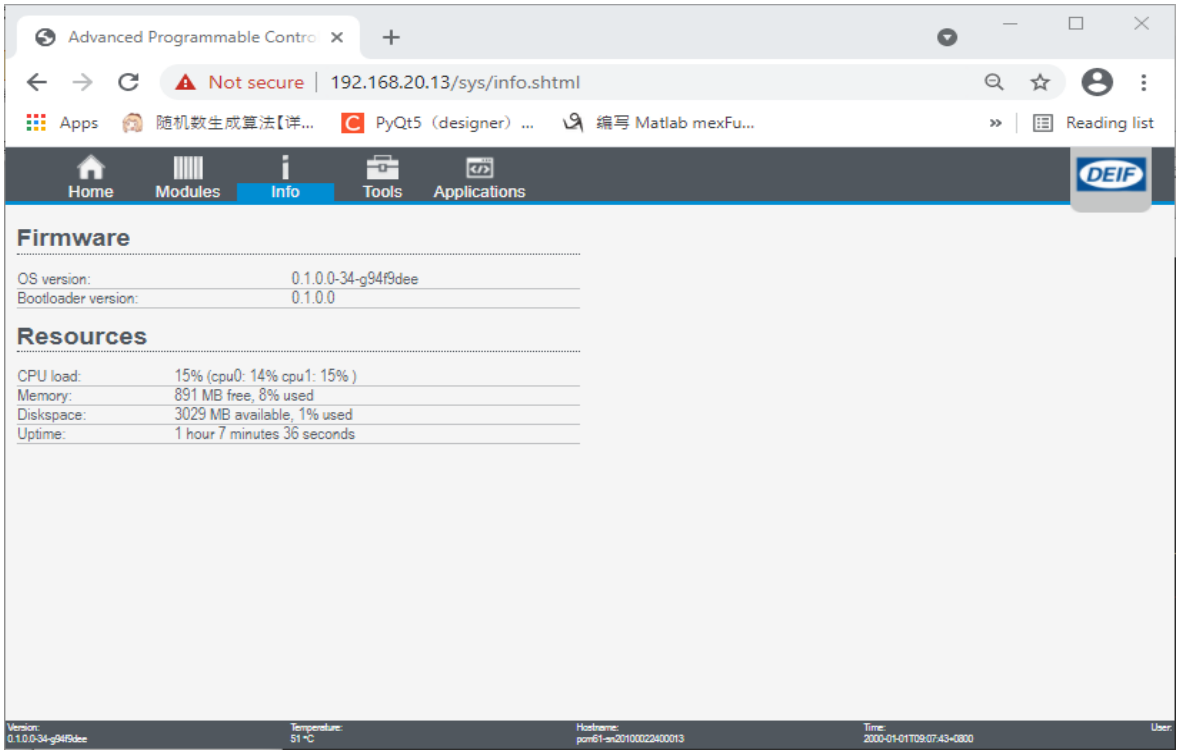


控制器负载情况也可以通过 PLC Shell、网页控制界面查看。

PLC Shell 查看控制器负载：控制器运行条件下，“Device/PLC Shell”页面在线输入指令“cpuload”，即可查看控制器负载情况。



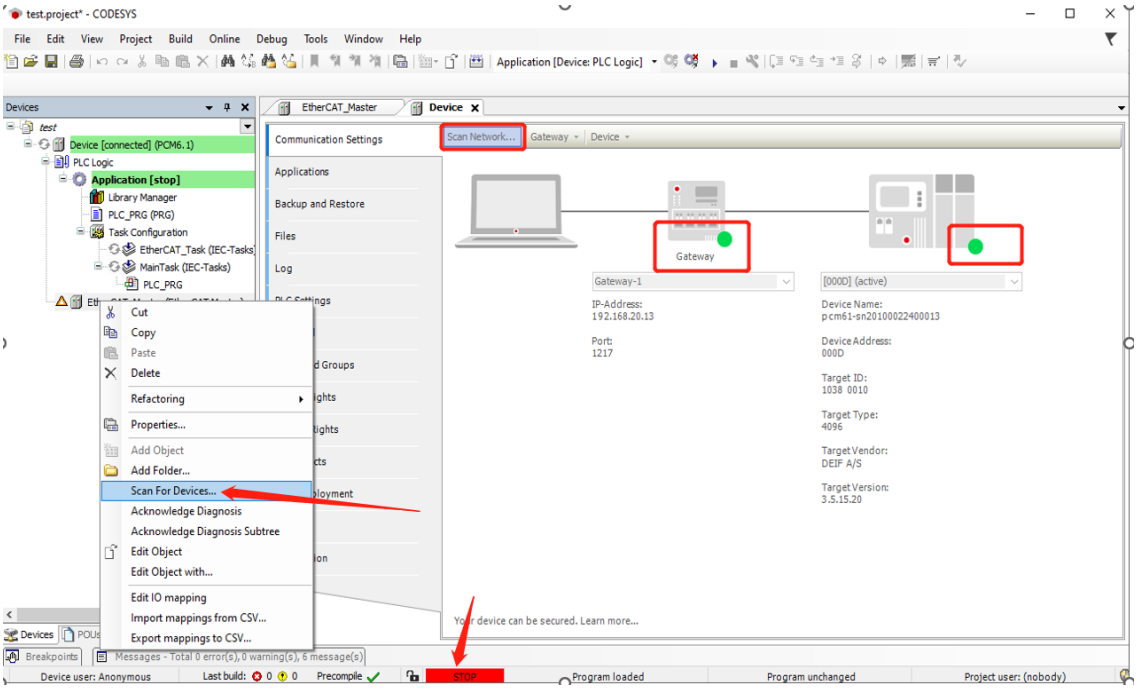
网页控制界面查看控制器负载：浏览器登录控制界面，“Info”页面“CPU load”。

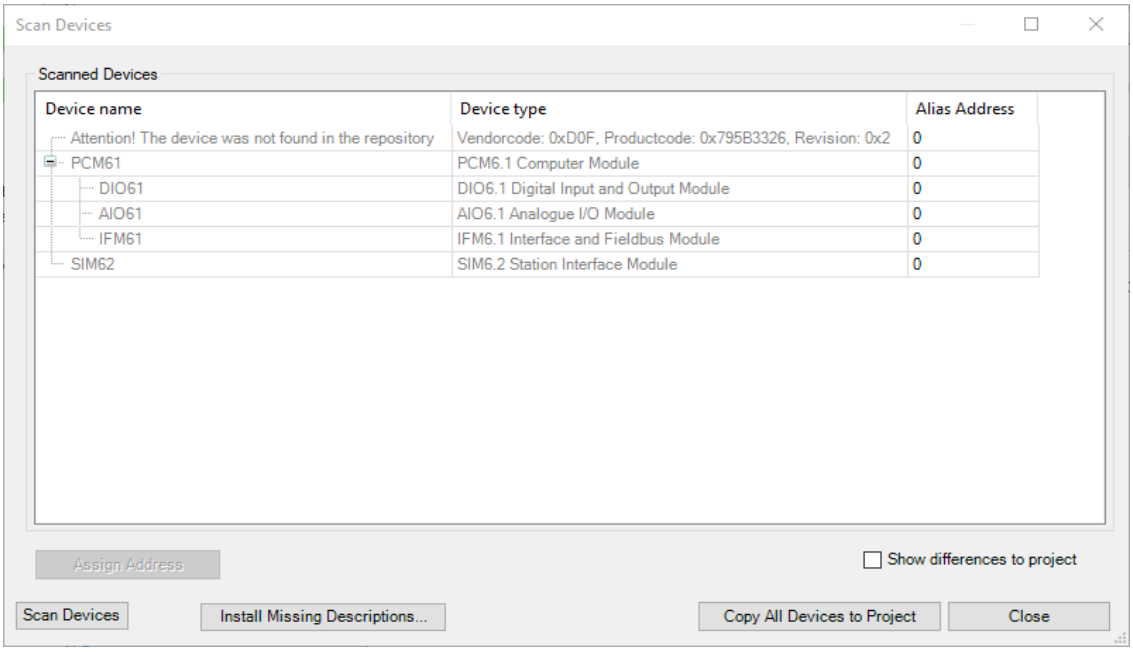


3.8 自动扫描设备

CODESYS 支持 AWP100 自动扫描添加设备功能，方法如下：

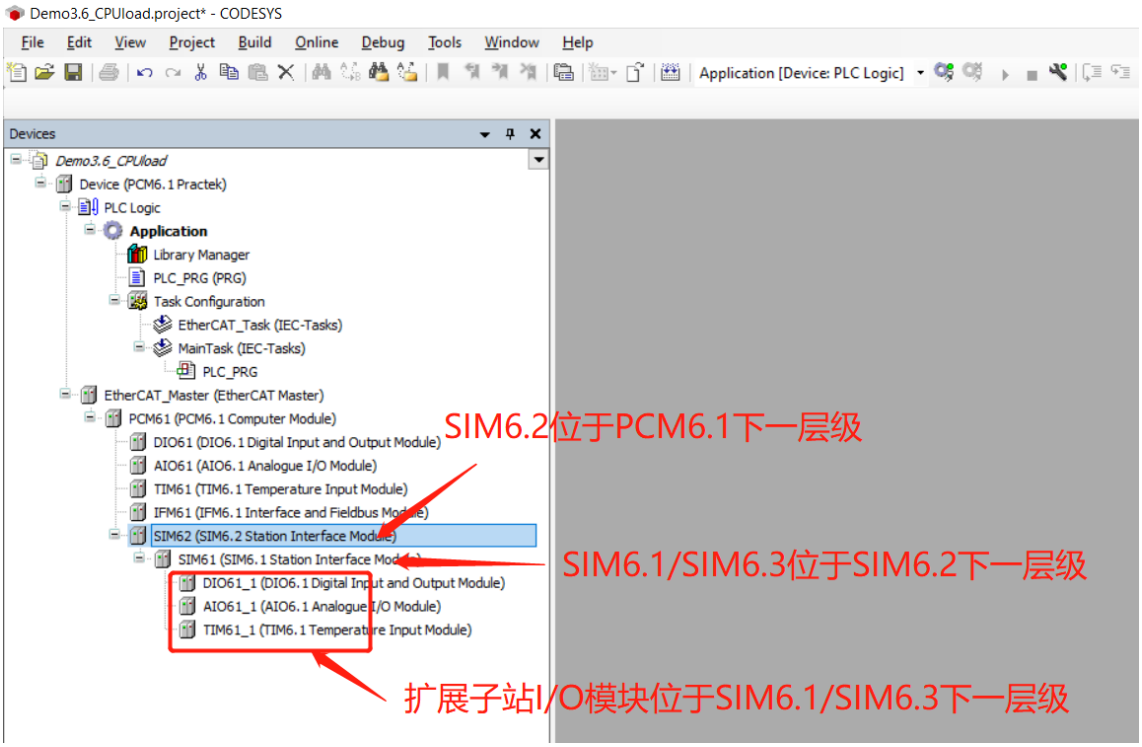
- 新建工程，选择“Standard project”。
- 右键点击“Device”弹出对话框选择“EtherCAT/EtherCAT Master”。
- 双击“EtherCAT_Master”修改配置为“Select network by name”，并将“Network Name”修改为“ecat0”。
- 双击“Device”，点击“Scan Network”扫描在线控制器，建立 Gateway 连接。
- 点击“Login”，但不要运行程序。
- 右键点击“EtherCAT_Master”选择“Scan for Devices”。
- 在线设备自动被扫描出来后，点击“Copy All Devices to Project”将自动扫描的设备导入到工程中。





特别提示：

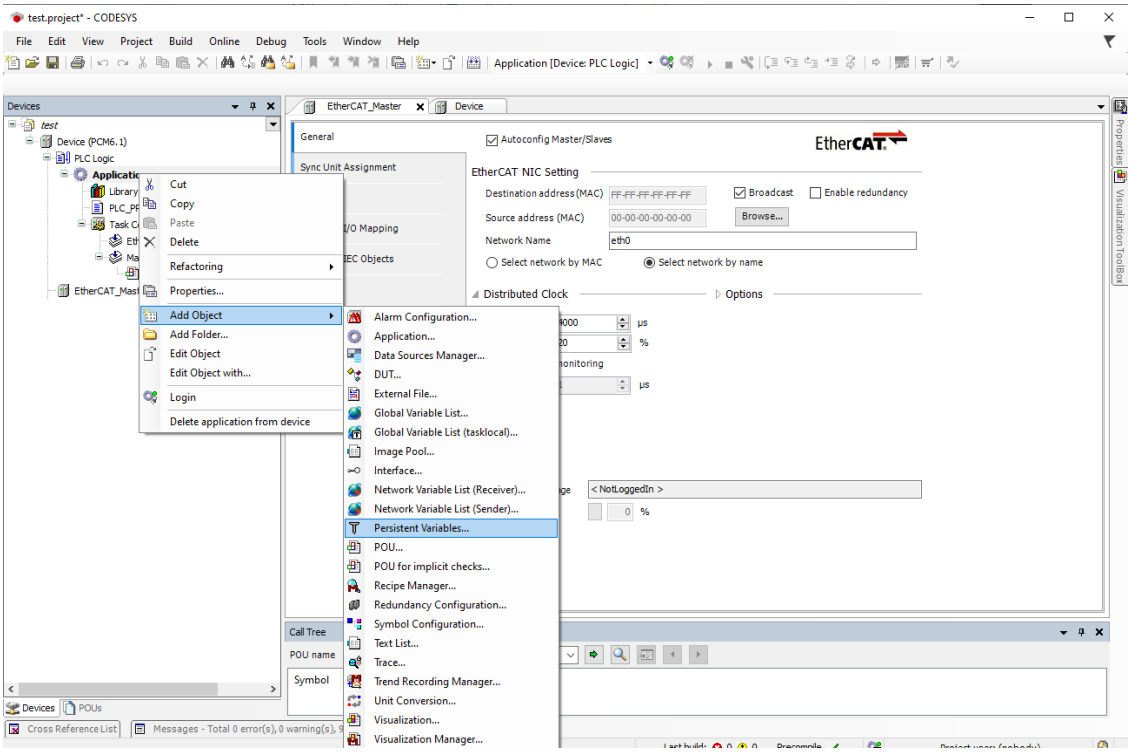
- 不是所有的硬件设备都可以被正确扫描并添加到工程项目中，因此建议核对已扫描添加的设备是否符合配置要求，推荐采用手动方式逐个添加硬件设备。
- SIM6.2 位于 PCM6.1 下一层级。
- 子站扩展模块 SIM6.1/SIM6.3 位于 SIM6.2 下一层级。

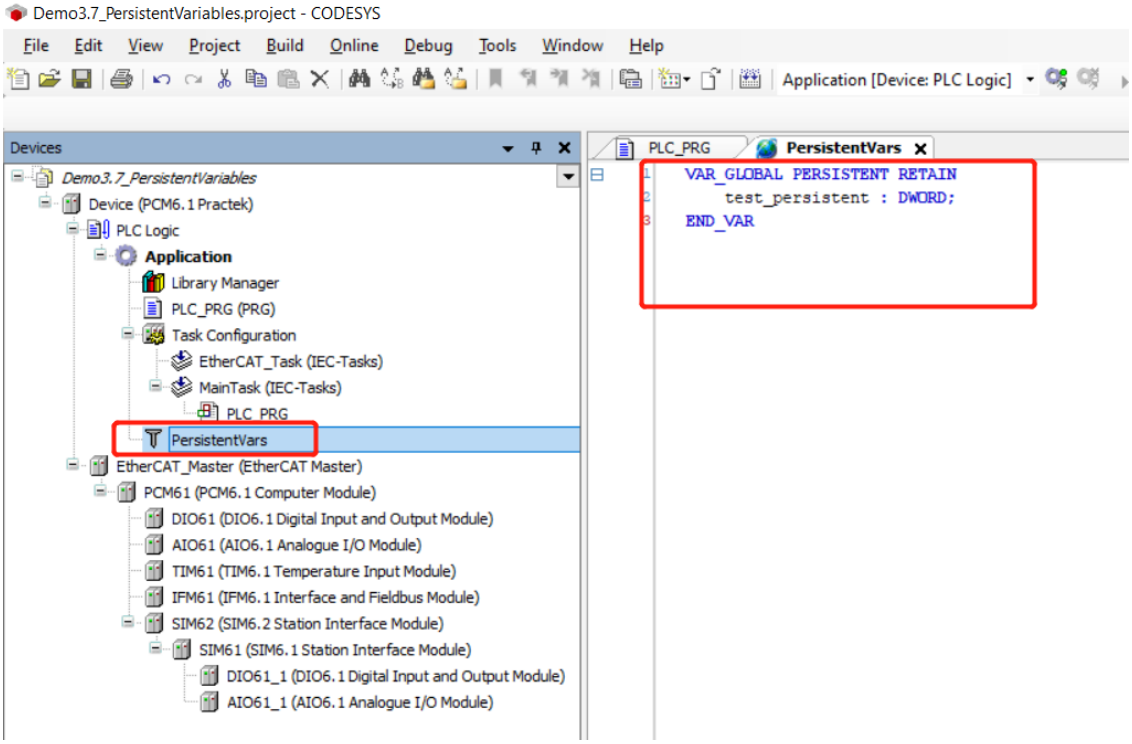


3.9 持久型变量

工程项目中某些数据需要长期存储，避免断电等操作导致数据丢失。可以将其声明为持久型变量，这些变量需要在 Persistent Variables 内声明。虽然持久型变量具有不易丢失的特性，但仍然建议在此基础上将相关数据保存备份，以防备可能的器件损坏、工程误更新等导致数据丢失。

右键点击“Application”，弹出对话框选择“Add Object/Persistent Variables”。参考例程 Demo3.9_PersistentVariables。

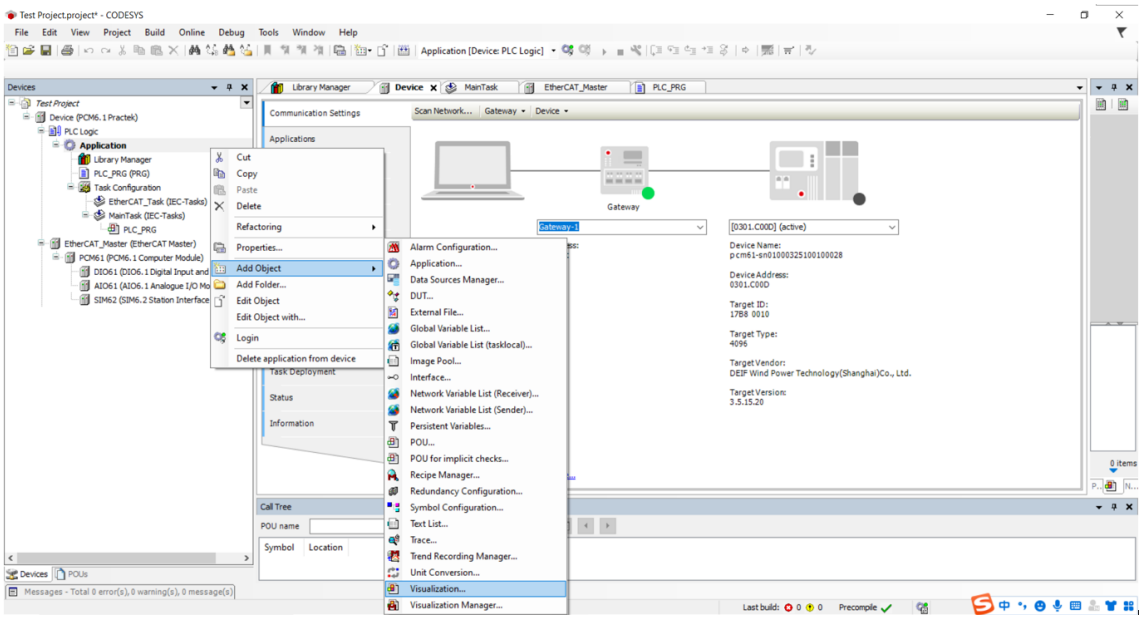




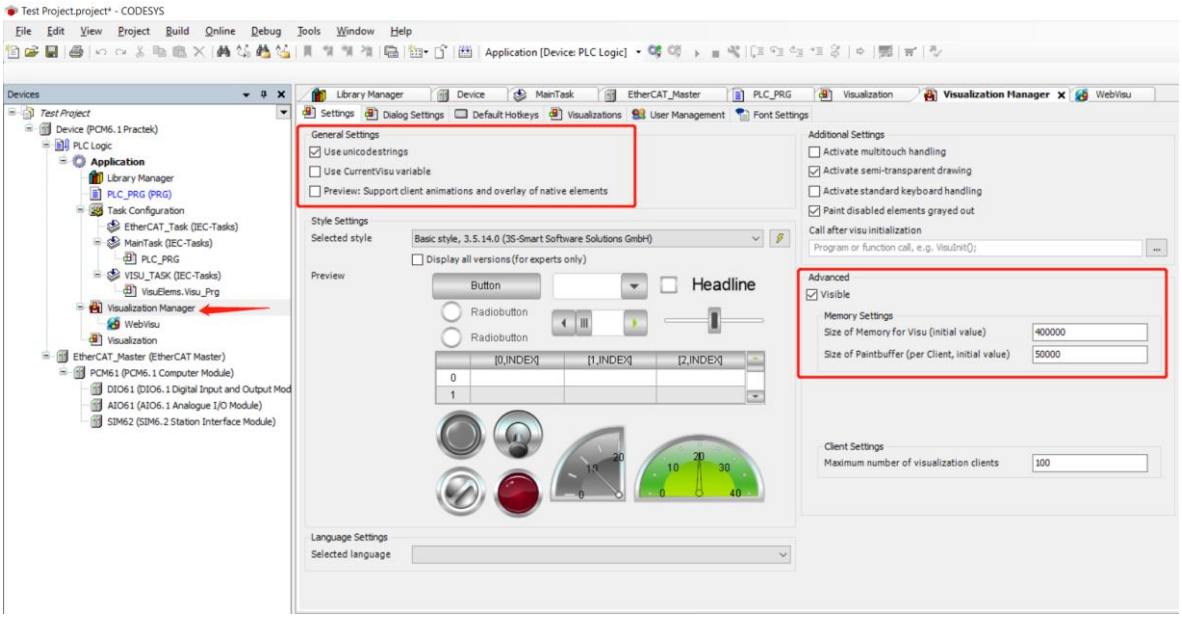
3.10 创建 HMI

CODESYS 支持创建和编辑 HMI 界面，并且可以通过浏览器访问 HMI。

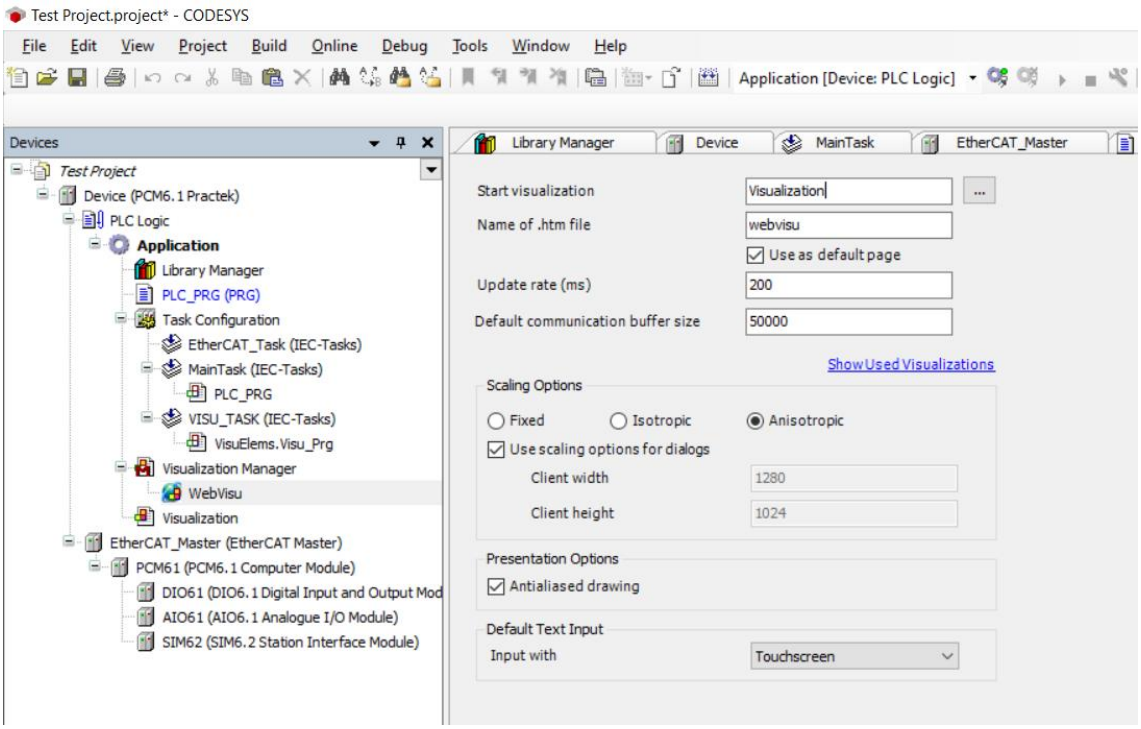
- 新建工程，选择“Standard project”。
- 右键点击“Application”，添加“Add Object/ Visualization”。



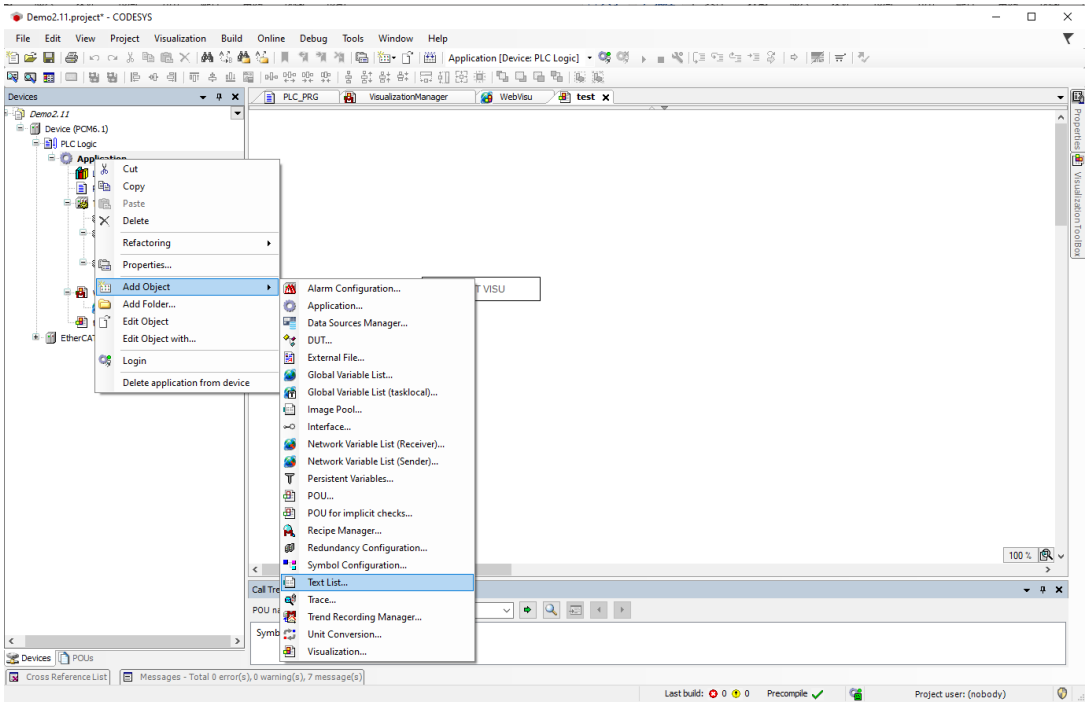
双击“Visualization Manager”，勾选“Visible”设置 HMI 存储区容量。HMI 存储区容量通常采用默认设置，当用户绘制的单一页面元素过多时，若存储区过小可能导致页面显示卡顿或白屏。



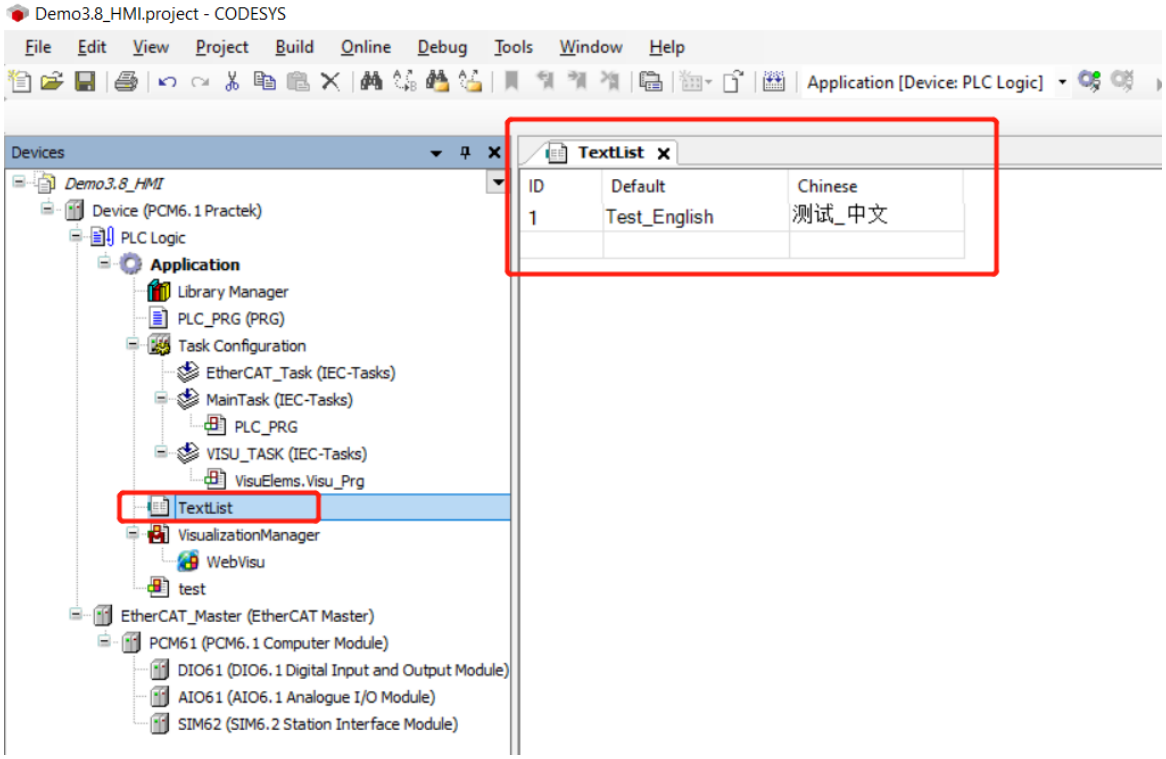
双击“Visualization Manager”下面的“WebVisu”，可以通过修改“Start visualization”设置用户登录到 HMI 界面时显示的首界面，同时在 WebVisu 中还可以对界面的刷新频率、缓存等参数进行设置（通常采用默认设置）。

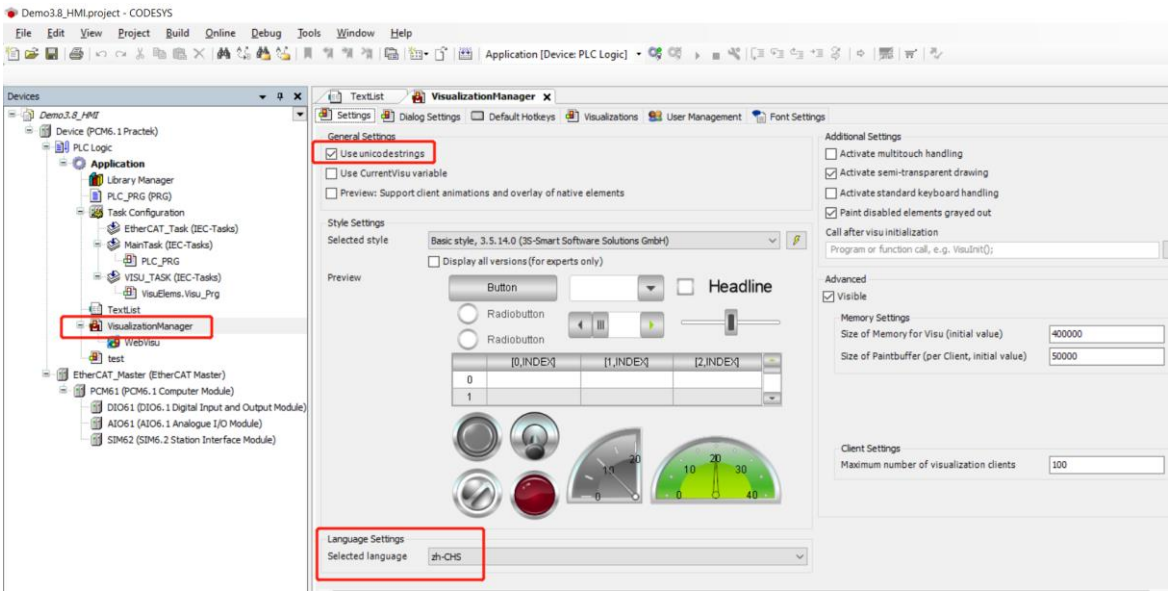


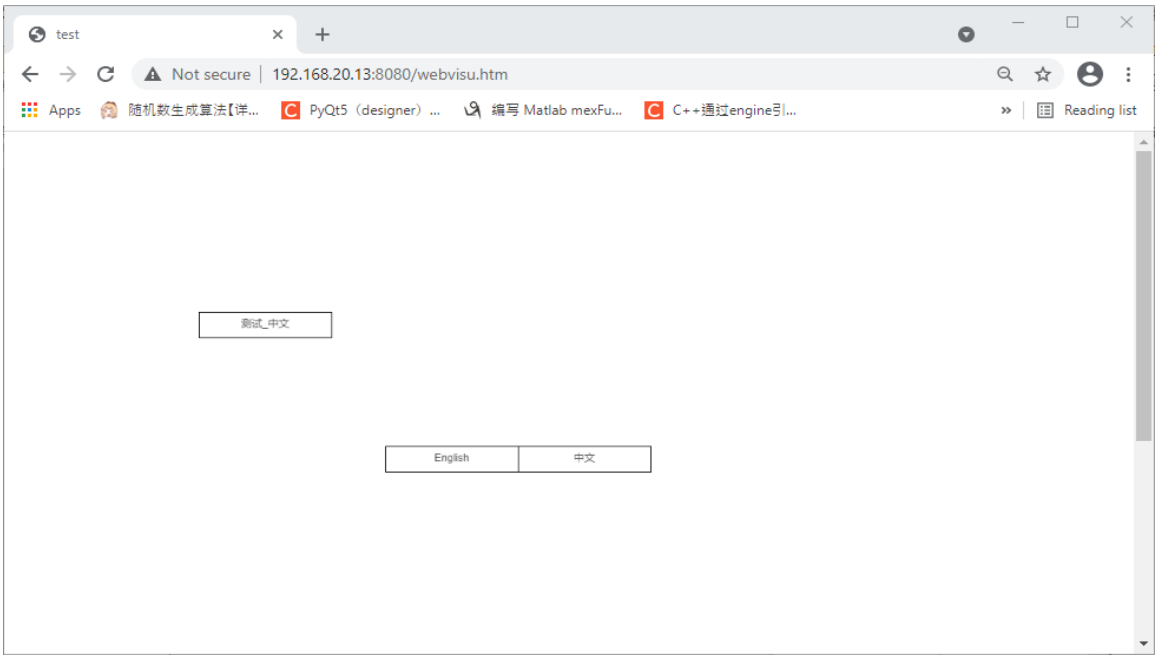
在项目开发过程中，添加到 HMI 中的所有静态文本都会在默认语言下自动添加到带有 ID 的全局文本列表 Text List 中。



如果 HMI 需要使用多种语言切换，可将多种语言添加到全局文本列表 Text List 中。请注意 Visualization Manager 界面勾选“Use Unicode string”并在“Selected language”中选择“zh-CHS”才能在界面中显示中文。参考例程 Demo3.10_HMI。






test
Not secure
192.168.20.13:8080/webvisu.htm
Apps
随机数生成算法【详...
PyQt5（designer）...
编写 Matlab mexFu...
C++通过engine引...
Reading list
测试_中文
English
中文

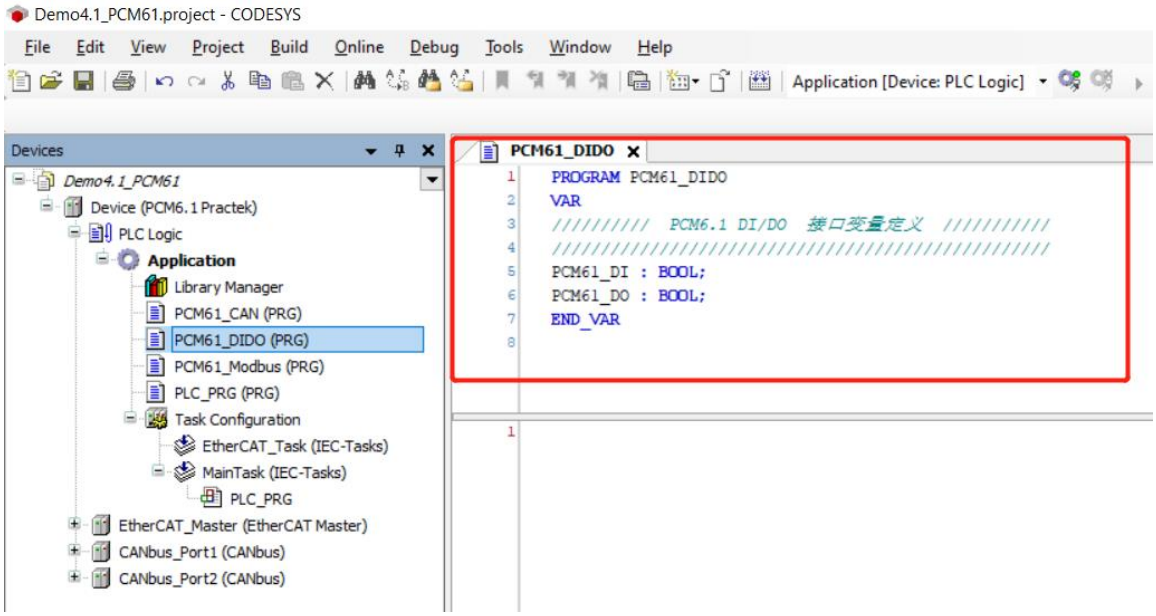
4 AWP100 程序配置

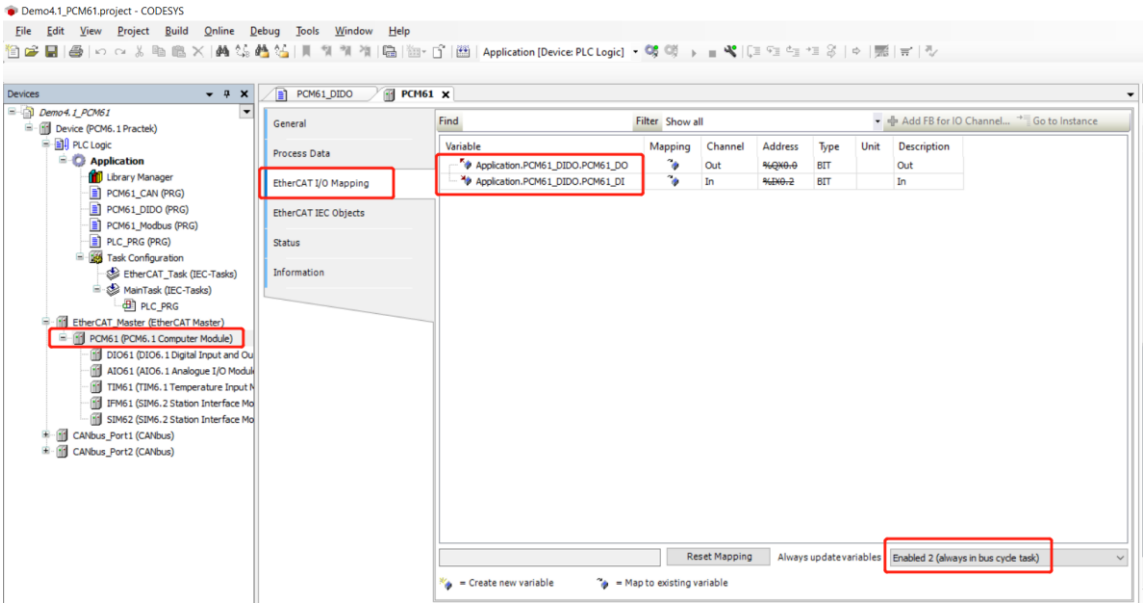
4.1 PCM6.1 程序配置

4.1.1 数字量程序配置

PCM6.1 模块具有 1 个数字量输入 DI 和 1 个数字量输出 DO，它们通常被用作控制器的看门狗干接点输入和输出，当用户系统或控制器运行异常时，通过硬件节点发出信号触发相应的硬件保护逻辑。

程序中完成 DI、DO 变量定义以后，在“EtherCAT I/O Mapping”中配置变量，并将右下角的“Always update variables”更改为“Enabled 2(always in bus cycle task)”，保证每个周期数据更新，无论该通道是否在程序中被调用。参考例程 Demo4.1_PCM61。

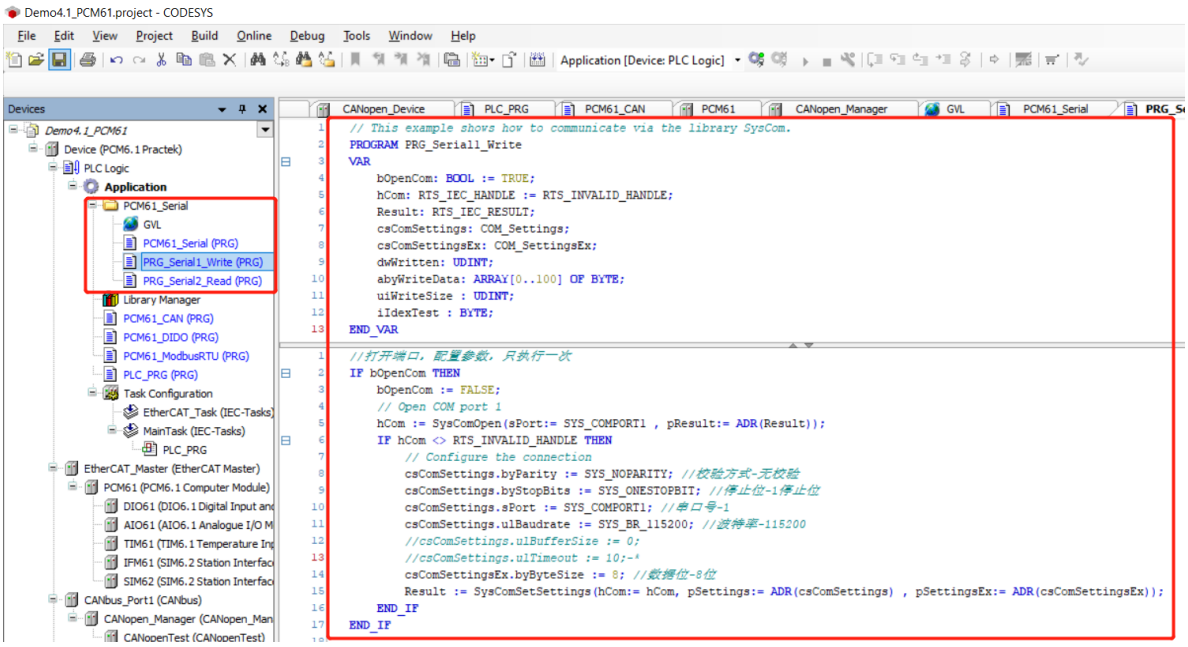




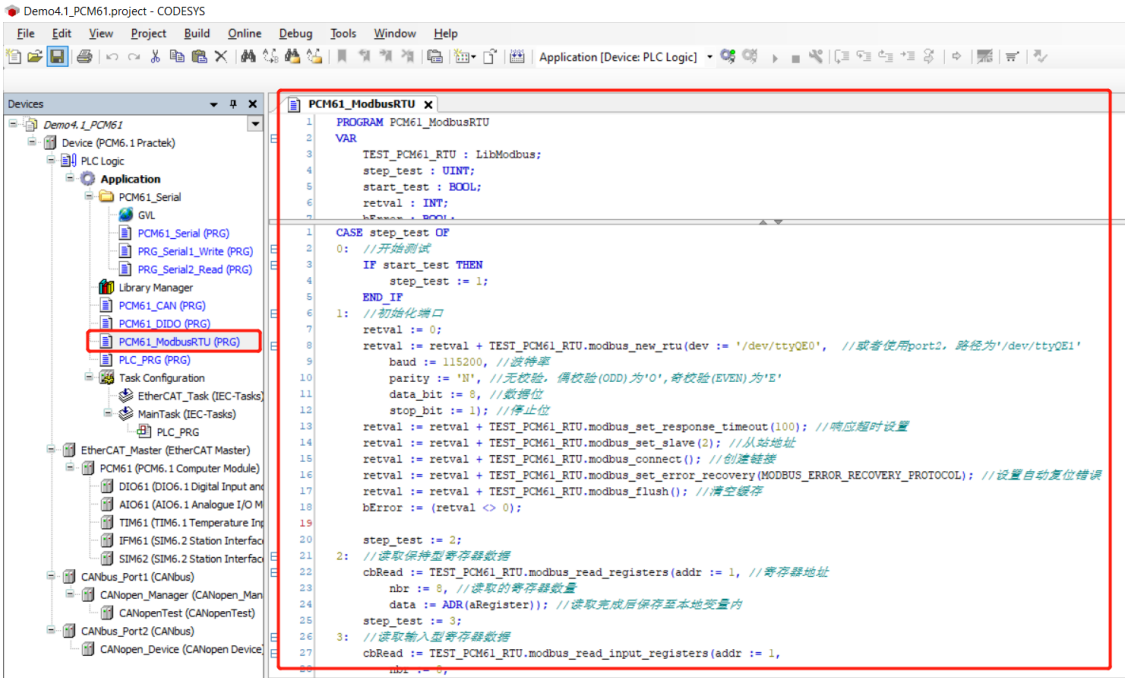
4.1.2 串口程序配置

PCM6.1 模块具有 2 个 RS-422/485 通道，支持 Modbus-RTU 通信，相关参数配置需要通过代码实现。参考例程 Demo4.1_PCM61。

采用 RS-485 通信时，端口选择、校验方式、停止位、波特率、数据位以及调用 Syscom 库函数的使用，全部通过代码实现。COM1 发送数据，COM2 接收数据。

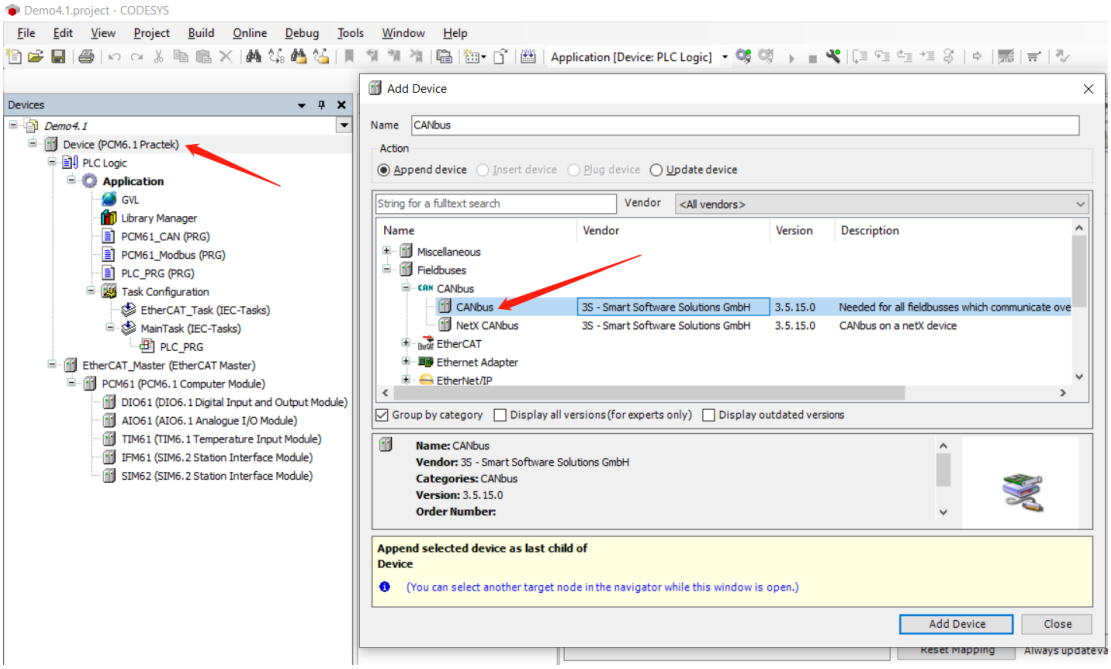


采用 Modbus-RTU 主站通信时，端口选择、校验方式、停止位、波特率、数据位以及调用 Libmodbus 库函数的使用，全部通过代码实现。

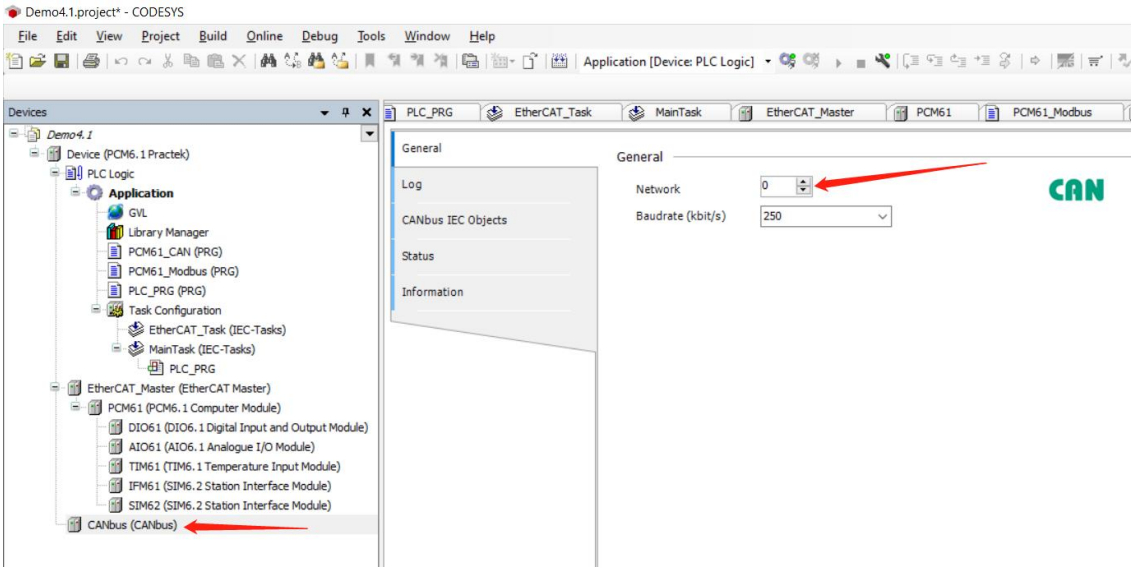


4.1.3 CANopen 程序配置

PCM6.1 模块具有 2 个 CAN 通道，每个接口均可以作为主站或从站使用。CAN bus 需要手动添加，右键点击“Device”选择“Add Device/CANbus”，弹出对话框选择“CANbus”添加。

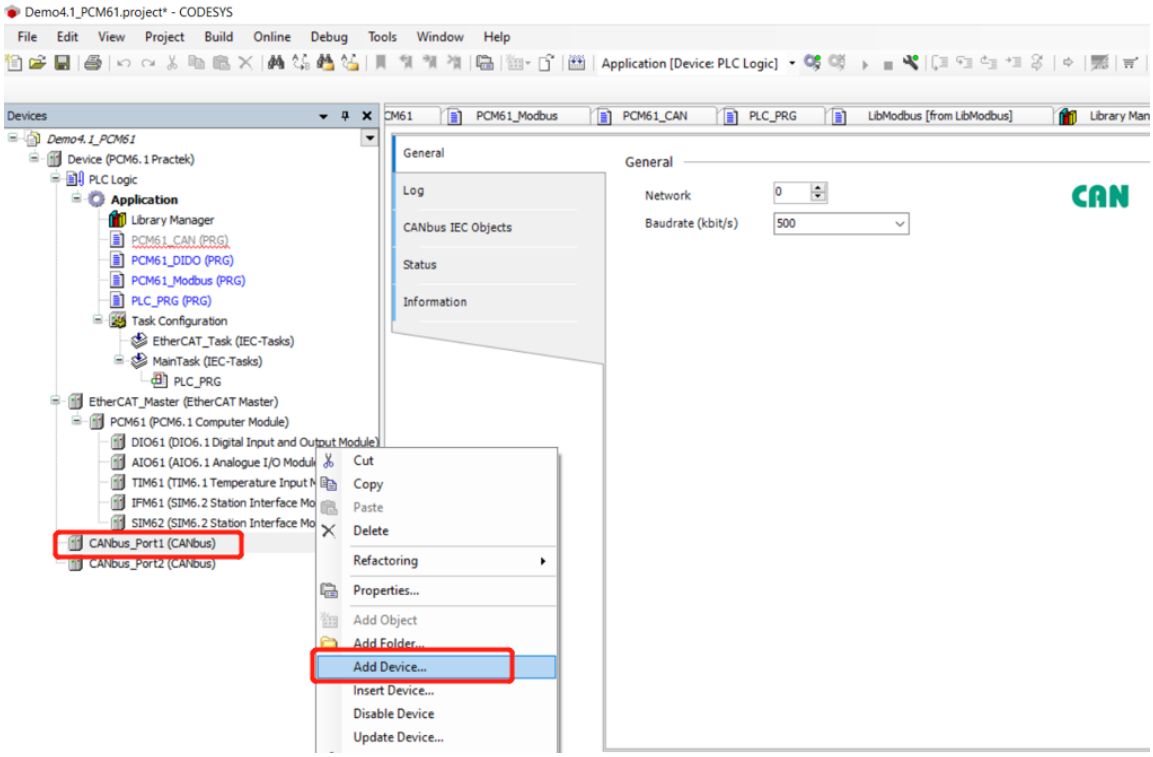


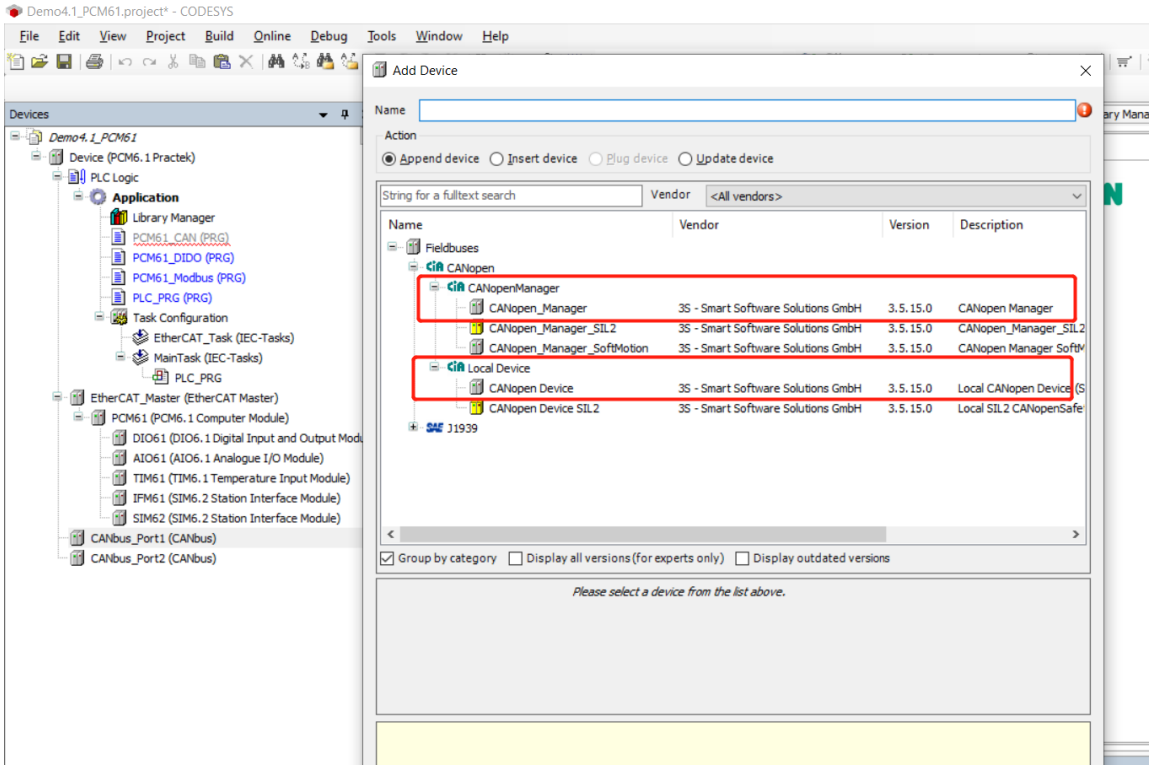
点击新添加的“CANbus/General”，将“Network”设置为 0，即使用 PCM6.1 的 CAN 接口 1。CAN Network 编号在控制器内部是从 0 开始编排的，即 PCM6.1 的 CAN 接口 1 对应 Network 编号为 0，PCM6.1 的 CAN 接口 2 对应 Network 编号为 1。当 AWP100 配置其他 CAN 功能模块时，其 CAN 接口将依次被编号为 2、3、4 等。



右键点击刚添加的“CANbus”，弹出窗口“Add Device/Fieldbuses/CANopen”有两种选择：

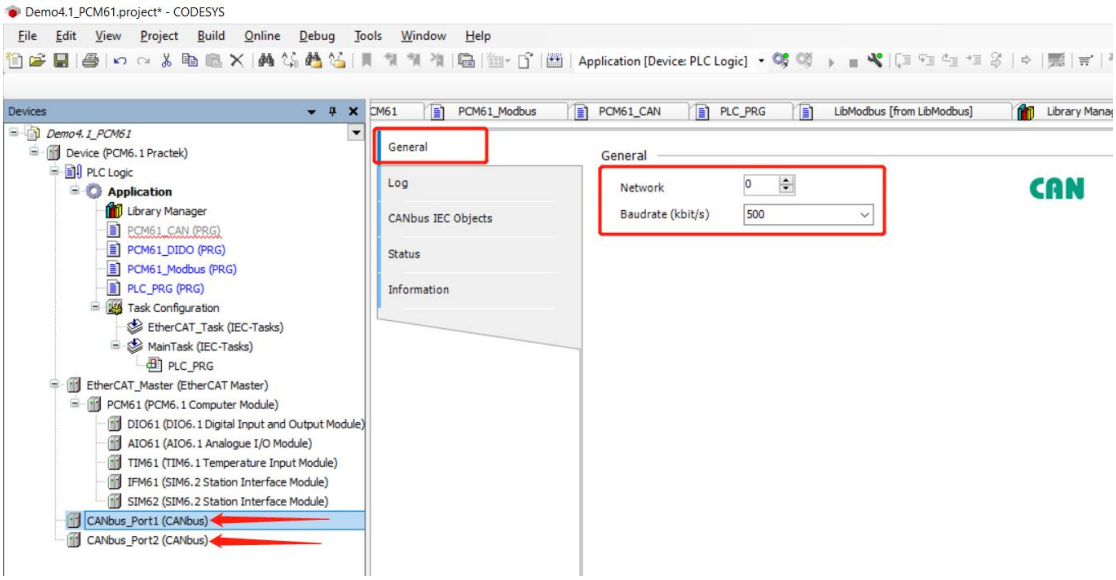
- CANopen 主站：CANopenManager/CANopen_Manager。
- CANopen 子站：Local Device/CANopen Device。



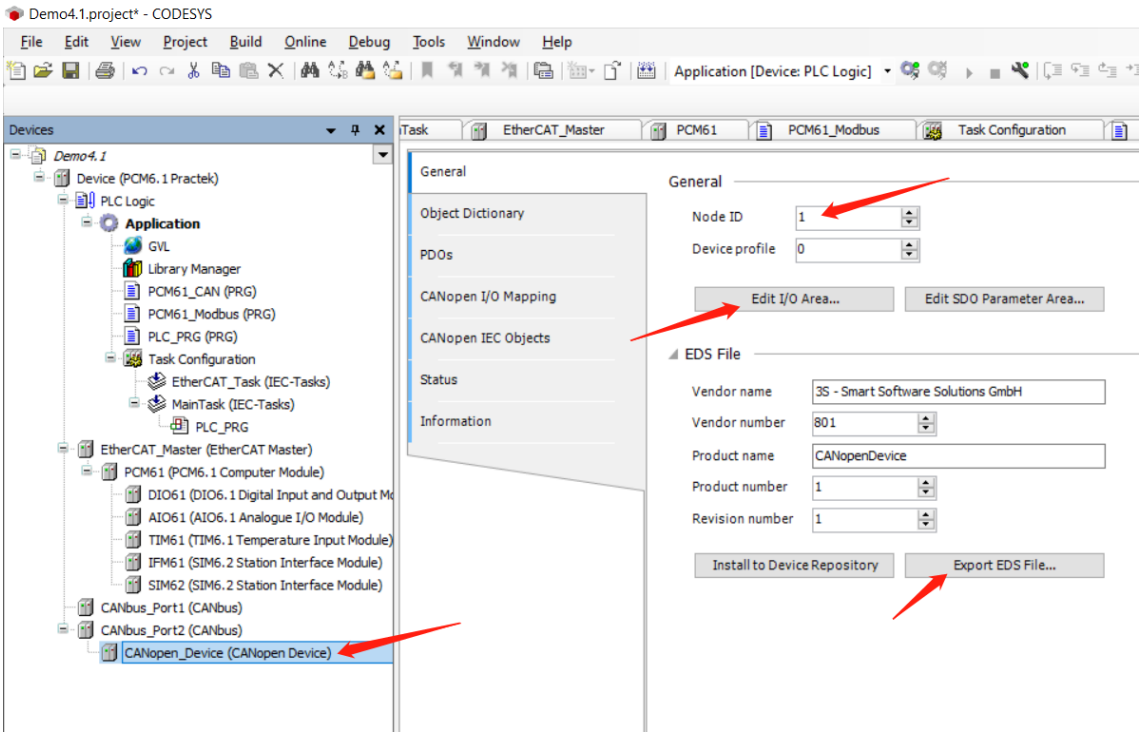


由于 PCM6.1 模块具有两个 CAN 接口，下面将 CAN 接口 1 配置为 CANopen 主站，将 CAN 接口 2 配置为 CANopen 子站，使其相互通讯。参考例程 Demo4.1_PCM6.1。

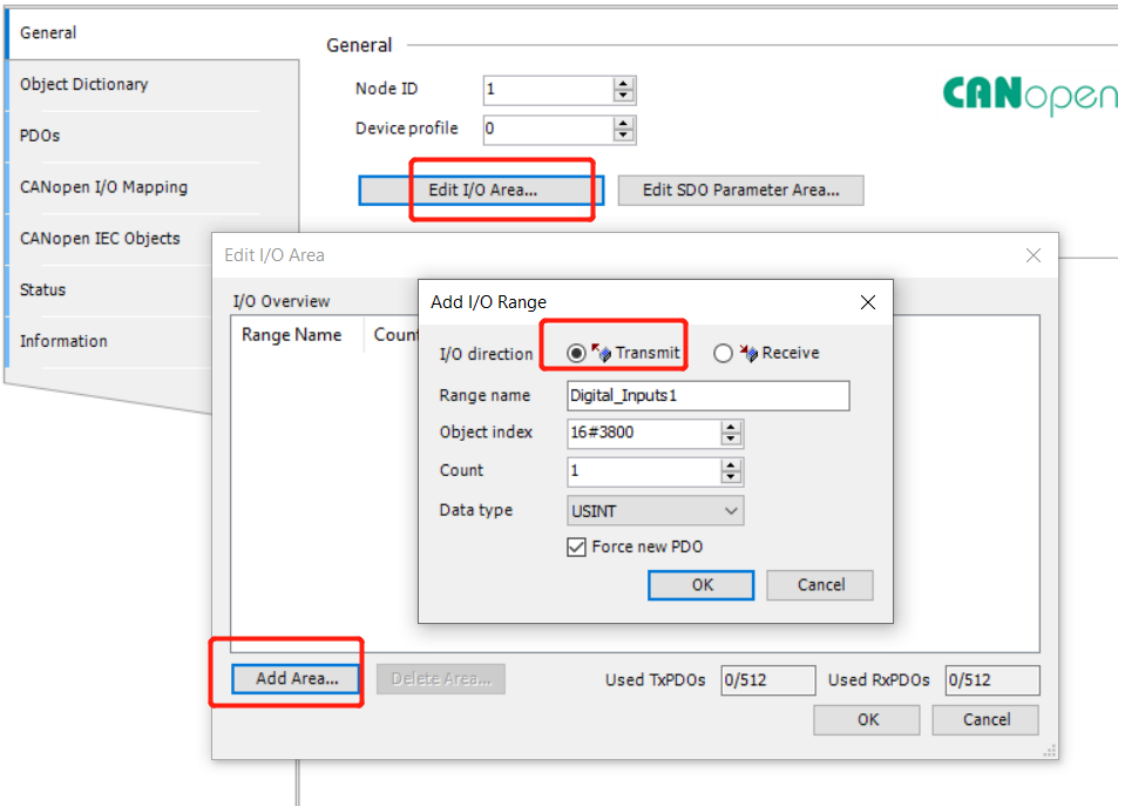
（1）按照前面所述方法添加两个 CANbus 设备，分别命名为 CANbus_Port1 和 CANbus_Port2，修改 CANbus_Port1 的 Network 为 0，修改 CANbus_Port2 的 Network 为 1，并将两个接口的波特率设置为 500kbit/s。

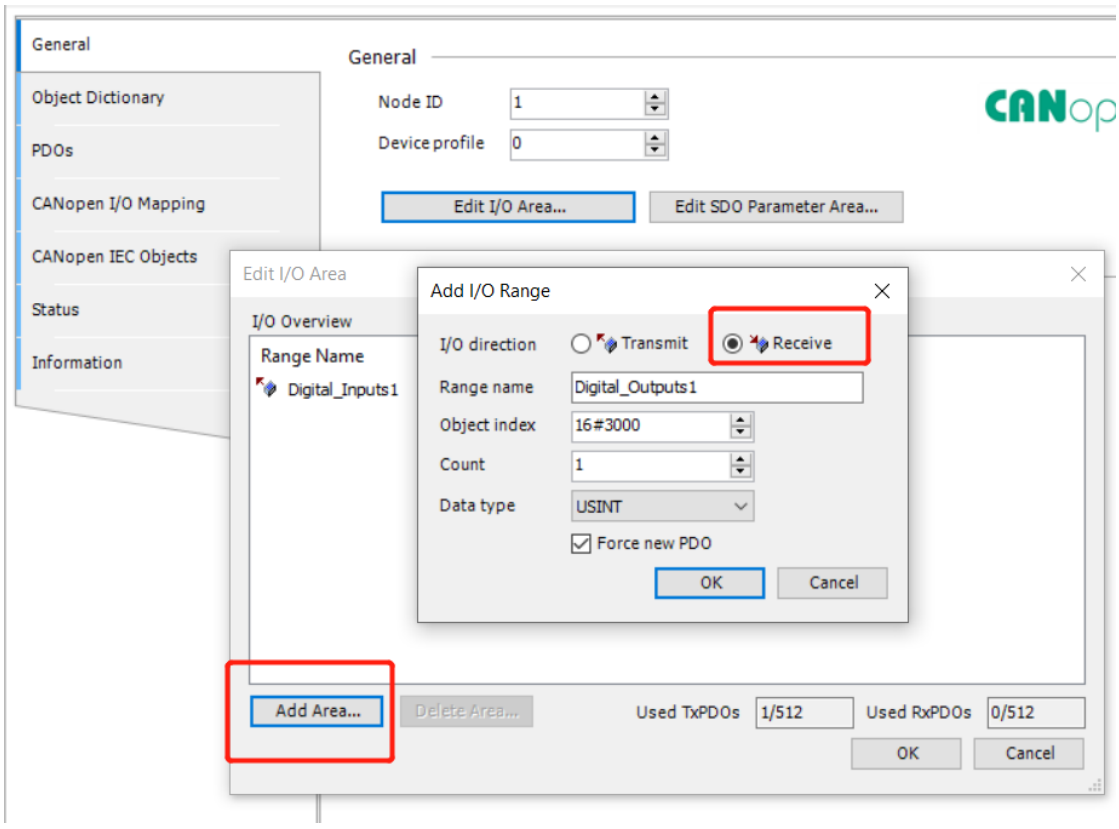


（2）点击“CANbus_Port2”添加“Local Device/CANopen Device”，并在“CANopen_Device/General”标签中将“Node ID”设置为 1。



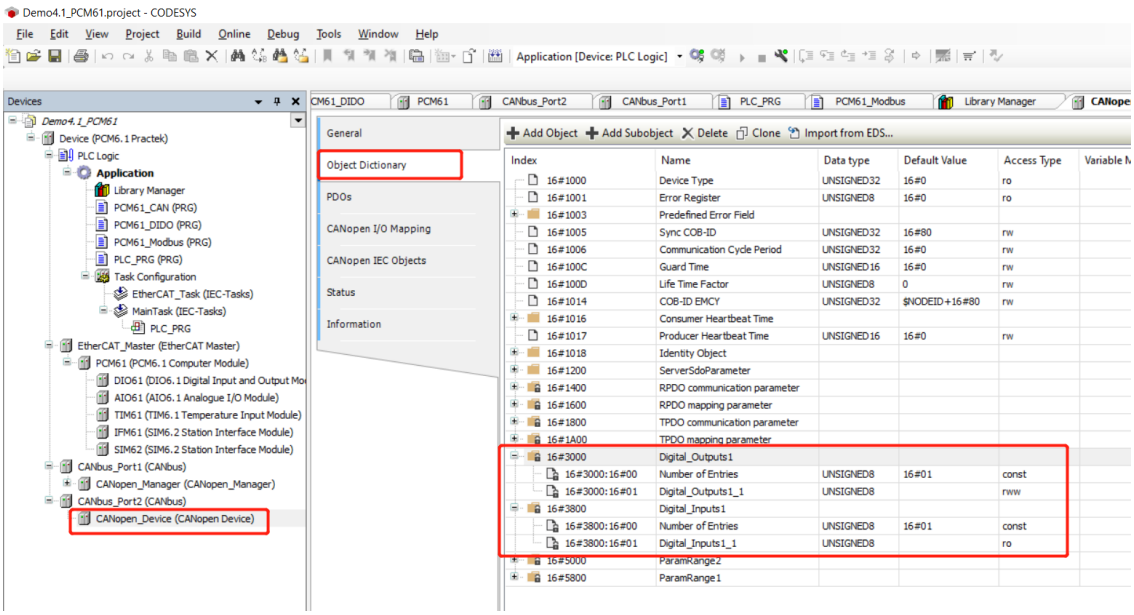
点击“Edit I/O Area”、“Add Area”，添加输入输出数据通道配置，这里添加一个 USINT 类型的输入和一个 USINT 类型的输出。



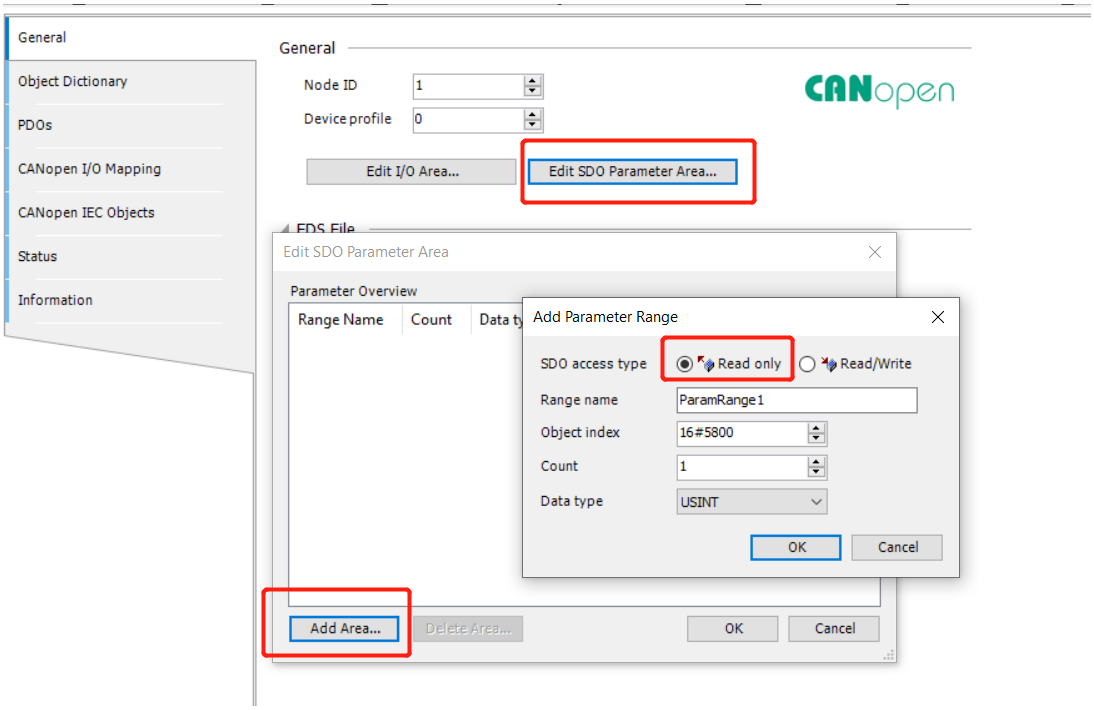


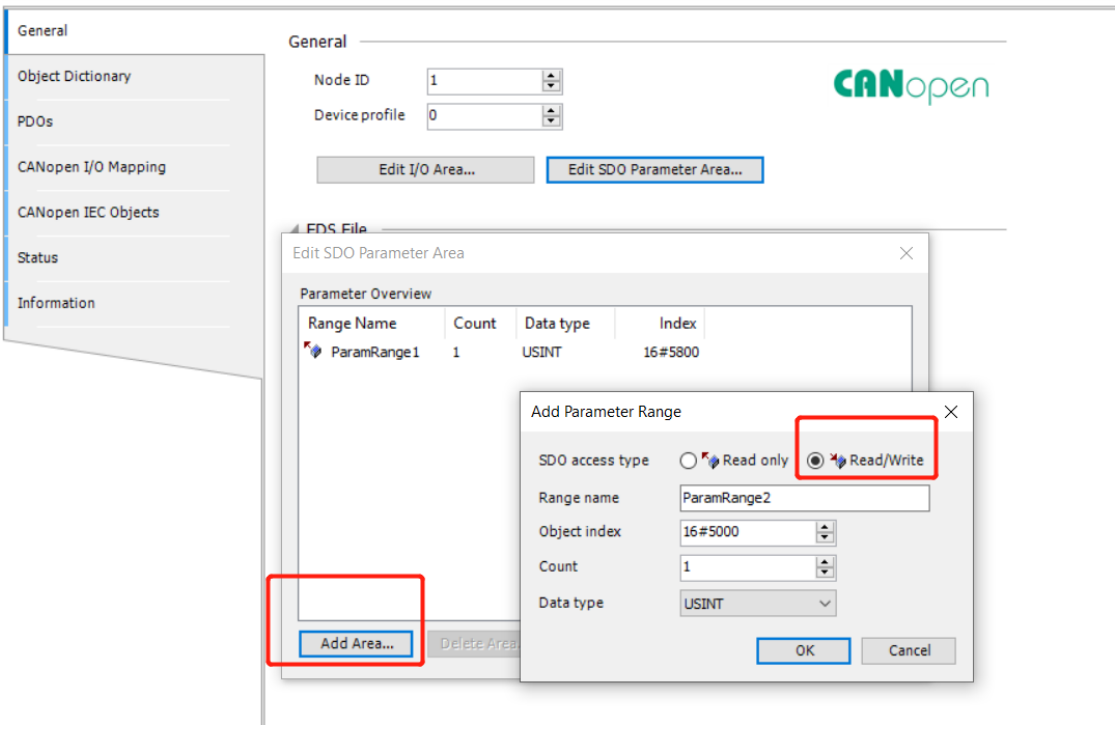
PDO 配置页面描述	
Transmit	主站接收数据，子站发送数据
Receive	主站发送数据，子站接收数据
Range name	自定义数据名称
Count	选择每组数据数量
Data type	选择每组数据类型
Force new PDO	勾选表示新建一组数据，否则在原数据组添加

添加完成 USINT 类型的输入和 USINT 类型的输出，可以在“Object Dictionary”内查看新添加数据。

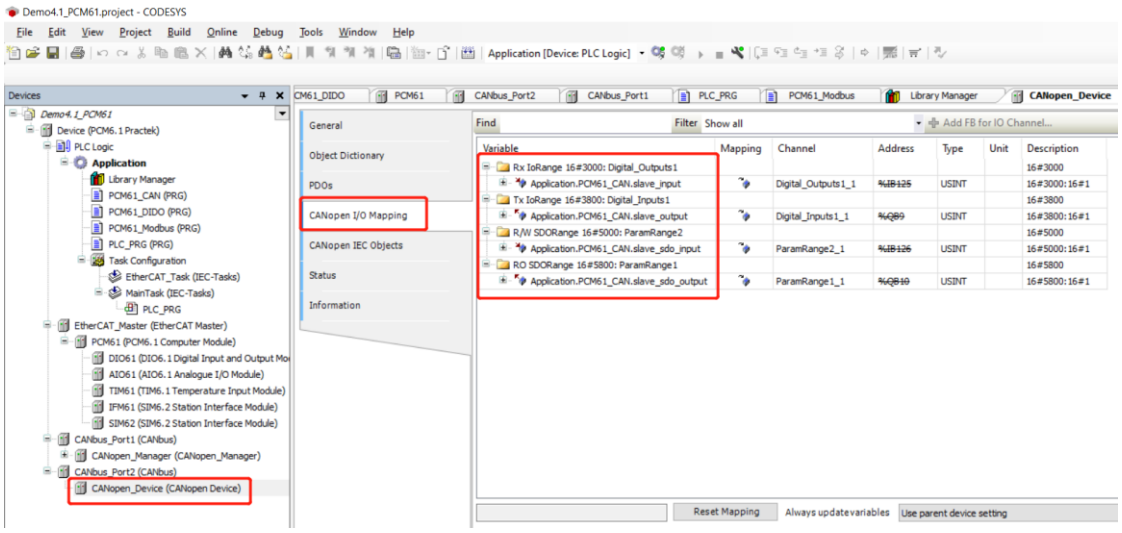


点击“Edit SDO Parameter Area”，可以配置从站的 SDO 通讯，这里添加一个 USINT 类型的输入和一个 USINT 类型的输出。

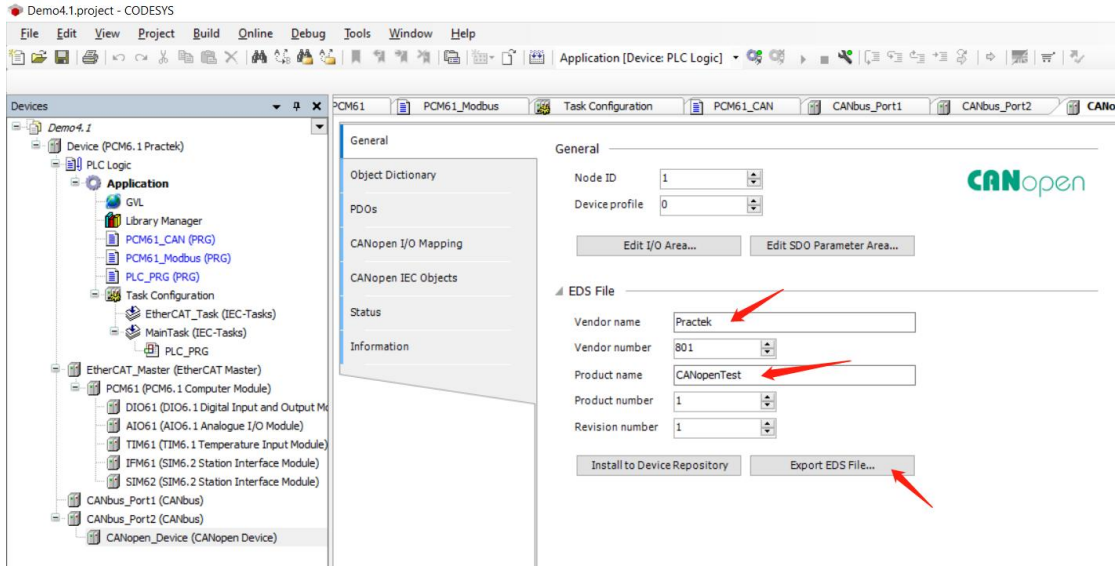

General
Object Dictionary
PDOs
CANopen I/O Mapping
CANopen IEC Objects
Status
Information
General
Node ID
1
Device profile
0
CANopen
Edit I/O Area...
Edit SDO Parameter Area...
EDS File
Edit SDO Parameter Area
Parameter Overview
Range Name
Count
Add Parameter Range
SDO access type
Read only
Read/Write
Range name
ParamRange1
Object index
16#5800
Count
1
Data type
USINT
OK
Cancel
Add Area...
Delete Area...
OK
Cancel


General
Object Dictionary
PDOs
CANopen I/O Mapping
CANopen IEC Objects
Status
Information
General
Node ID
1
Device profile
0
CANopen
Edit I/O Area...
Edit SDO Parameter Area...
EDS File
Edit SDO Parameter Area
Parameter Overview
Range Name
Count
Data type
Index
ParamRange1
1
USINT
16#5800
Add Parameter Range
SDO access type
Read only
Read/Write
Range name
ParamRange2
Object index
16#5000
Count
1
Data type
USINT
OK
Cancel
Add Area...

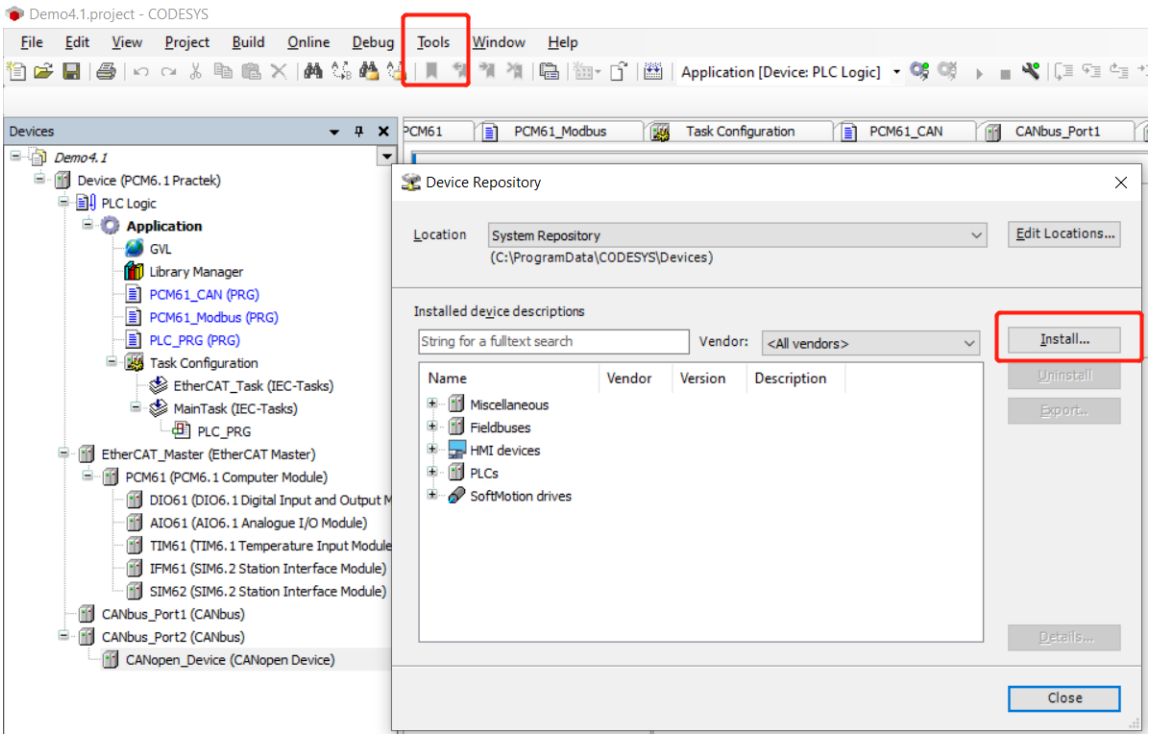
（3）在“CANopen_Device / CANopen I/O Mapping”页面关联程序变量。



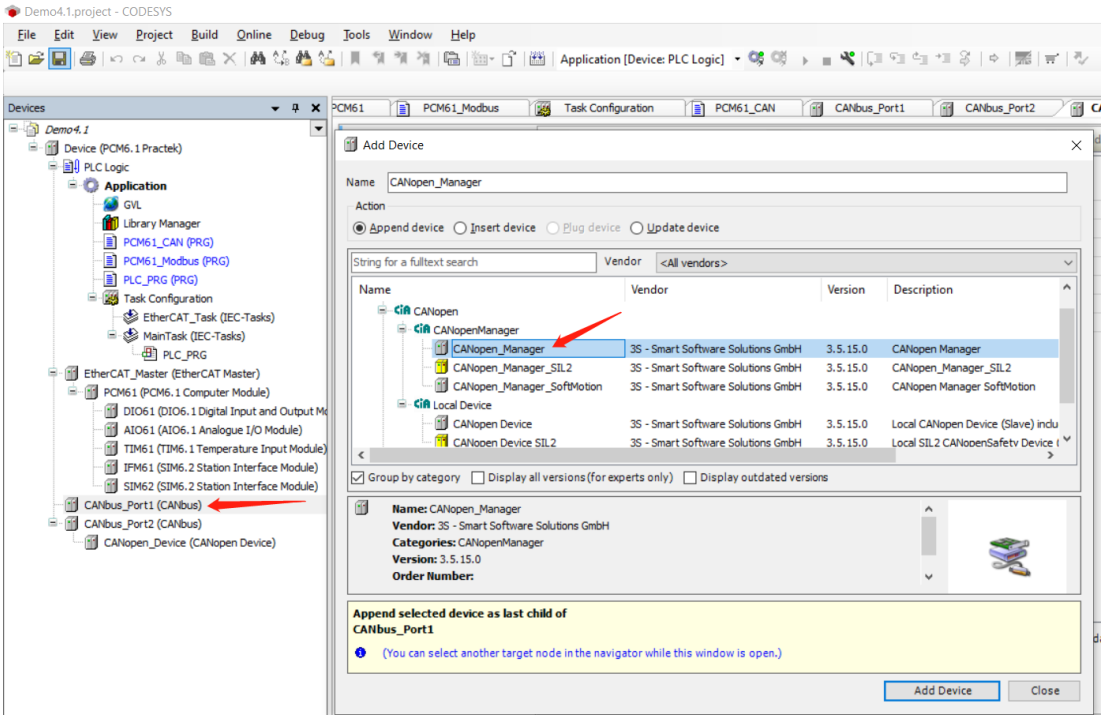
（4）在“CANopen_Device/General”页面，点击“Export EDS file”导出从站的 EDS 文件，以方便主站配置添加。Vendor name、Product name 等信息可根据实际情况进行配置。



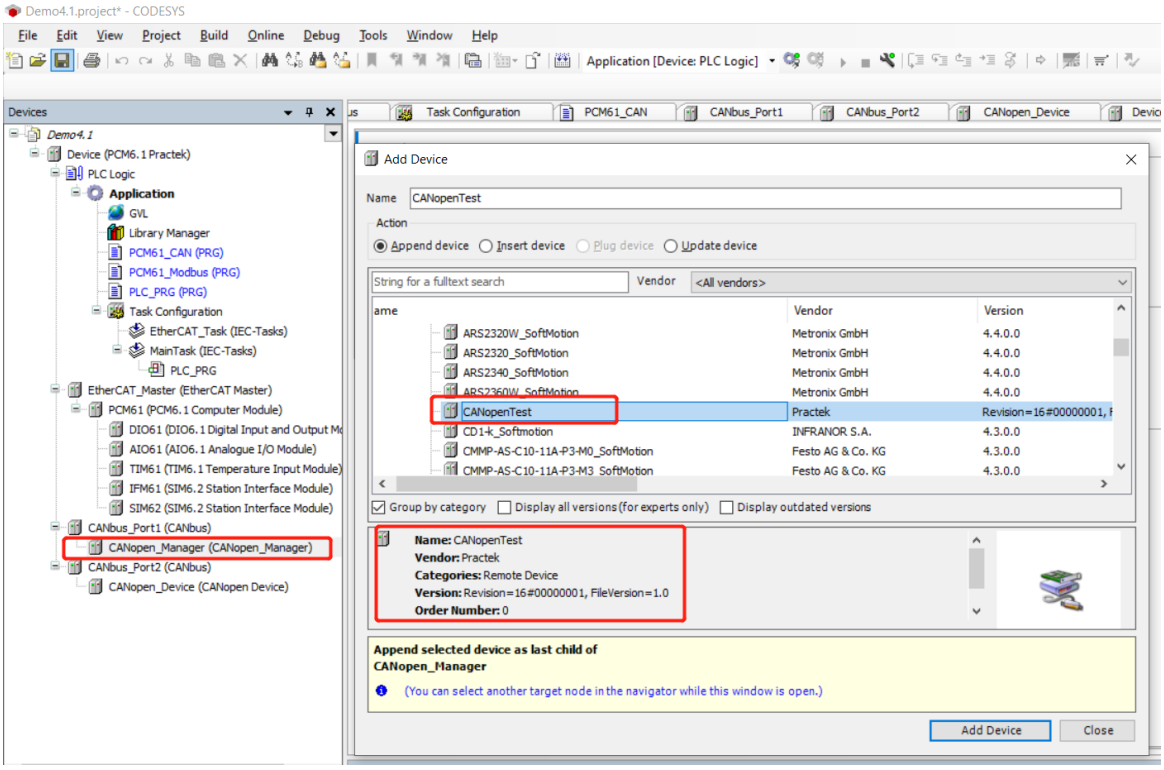
（5）点击菜单栏“Tools/Device Respository”，在弹出窗口中点击“Install”，选择子站 EDS 文件加载添加。



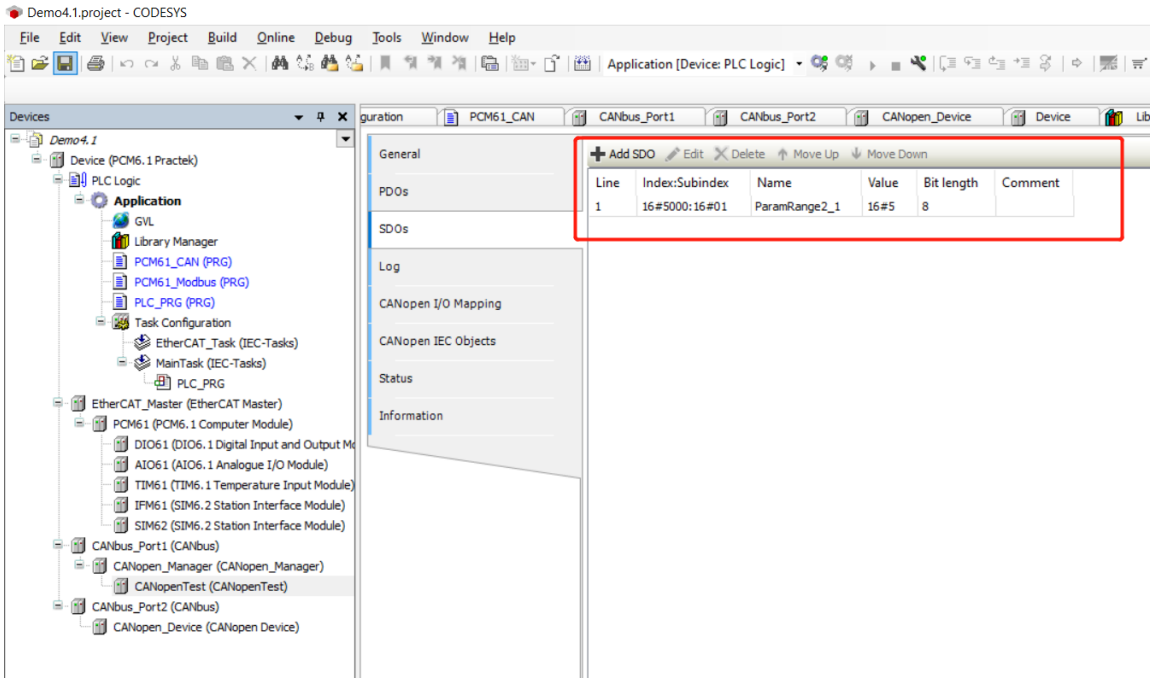
（6）右键点击“CANbus_Port1”添加 CANopen 主站设备，选择“Add Device/CANopen_Manager”。



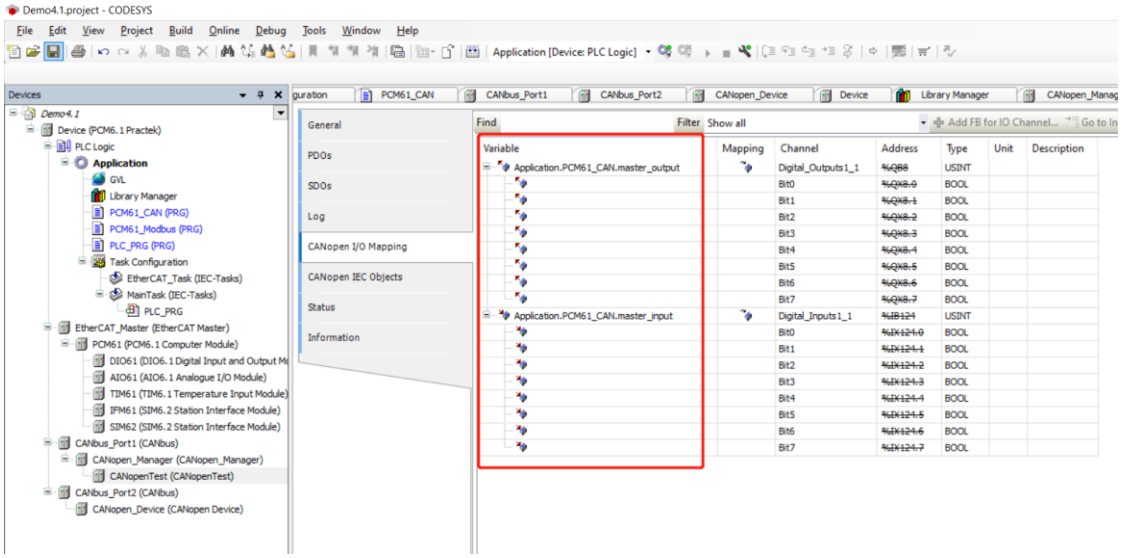
（7）右键点击刚添加的“CANopen_Manager”，选择“Add Device”，弹出窗口选择已经完成添加的子站设备。



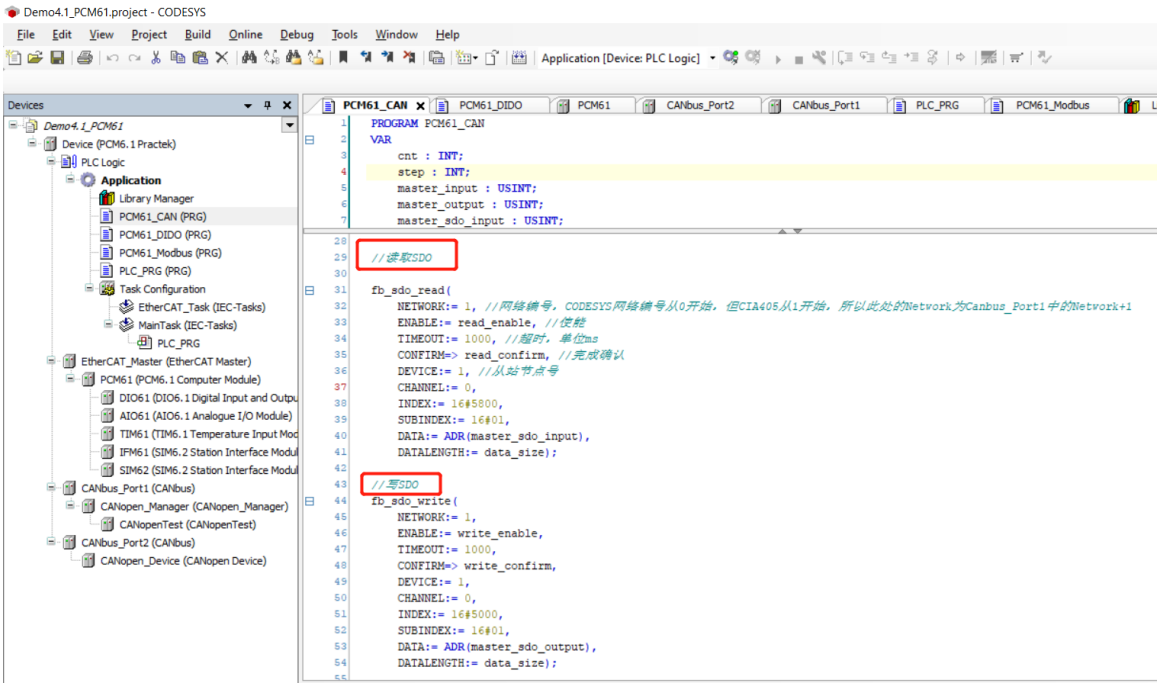
（8）在主站“SDOs”页面中可以设置 SDO 启动参数，即在从站运行开始时主站即可发送给从站的 SDO 参数。在主站“CANopen I/O Mapping”页面中进行变量链接。



（9）在主站“CANopen I/O Mapping”页面中进行 PDO 变量链接配置。



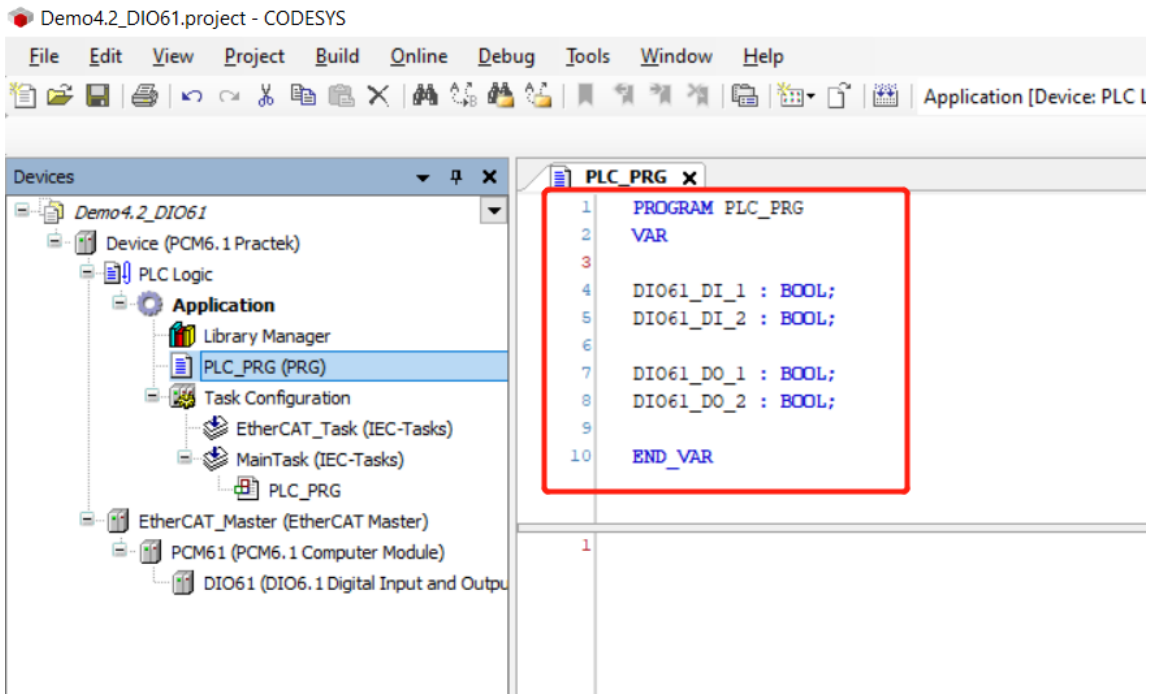
正常通讯阶段，写入和读取 SDO 数据不能通过在 CANopen I/O Mapping 中关联变量的方式进行，而需要使用库函数 CiA405.SDO_READ()以及 CiA405.SDO_WRITE()来实现，而且 CODESYS 对 CAN Network 的编号自 0 开始，而 CiA405 对 CAN Network 的编号自 1 开始。

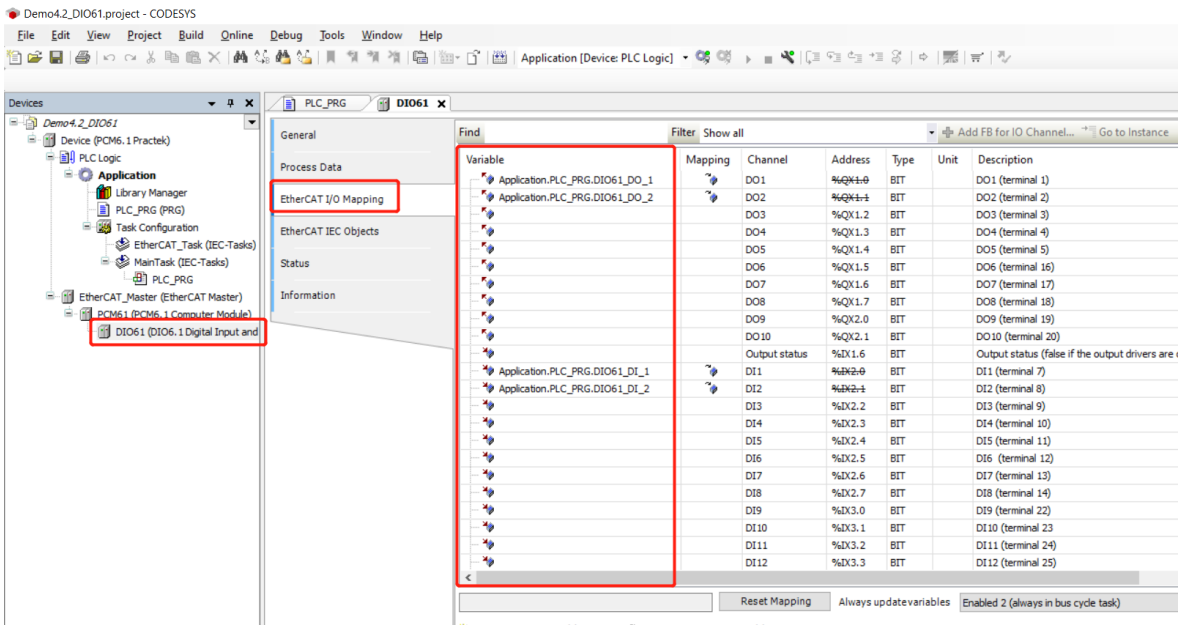


4.2 DIO6.1 程序配置

DIO6.1 为数字量输入输出模块，具有 10 个 DO 和 16 个 DI。程序中声明变量后，可直接在“EterCAT I/O Mapping”中将变量关联到硬件通道上。

参考例程 Demo4.2_DIO61。



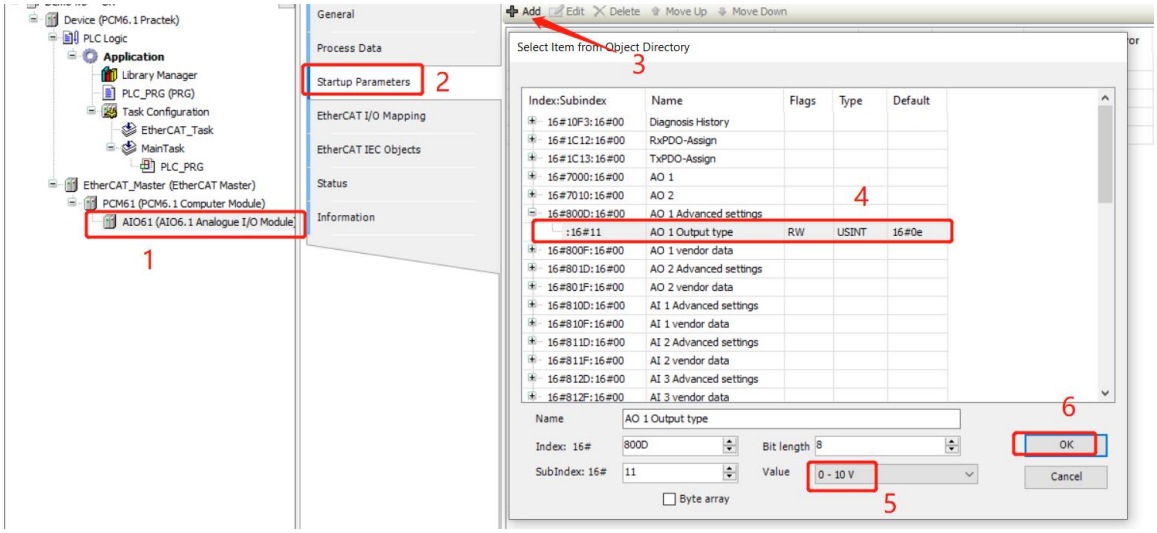


4.3 AIO6.1 程序配置

AIO6.1 为模拟量输入输出模块，具有 2 个 AO 和 16 个 AI。根据实际使用传感器的信号类型需要逐个通道进行启动参数配置。启动参数不可以批量操作，每次新建启动参数只可以配置一个通道。

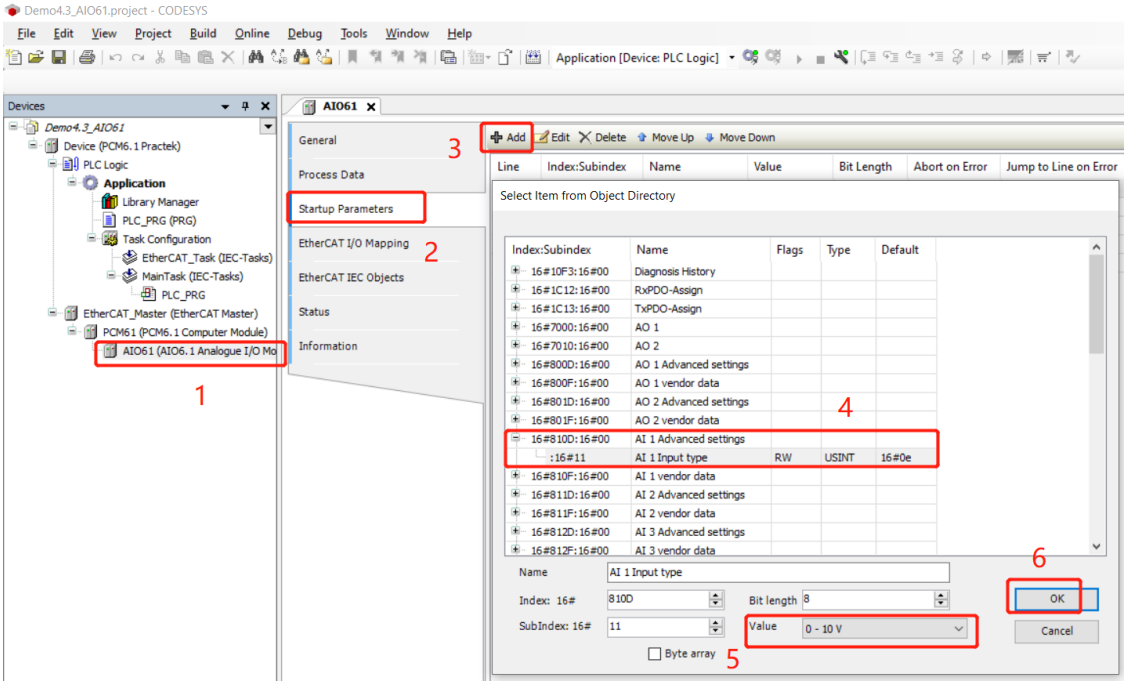
AO1 启动参数配置方法：

- 点击设备“AIO6.1/Startup Parameters”。
- 点击“Add”，弹出对话框中进行启动参数设置，点击“AO 1 Advanced settings/AO 1 Output type”。
- 选择“Value”数据范围 0~10 V / 0~20 mA / 4~20 mA。

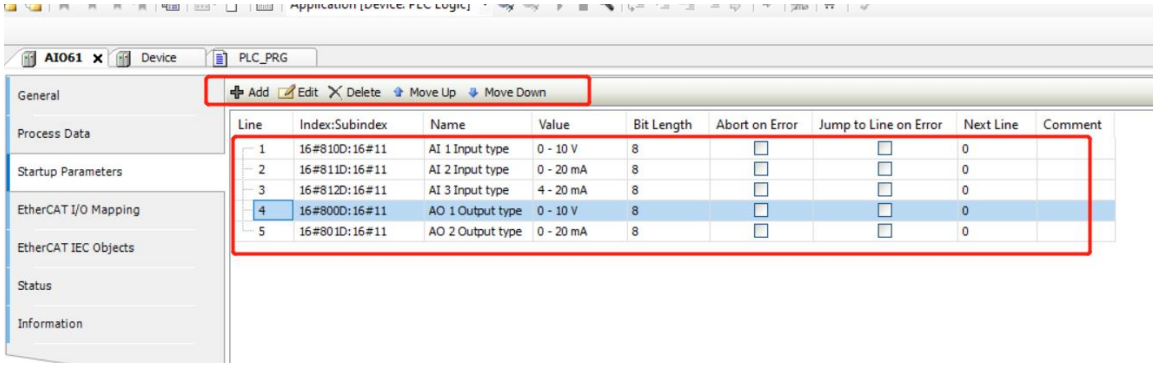


AI1 启动参数配置方法：

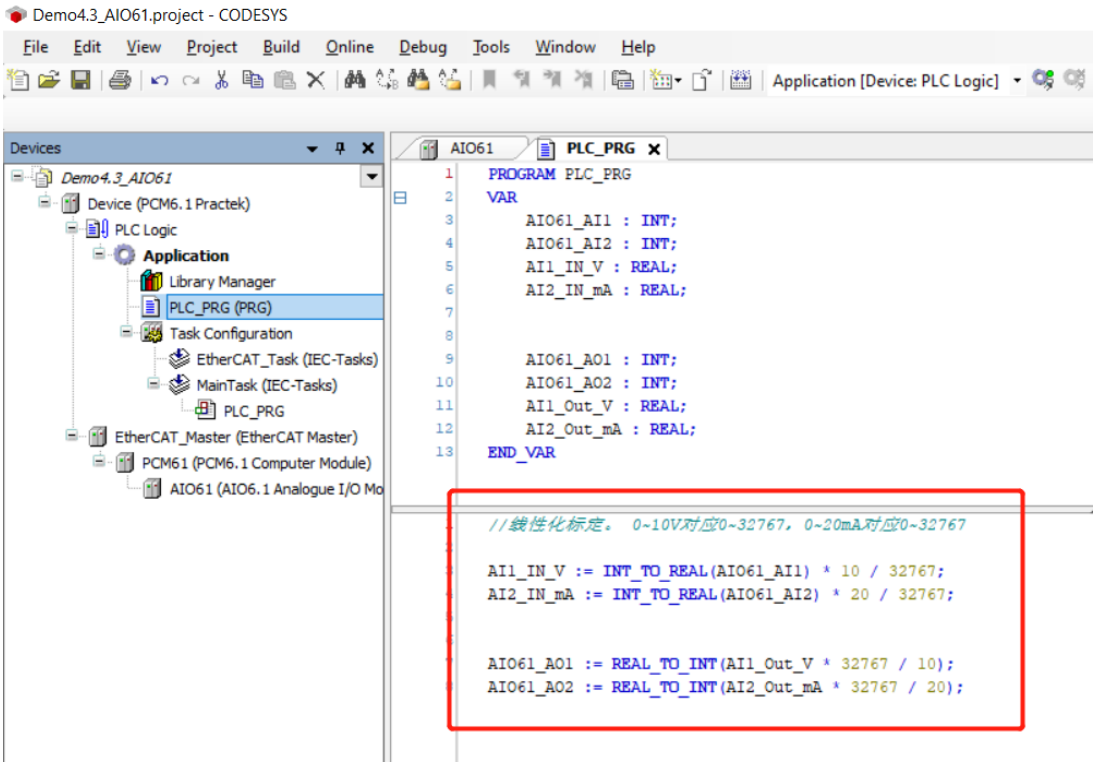
- 点击设备“AIO6.1/Startup Parameters”。
- 点击“Add”，弹出对话框中进行启动参数设置，点击“AI 1 Advanced settings/AI 1 Input type”。
- 选择“Value”数据范围 0~10 V / 0~20 mA / 4~20 mA。



模拟量启动参数设置完成后，可以通过工具栏进行编辑、删除、上移、下移操作。每次点击“Add”添加只能配置一个通道的启动参数，配置多个参数需要多次点击“Add”。

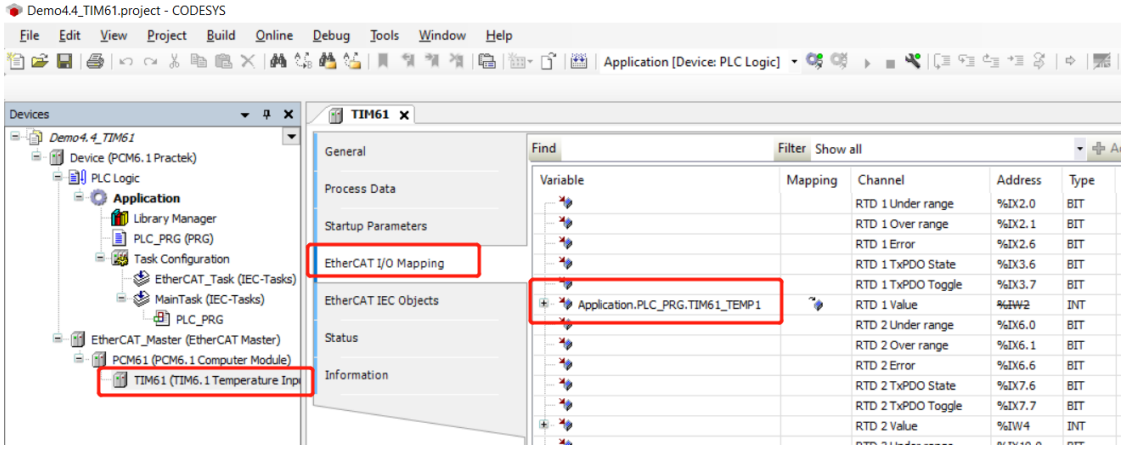


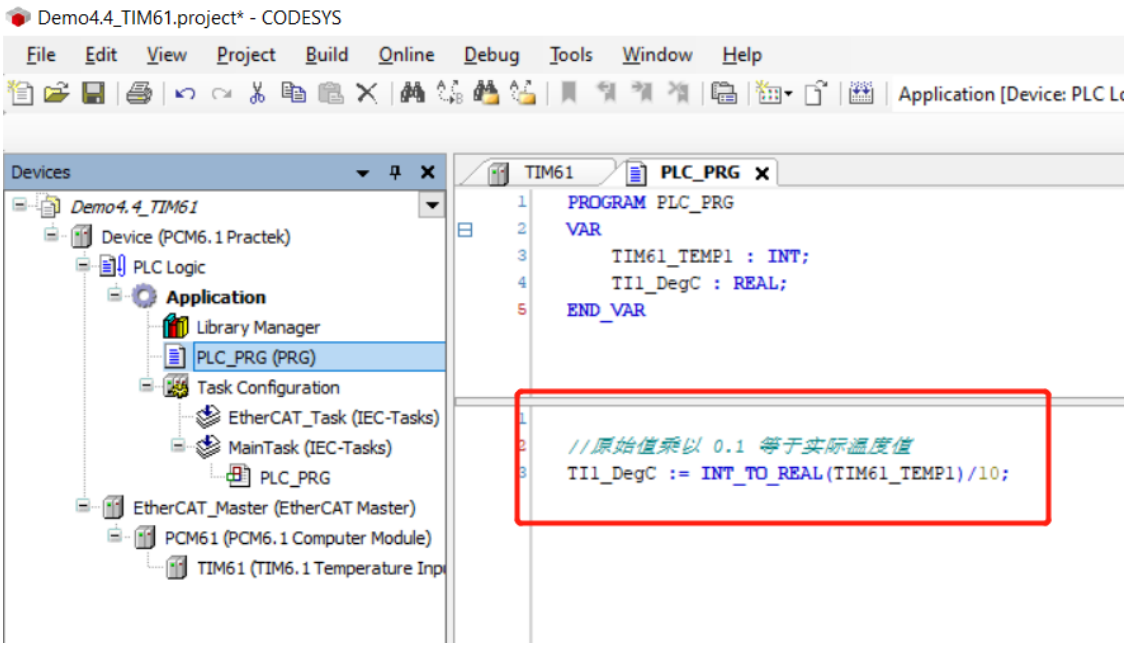
模拟量数据需要标定计算，0~10V/0~20mA 线性对应 0~32767，参考例程 Demo4.3_AIO61。



4.4 TIM6.1 程序配置

TIM6.1 为温度输入模块，具有 14 个 TEMP 温度输入。在程序中声明变量后，可直接在“EterCAT I/O Mapping”中将变量关联到硬件通道上。温度输入通道的原始值需要乘以 0.1 换算成实际温度值，参考例程 Demo4.4_TIM61。





4.5 IFM6.1 程序配置

4.5.1 Profibus DP 程序配置

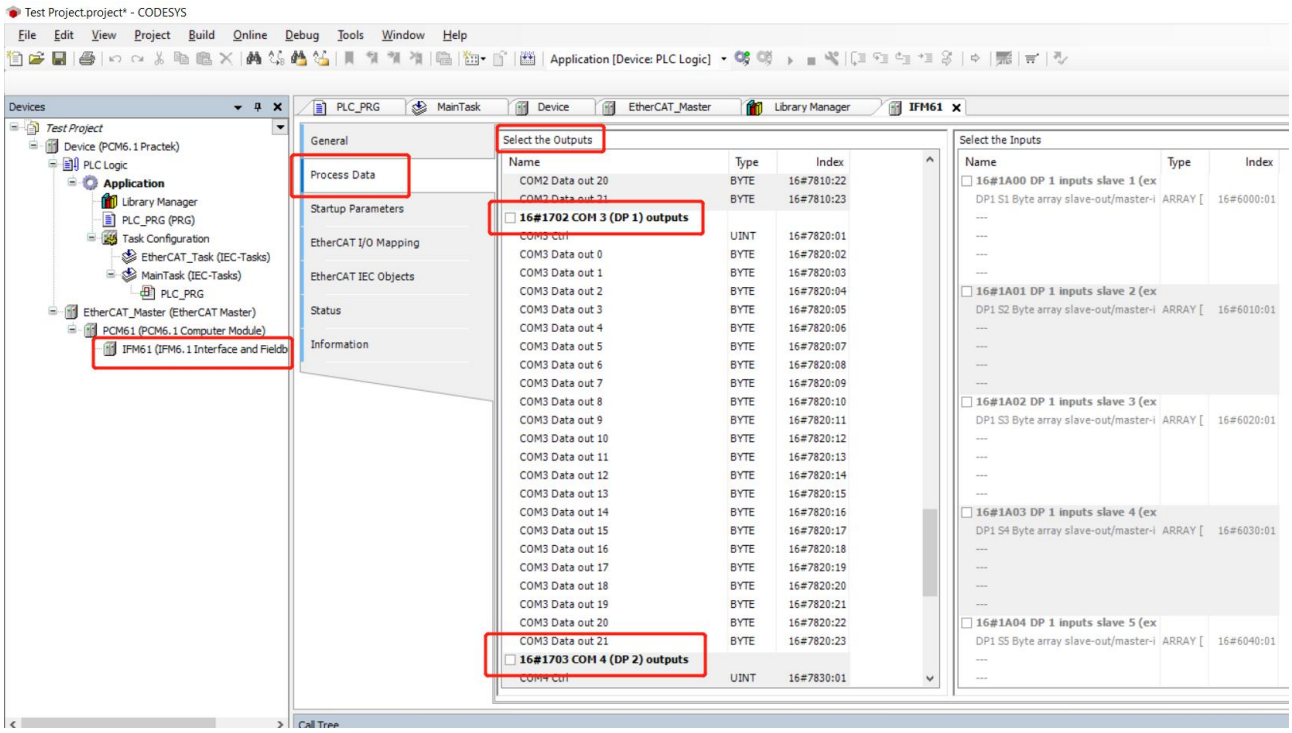
IFM6.1 通信模块具有 2 个 Profibus DP Master 端口，在进行 Profibus DP 变量链接之前，需要进行 Process Data 设置。IFM6.1 提供一个默认 122 字节的数组来实现与 Profibus DP 子站的数据交互，该数组与 Profibus 通信数据的映射是由 PDO 实现的。

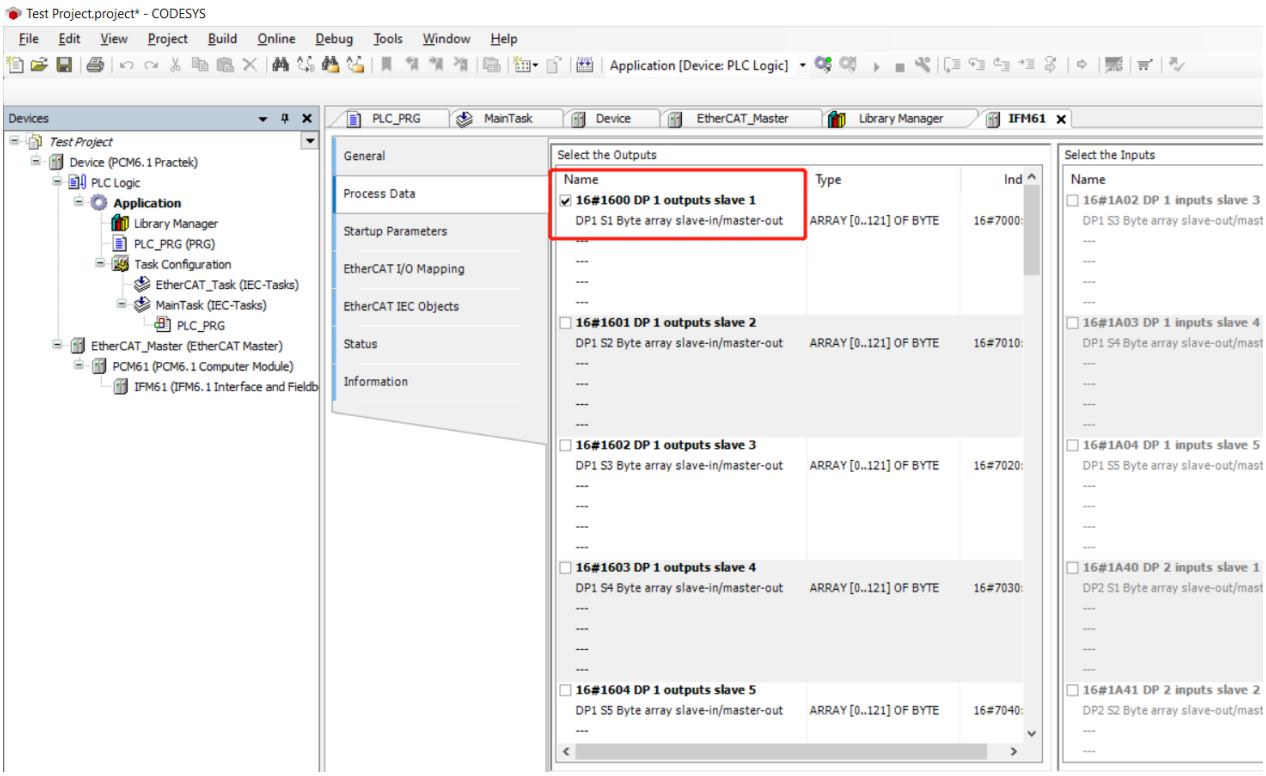
Process Data 设置需要将“16#1702”替换成“16#1600”（用于 slave1）、“16#1601”（用于 slave2）、“16#1602”（用于 slave3）、“16#1603”（用于 slave4）、“16#1604”（用于 slave5）；将“16#1B02”替换成“16#1A00”（用于 slave1）、“16#1A01”（用于 slave2）、“16#1A02”（用于 slave3）、“16#1A03”（用于 slave4）、“16#1A04”（用于 slave5），完成 Profibus DP1 的 slave 1、slave 2、slave 3、slave 4、slave 5 的 Process Data 设置。“16#1703”和“16#1B03”设置方法与上面相同，用于 Profibus DP2 的数据通信。

单击“IFM61”、“Process Data”进入 Process Data 配置页面。在“select the Outouts”分组内取消“16#1702”和“16#1703”的勾选，勾选“16#1600”和

“16#1640”，这样就设置了 Profibus Outputs 的 DP1 slave1 和 DP2 slave1。如果有两个子站同时连接主站，那么需要继续勾选“16#1601”和“16#1641”，设置 Profibus Outputs 的 DP1 slave2 和 DP2 slave2。

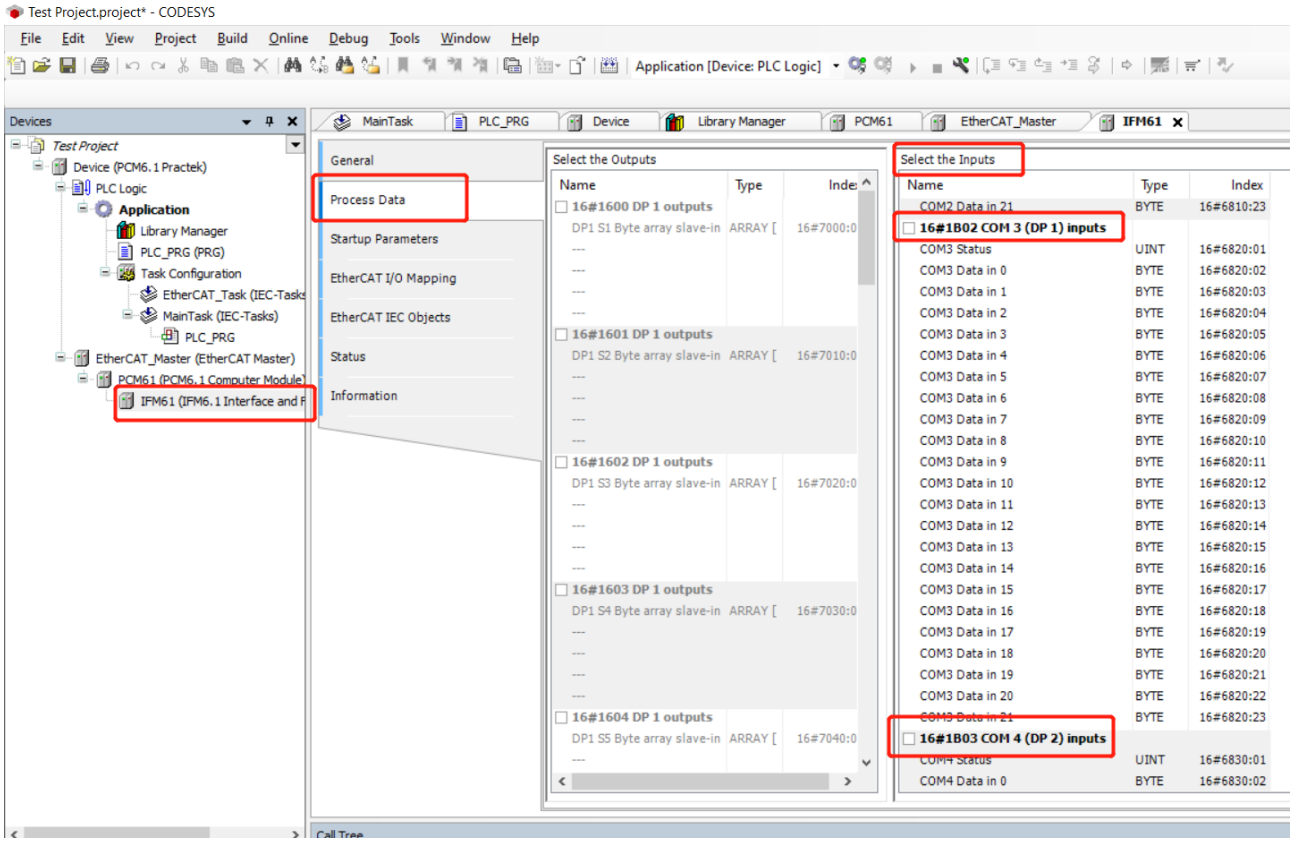
Profibus Outputs									
16#1702					16#1703				
16#1600	16#1601	16#1602	16#1603	16#1604	16#1640	16#1641	16#1642	16#1643	16#1644
DP1 slave1	DP1 slave2	DP1 slave3	DP1 slave4	DP1 slave5	DP2 slave1	DP2 slave2	DP2 slave3	DP2 slave4	DP2 slave5





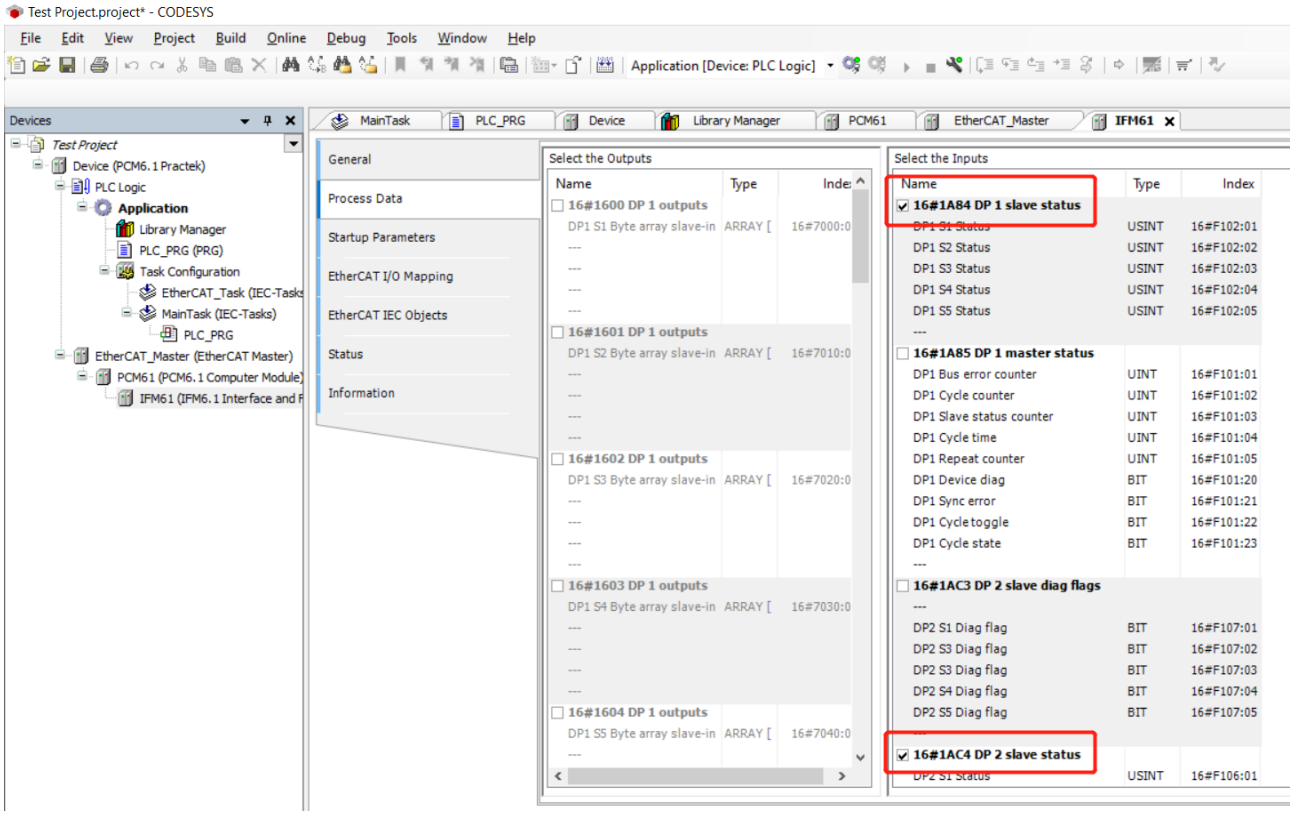
单击“IFM61”、“Process Data”进入 Process Data 配置页面。在“select the Inputs”分组内取消“16#1B02”和“16#1B03”的勾选，勾选“16#1A00”、“16#1A40”。这样就设置了 Profibus Inputs 的 DP1 slave1 和 DP2 slave1。如果有两个子站同时连接主站，那么需要继续勾选“16#1A01”和“16#1A41”，设置 Profibus Inputs 的 DP1 slave2 和 DP2 slave2。

Profibus Inputs									
16#1B02					16#1B03				
16#1A00	16#1A01	16#1A02	16#1A03	16#1A04	16#1A40	16#1A41	16#1A42	16#1A43	16#1A44
DP1 slave1	DP1 slave2	DP1 slave3	DP1 slave4	DP1 slave5	DP2 slave1	DP2 slave2	DP2 slave3	DP2 slave4	DP2 slave5

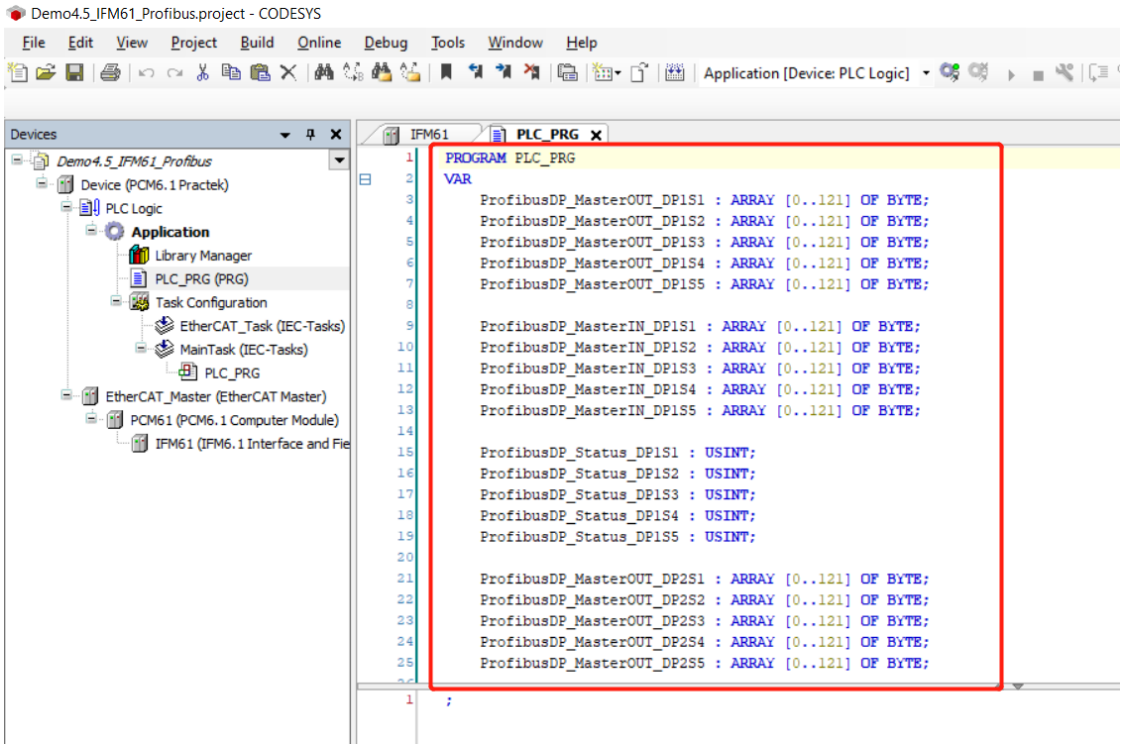


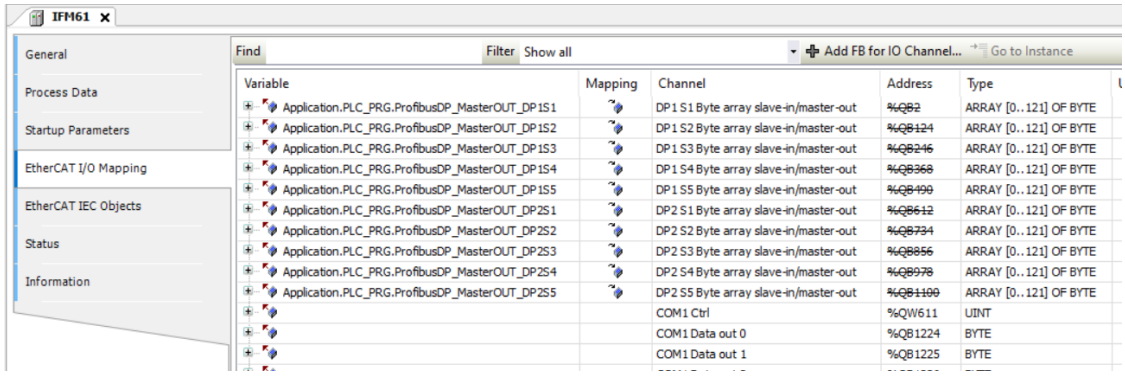
在“select the Inputs”分组内勾选“16#1A84”、“16#1AC4”，这样就设置了Profibus Inputs 的 DP1 和 DP2 子站通信状态。

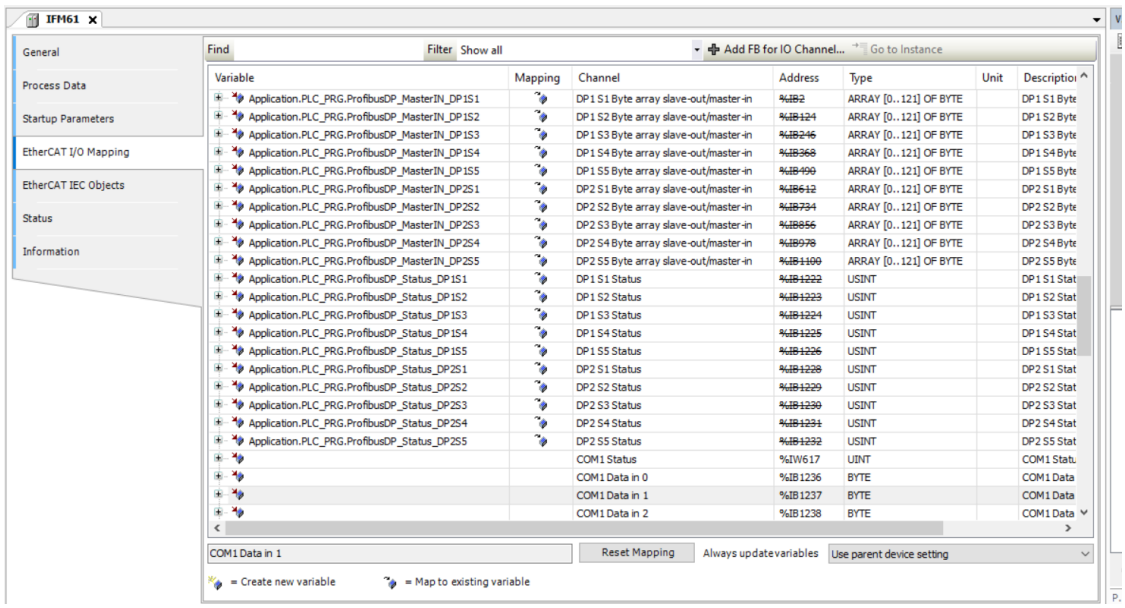
Profibus Inputs									
16#1B02					16#1B03				
16#1A84					16#1AC4				
DP1 S1 Status	DP1 S2 Status	DP1 S3 Status	DP1 S4 Status	DP1 S5 Status	DP2 S1 Status	DP2 S2 Status	DP2 S3 Status	DP2 S4 Status	DP2 S5 Status



程序中声明数据输入输出数组（122 字节）和子站状态变量以后，在“EtherCAT I/O Mapping”页面关联 DP 数据 slave-in/master-out 和 slave-out/master-in 数组以及 DP 子站状态变量。

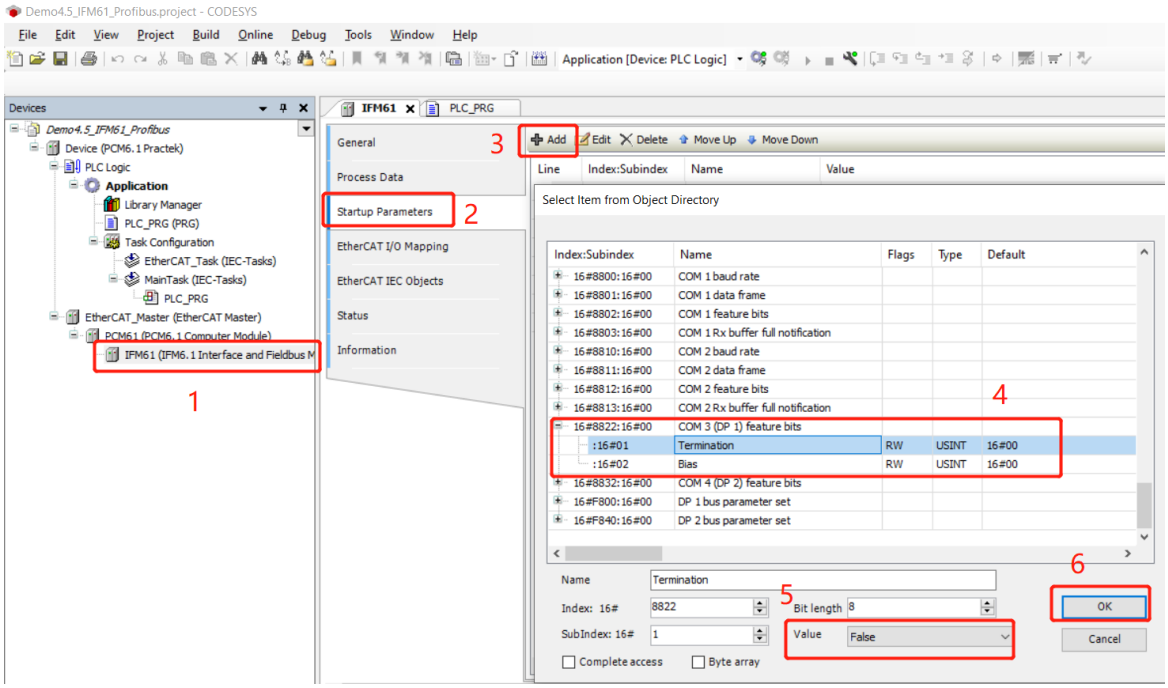

Demo4.5_IFM61_Profibus.project - CODESYS
File Edit View Project Build Online Debug Tools Window Help
Application [Device: PLC Logic]
Devices
Demo4.5_IFM61_Profibus
Device (PCM6.1 Practek)
PLC Logic
Application
Library Manager
PLC_PRG (PRG)
Task Configuration
EtherCAT_Task (IEC-Tasks)
MainTask (IEC-Tasks)
PLC_PRG
EtherCAT_Master (EtherCAT Master)
PCM61 (PCM6.1 Computer Module)
IFM61 (IFM6.1 Interface and Fie
IFM61
PLC_PRG
PROGRAM PLC_PRG
VAR
ProfibusDP_MasterOUT_DP1S1 : ARRAY [0..121] OF BYTE;
ProfibusDP_MasterOUT_DP1S2 : ARRAY [0..121] OF BYTE;
ProfibusDP_MasterOUT_DP1S3 : ARRAY [0..121] OF BYTE;
ProfibusDP_MasterOUT_DP1S4 : ARRAY [0..121] OF BYTE;
ProfibusDP_MasterOUT_DP1S5 : ARRAY [0..121] OF BYTE;
ProfibusDP_MasterIN_DP1S1 : ARRAY [0..121] OF BYTE;
ProfibusDP_MasterIN_DP1S2 : ARRAY [0..121] OF BYTE;
ProfibusDP_MasterIN_DP1S3 : ARRAY [0..121] OF BYTE;
ProfibusDP_MasterIN_DP1S4 : ARRAY [0..121] OF BYTE;
ProfibusDP_MasterIN_DP1S5 : ARRAY [0..121] OF BYTE;
ProfibusDP_Status_DP1S1 : USINT;
ProfibusDP_Status_DP1S2 : USINT;
ProfibusDP_Status_DP1S3 : USINT;
ProfibusDP_Status_DP1S4 : USINT;
ProfibusDP_Status_DP1S5 : USINT;
ProfibusDP_MasterOUT_DP2S1 : ARRAY [0..121] OF BYTE;
ProfibusDP_MasterOUT_DP2S2 : ARRAY [0..121] OF BYTE;
ProfibusDP_MasterOUT_DP2S3 : ARRAY [0..121] OF BYTE;
ProfibusDP_MasterOUT_DP2S4 : ARRAY [0..121] OF BYTE;
ProfibusDP_MasterOUT_DP2S5 : ARRAY [0..121] OF BYTE;
;


IFM61
General
Process Data
Startup Parameters
EtherCAT I/O Mapping
EtherCAT IEC Objects
Status
Information
Find
Filter Show all
Add FB for IO Channel...
Go to Instance
Variable Mapping Channel Address Type
Application.PLC_PRG.ProfibusDP_MasterOUT_DP1S1 DP1 S1 Byte array slave-in/master-out %QB2 ARRAY [0..121] OF BYTE
Application.PLC_PRG.ProfibusDP_MasterOUT_DP1S2 DP1 S2 Byte array slave-in/master-out %QB124 ARRAY [0..121] OF BYTE
Application.PLC_PRG.ProfibusDP_MasterOUT_DP1S3 DP1 S3 Byte array slave-in/master-out %QB246 ARRAY [0..121] OF BYTE
Application.PLC_PRG.ProfibusDP_MasterOUT_DP1S4 DP1 S4 Byte array slave-in/master-out %QB368 ARRAY [0..121] OF BYTE
Application.PLC_PRG.ProfibusDP_MasterOUT_DP1S5 DP1 S5 Byte array slave-in/master-out %QB490 ARRAY [0..121] OF BYTE
Application.PLC_PRG.ProfibusDP_MasterOUT_DP2S1 DP2 S1 Byte array slave-in/master-out %QB612 ARRAY [0..121] OF BYTE
Application.PLC_PRG.ProfibusDP_MasterOUT_DP2S2 DP2 S2 Byte array slave-in/master-out %QB734 ARRAY [0..121] OF BYTE
Application.PLC_PRG.ProfibusDP_MasterOUT_DP2S3 DP2 S3 Byte array slave-in/master-out %QB856 ARRAY [0..121] OF BYTE
Application.PLC_PRG.ProfibusDP_MasterOUT_DP2S4 DP2 S4 Byte array slave-in/master-out %QB978 ARRAY [0..121] OF BYTE
Application.PLC_PRG.ProfibusDP_MasterOUT_DP2S5 DP2 S5 Byte array slave-in/master-out %QB1100 ARRAY [0..121] OF BYTE
COM1 Ctrl %QW611 UINT
COM1 Data out 0 %QB1224 BYTE
COM1 Data out 1 %QB1225 BYTE


IFM61
General
Process Data
Startup Parameters
EtherCAT I/O Mapping
EtherCAT IEC Objects
Status
Information
Find
Filter Show all
Add FB for IO Channel...
Go to Instance
Variable Mapping Channel Address Type Unit Description
Application.PLC_PRG.ProfibusDP_MasterIN_DP1S1 DP1 S1 Byte array slave-out/master-in %IB2 ARRAY [0..121] OF BYTE DP1 S1 Byte
Application.PLC_PRG.ProfibusDP_MasterIN_DP1S2 DP1 S2 Byte array slave-out/master-in %IB124 ARRAY [0..121] OF BYTE DP1 S2 Byte
Application.PLC_PRG.ProfibusDP_MasterIN_DP1S3 DP1 S3 Byte array slave-out/master-in %IB246 ARRAY [0..121] OF BYTE DP1 S3 Byte
Application.PLC_PRG.ProfibusDP_MasterIN_DP1S4 DP1 S4 Byte array slave-out/master-in %IB368 ARRAY [0..121] OF BYTE DP1 S4 Byte
Application.PLC_PRG.ProfibusDP_MasterIN_DP1S5 DP1 S5 Byte array slave-out/master-in %IB490 ARRAY [0..121] OF BYTE DP1 S5 Byte
Application.PLC_PRG.ProfibusDP_MasterIN_DP2S1 DP2 S1 Byte array slave-out/master-in %IB612 ARRAY [0..121] OF BYTE DP2 S1 Byte
Application.PLC_PRG.ProfibusDP_MasterIN_DP2S2 DP2 S2 Byte array slave-out/master-in %IB734 ARRAY [0..121] OF BYTE DP2 S2 Byte
Application.PLC_PRG.ProfibusDP_MasterIN_DP2S3 DP2 S3 Byte array slave-out/master-in %IB856 ARRAY [0..121] OF BYTE DP2 S3 Byte
Application.PLC_PRG.ProfibusDP_MasterIN_DP2S4 DP2 S4 Byte array slave-out/master-in %IB978 ARRAY [0..121] OF BYTE DP2 S4 Byte
Application.PLC_PRG.ProfibusDP_MasterIN_DP2S5 DP2 S5 Byte array slave-out/master-in %IB1100 ARRAY [0..121] OF BYTE DP2 S5 Byte
Application.PLC_PRG.ProfibusDP_Status_DP1S1 DP1 S1 Status %IB1222 USINT DP1 S1 Stat
Application.PLC_PRG.ProfibusDP_Status_DP1S2 DP1 S2 Status %IB1223 USINT DP1 S2 Stat
Application.PLC_PRG.ProfibusDP_Status_DP1S3 DP1 S3 Status %IB1224 USINT DP1 S3 Stat
Application.PLC_PRG.ProfibusDP_Status_DP1S4 DP1 S4 Status %IB1225 USINT DP1 S4 Stat
Application.PLC_PRG.ProfibusDP_Status_DP1S5 DP1 S5 Status %IB1226 USINT DP1 S5 Stat
Application.PLC_PRG.ProfibusDP_Status_DP2S1 DP2 S1 Status %IB1228 USINT DP2 S1 Stat
Application.PLC_PRG.ProfibusDP_Status_DP2S2 DP2 S2 Status %IB1229 USINT DP2 S2 Stat
Application.PLC_PRG.ProfibusDP_Status_DP2S3 DP2 S3 Status %IB1230 USINT DP2 S3 Stat
Application.PLC_PRG.ProfibusDP_Status_DP2S4 DP2 S4 Status %IB1231 USINT DP2 S4 Stat
Application.PLC_PRG.ProfibusDP_Status_DP2S5 DP2 S5 Status %IB1232 USINT DP2 S5 Stat
COM1 Status %IW617 UINT COM1 Statu
COM1 Data in 0 %IB1236 BYTE COM1 Data
COM1 Data in 1 %IB1237 BYTE COM1 Data
COM1 Data in 2 %IB1238 BYTE COM1 Data
COM1 Data in 1
Reset Mapping
Always update variables
Use parent device setting
= Create new variable
= Map to existing variable

IFM6.1 Profibus DP Master 端口需要设置 8 个启动参数，分别为 Termination、Bias、Master address、Data rate、Station address、Ident number、PRM data、CFG data。Profibus DP1 端口启动参数设置如下, Profibus DP2 端口设置方法相同。

点击设备“IFM61”、“Startup Parameters”、“Add”开始添加启动参数，弹出对话框中设置相应的启动参数。每次点击“Add”新建启动参数只能添加一个参数，添加多个启动参数则需要多次点击“Add”新建。



- 16#8822:16#00 COM3(DP1) feature bits 设置：

  分项 16#01 Termination，Value 设置为 True。

  分项 16#02 Bias，Value 设置为 True。

- 16#F800:16#00 DP1 bus parameter set 设置：

  分项 16#01 Master address，Value 设置为 1 且不同于子站站号。

  分项 16#02 Data rate，Value 按实际波特率设置。

- 16#8000:16#00 DP1 communication parameter slave 1 设置：

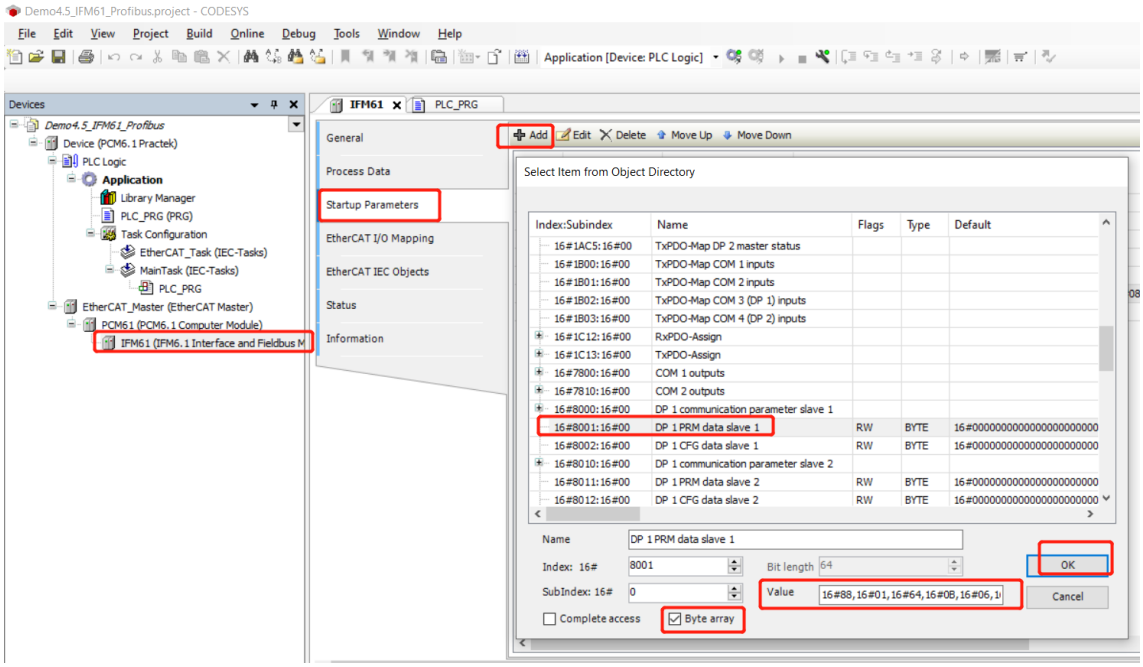
  分项 16#01 Station address，Value 按实际子站站号设置。

分项 16#04 Ident number，Value 按实际子站 GSD 参数设置。

- 16#8001:16#00 DP1 PRM data slave 1 设置：

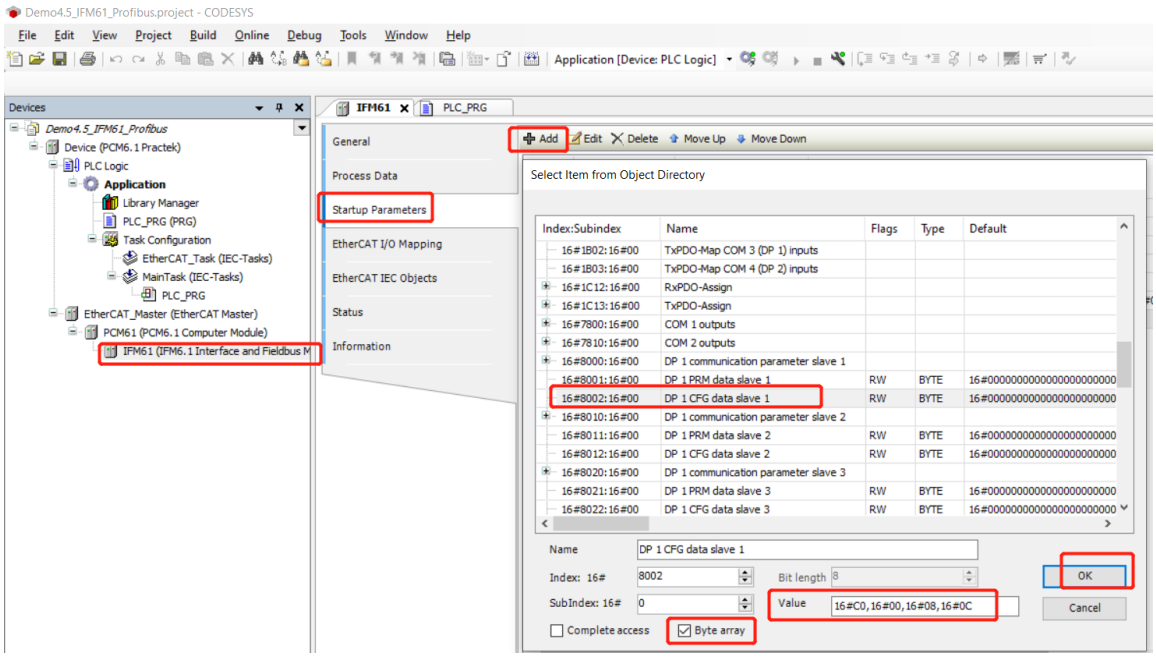
PRM data 需要根据实际连接的子站配置信息确定，比如子站 Ident number 为 0x06FC，Own PrmData 为 C0 00 08 0C。则勾选“Byte array”后填入 PRM data 为 16#88,16#01,16#64,16#0B,16#06,16#FC,16#00,16#C0,16#00,16#08,16#0C。

88 01 64 0B	06 FC	00	C0 00 80 0C
推荐参数	Ident number	固定格式	Own PrmData

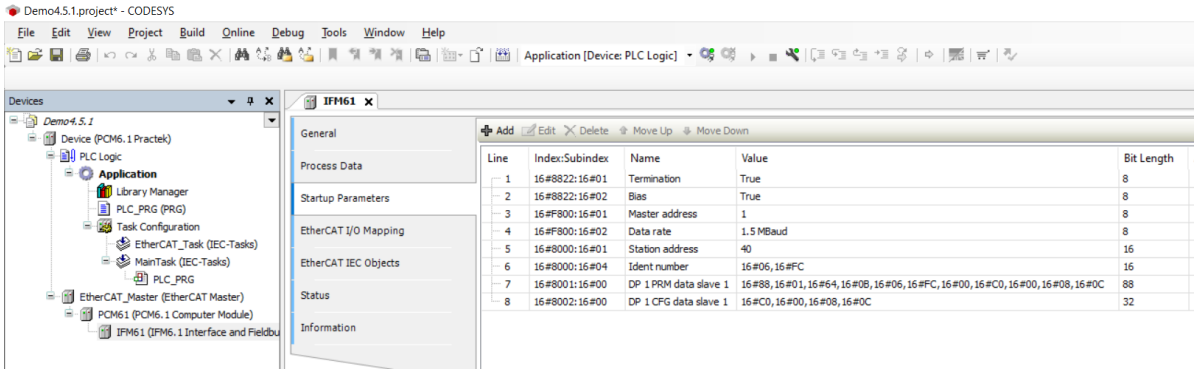


- 16#8002:16#00 DP1 CFG data slave 1 设置：

CFG data 需要根据实际连接的子站配置信息确定，比如子站 Own PrmData 为 C0 00 08 0C。则勾选“Byte array”后填入 CFG data 为 16#C0,16#00,16#08,16#0C。



IFM6.1 Profibus DP 启动参数设置参考例程 Demo4.5_IFM61_Profibus，DP1 端口启动参数设置完成如下所示，DP2 启动参数设置方法相同。



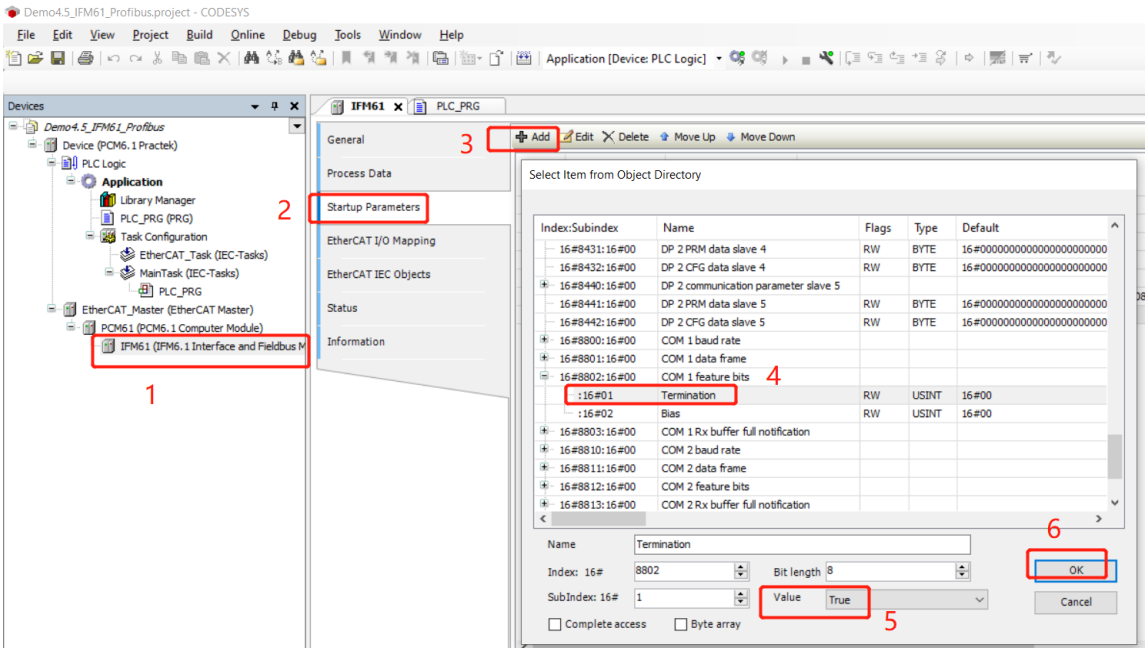
4.5.2 串口程序配置

IFM6.1 通信模块具有 2 个 RS-485 端口，支持 Modbus-RTU 通信，支持 IFM6.1 作为主站与外接传感器或其他设备通讯。

IFM6.1 COM 端口 Modbus-RTU 需要设置 4 个启动参数，分别为 Termination、Bias、Baud rate、Frame format。IFM6.1 COM1 Modbus-RTU 通信启动参数设置如下，COM2 端口设置方法相同。

点击设备“IFM61”、“Startup Parameters”、“Add”开始添加启动参数，弹出对

话框中设置相应的启动参数。每次点击“Add”新建启动参数只能添加一个参数，添加多个启动参数则需要多次点击“Add”新建。



- 16#8802:16#00 COM1 feature bits 设置：

  分项 16#01 Termination，Value 设置为 True。

  分项 16#02 Bias，Value 设置为 True。

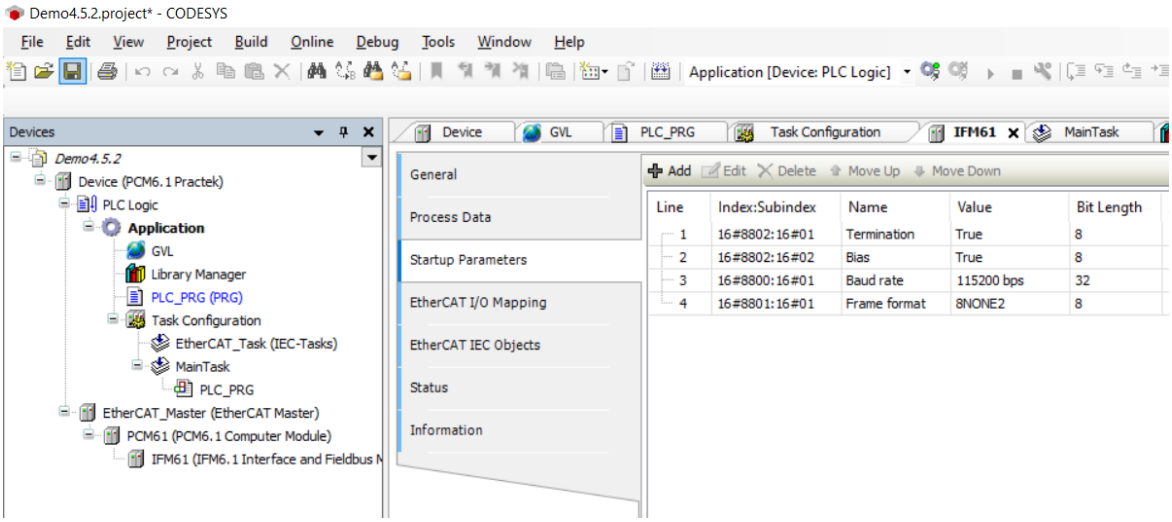
- 16#8800:16#00 COM1 baud rate 设置：

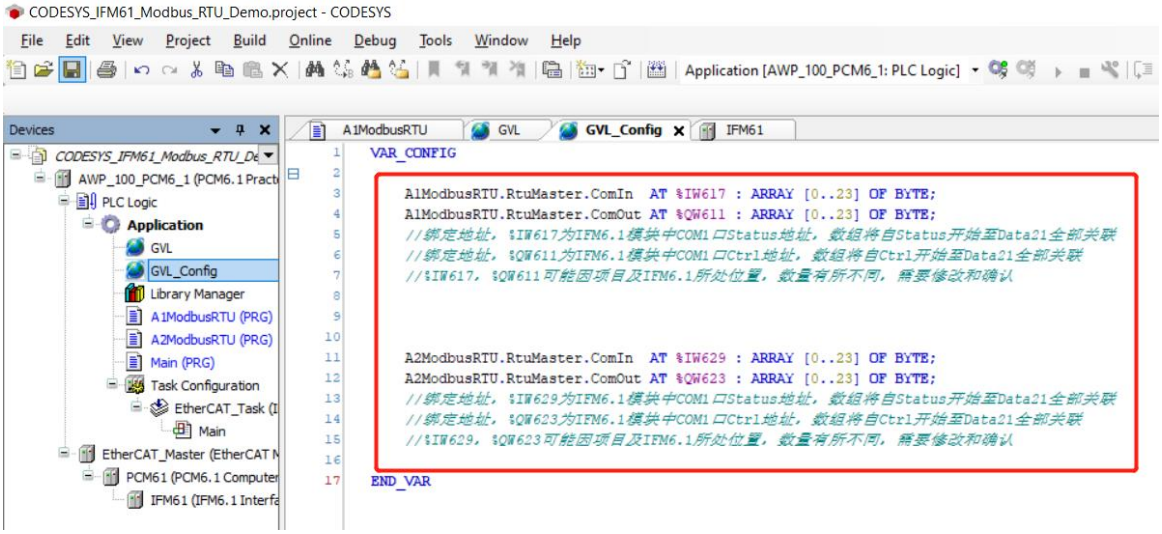
  分项 16#01 Baud rate，Value 按需设置波特率。

- 16#8801:16#00 COM1 data frame 设置：

  分项 16#01 Frame format，Value 按需设置报文格式。

IFM6.1 COM1 端口 Modbus-RTU 启动参数设置完成，硬件通道变量链接需要使用“AT”COM 端口地址方式，不可以使用常用的“Mapping”。输入数组首位绑定 COM 端口 Status 地址，输出数组首位绑定 COM 端口 Ctrl 地址。绑定地址可能跟随 IFM6.1 所处位置变化而发生变化，需要注意核对地址是否正确。参考例程 Demo4.5_IFM61_ModbusRTU。





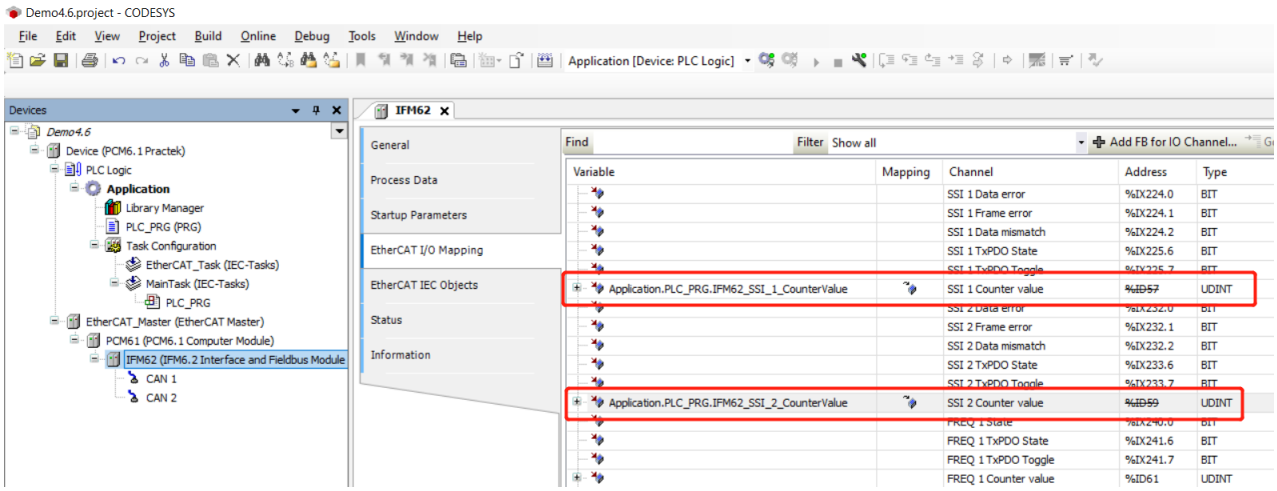
4.6 IFM6.2 程序配置

4.6.1 CANopen 程序配置

IFM6.2 通信模块具有 2 个 CAN 端口，CANopen 程序配置方法与 PCM6.1 基本相同，此处不再赘述。

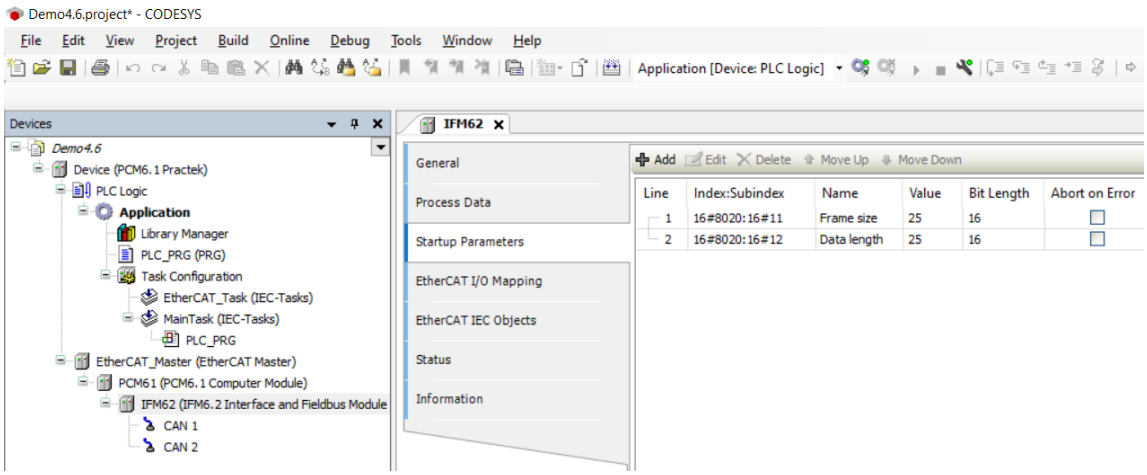
4.6.2 SSI 程序配置

IFM6.2 通信模块具有 2 个 SSI 端口，在“EtherCAT I/O Mapping”中将变量关联到硬件通道上。



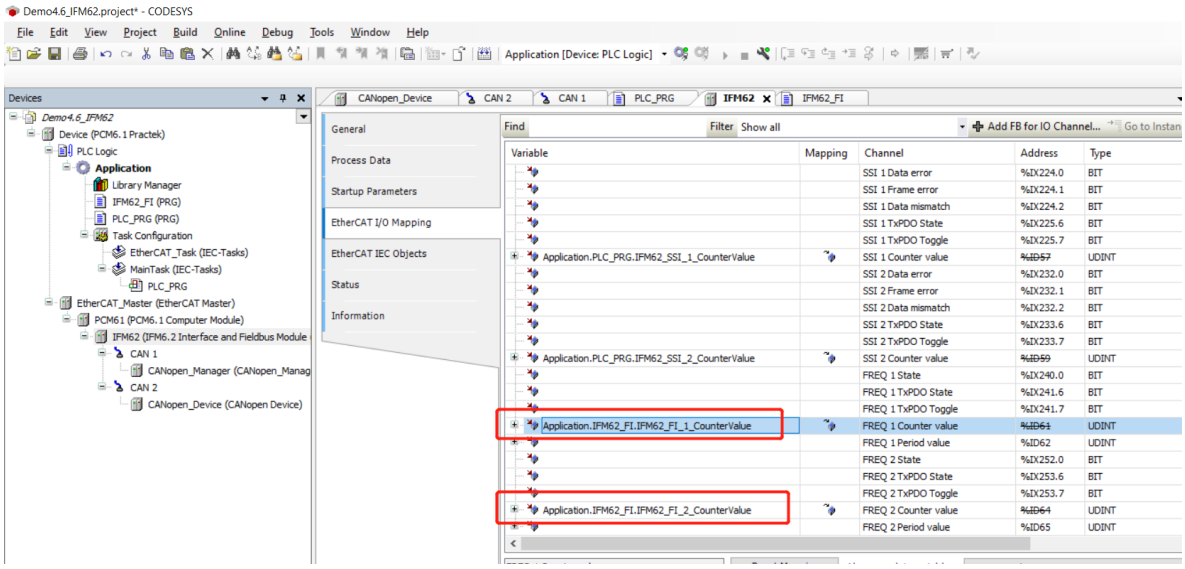
点击设备“IFM62”、“Startup Parameters”、“Add”新建启动参数，在弹出对话框内选择需要配置的参数进行配置。每次点击“Add”新建启动参数只能添加一个参数，添加多个启动参数则需要多次点击“Add”新建。

SSI 端口需要配置 2 个启动参数，分别为 Frame size 和 Data length。例如编码器 Baumer GM400.Z103, 8192 × 4096, 25 bit Gray code，由于此编码器是 25 位格雷码则 Frame size 设置为 25，由于此编码器 8192 × 4096 则 Data length 设置为 25。参考例程 Demo4.6_IFM62。



4.6.3 FI 程序配置

IFM6.2 通信模块具有 2 个 FI（Digital Frequency Input）端口，在“EtherCAT I/O Mapping”中将变量关联到硬件通道上。Counter Value 将在每个输入脉冲的上升沿递增，两个脉冲周期间隔时间为 10 ns。参考例程 Demo4.6_IFM62。

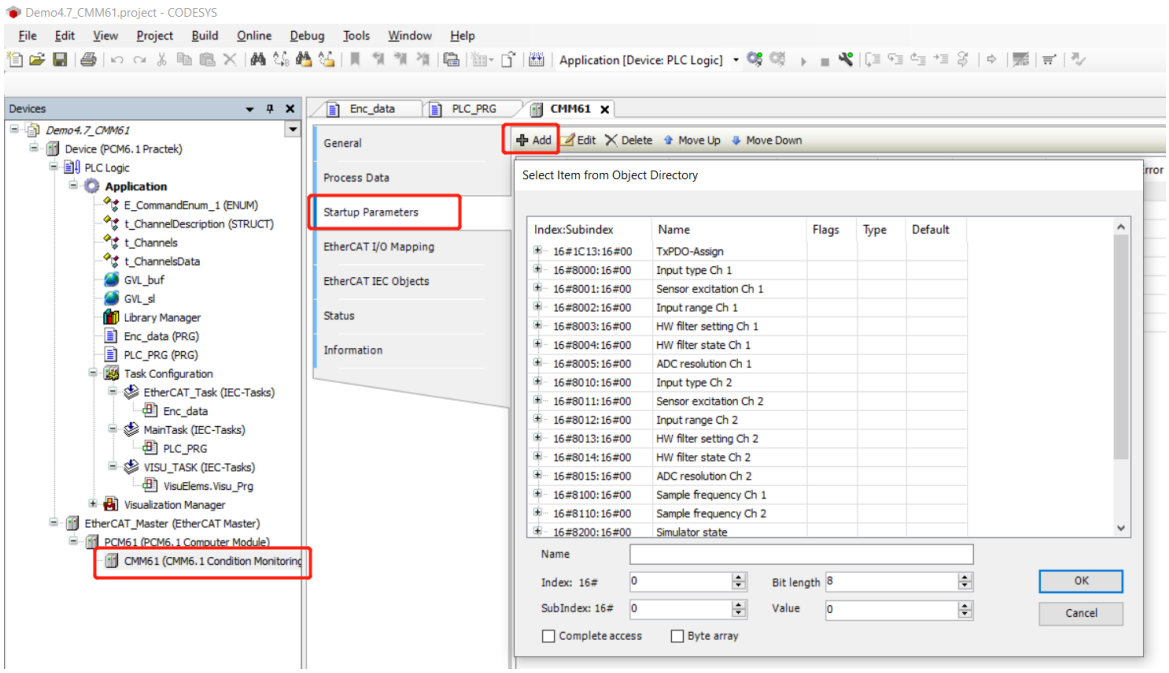


4.7 CMM6.1 程序配置

CMM6.1 为高频模拟量输入模块，具有 2 个端口，通常用来对机械部件振动情况进行超采样，对采样数据进行频谱分析。

CMM6.1 需要设置 4 个启动参数，分别为 Input type、Sensor excitation、Input range、Sample frequency。下面以通道 Ch 1 设置启动参数举例，通道 Ch 2 设置启动参数方法相同。参考例程 Demo4.7_CMM61。

点击设备“CMM61”、“Startup Parameters”、“Add”新建启动参数，在弹出对话框内选择需要配置的参数进行配置。每次点击“Add”新建启动参数只能添加一个参数，添加多个启动参数则需要多次点击“Add”新建。



- 16#8000:16#00 Input type Ch 1 设置：

分项 16#01 Input type，Value 设置为 AC Mode，外接传感器输入交流信号。

- 16#8001:16#00 Sensor excitation Ch 1 设置：

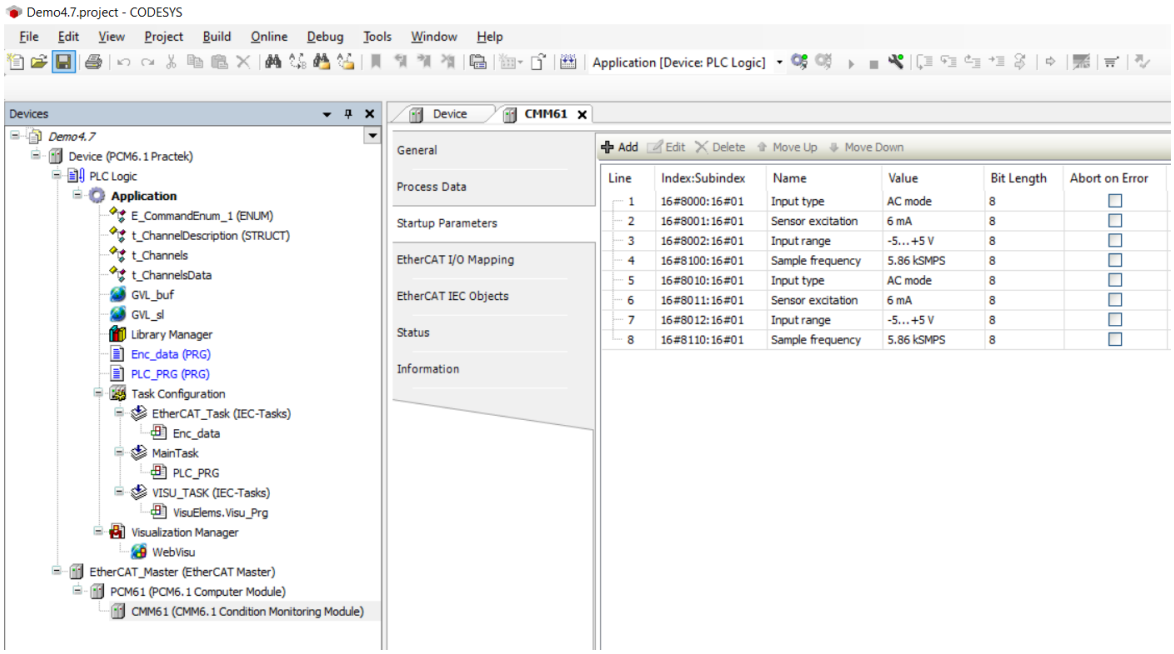
分项 16#01 Sensor excitation，Value 按需设置传感器响应电流。

- 16#8002:16#00 Input range Ch 1 设置：

分项 16#01 Input range，Value 按需设置测量范围。

- 16#8100:16#00 Sample frequency Ch 1 设置：

分项 16#01 Sample frequency，Value 按需设置采样频谱。



CMM6.1 硬件通道链接只能通过代码定义“AT” CMM61 数据地址，不可以使用 CMM61 中“Mapping”。