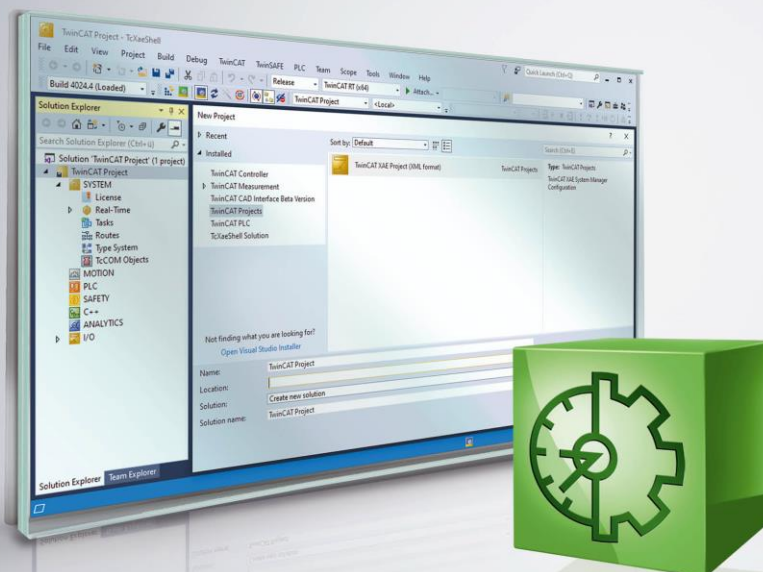


BECKHOFF New Automation Technology

TE1400

TwinCAT3 | Target for Simulink



中文版本 Ver 1.0

前言

通过 TwinCAT 3 Target for Simulink 的封装，Simulink 中开发的模型可以在 TwinCAT 3 平台上使用。诸如 SimScape、Stateflow、DSP System Toolbox 之类的工具箱都可以集成在 Simulink 模型中，这些模型使用 Simulink Coder 自动转换为 C/C++ 代码，并通过 TwinCAT 3 Target for Simulink 转换为 TwinCAT 对象，极大地拓展了 TwinCAT 的应用场景。

由 Simulink 创建的 TwinCAT 对象具有与其它 TwinCAT 对象相同的接口和属性。它们可以在 TwinCAT 3 开发环境中得到充分应用。例如，可以将这些对象添加到工程项目中连同 PLC 代码以及现场总线模块组成一个可以调试的完整项目。在 TwinCAT 开发环境中可以看到完整的 Simulink 框图。除了支持 Simulink 外部模式外，嵌入在开发环境中的模型框图也可以用作参数调整、调试、信号和状态监测的控件。自动生成的模块既可以作为 TcCOM 对象，也可以作为 PLC 功能块集成到 TwinCAT 解决方案中。

TwinCAT 对象可以被分配给实时环境中不同的 CPU 核。这就意味着，即便是大型项目，也可以很容易地对其进行扩展，例如仿真整个风力发电场。如果个别对象对运算速度有要求，也可以将代码放在多个核上并行处理。这样，只需一个工具即可全面支持从简单的控制器到完整的设备控制和实时仿真等所有的应用。

TwinCAT 3 Target for Simulink 功能推出以后深受广大倍福客户的青睐，尤其在涉及到复杂控制算法和模型的应用场景中。2022 年以后倍福对其功能做了升级，从 V1.2 版本升级到 V2.3 版本。这其中增加了很多功能，操作的方法也与之前版本有较大的区别。因此，我们编辑了这个文档，将 V2.3 版本的相关操作和功能分享给大家，希望能够帮助大家在使用相关产品的过程中更加简便快捷。

本文档主要针对使用 TE1400 封装 Simulink 模型的客户，为其提供一个基本的比较全面的入门手册。后续随着软件功能的更新，本文档内容会有增删。

由于个人水平有限，文档内容难免会有纰漏或不妥之处，欢迎大家提出宝贵意见和建议。

目录

| | |
|--|----|
| 第一章 安装 | 5 |
| 一、系统要求..... | 5 |
| 二、在编程 PC 上..... | 5 |
| 三、在运行时 PC 上..... | 5 |
| 四、安装步骤..... | 5 |
| 五、授权..... | 8 |
| 第二章 设置驱动程序签名 | 8 |
| 一、简言之，我们最需要关注如下几点：..... | 8 |
| 二、如何使用 Level2 等级的 OEM 证书对驱动程序签名呢？..... | 9 |
| 三、TwinCAT 运行时中需要做什么操作呢？..... | 12 |
| 四、如何得到 TwinCAT user certificate 的会签签名？..... | 13 |
| 五、OEM 证书请求文件的创建步骤..... | 13 |
| 六、确定 OEM 证书请求文件的指纹码..... | 15 |
| 第三章 快速入门操作 | 15 |
| 一、打开一个简单的 Simulink 模型..... | 16 |
| 二、在 Simulink 中的配置步骤..... | 17 |
| 三、在 TwinCAT 中的操作..... | 17 |
| 四、TcCOM 组件的参数设置及链接..... | 19 |
| 第四章 Simulink 中的 TwinCAT 库 | 20 |
| 第五章 将 Simulink 模型直接封装成 PLC 库的操作 | 22 |
| 一、创建并安装 PLC 库的步骤及参数设置..... | 22 |
| 二、TcCOM Wrapper FB 的应用..... | 25 |

三. PLC-FB 功能块的应用28

第六章 TcCOM 组件在线更新操作29

第一章 安装

一、系统要求

在使用 TE1400 对 Simulink 模型进行封装的过程中，涉及到编程 PC 和运行时 PC 两个不同的硬件平台，下面将对两者进行区别描述。在编程 PC 上，Simulink 模型使用 TwinCAT Target for Simulink (TE1400) 转换为 TwinCAT 对象。TwinCAT 对象一旦转换完成就可以在 TwinCAT 解决方案中使用了，编程 PC 将 TwinCAT 解决方案下载到 TwinCAT 运行时环境中运行，以执行项目代码。

二、在编程 PC 上

MATLAB R2019a 或者以上版本。

Visual Studio 2017 或者以上版本(专业版、旗舰版或者等价版本)。注意：在安装时必须勾选 Desktop development with C++。

TwinCAT 3.1.4024.7 或者以上版本。

确保 Visual Studio 勾选 Desktop development with C++ 并安装完成后再安装 TwinCAT3。

三、在运行时 PC 上

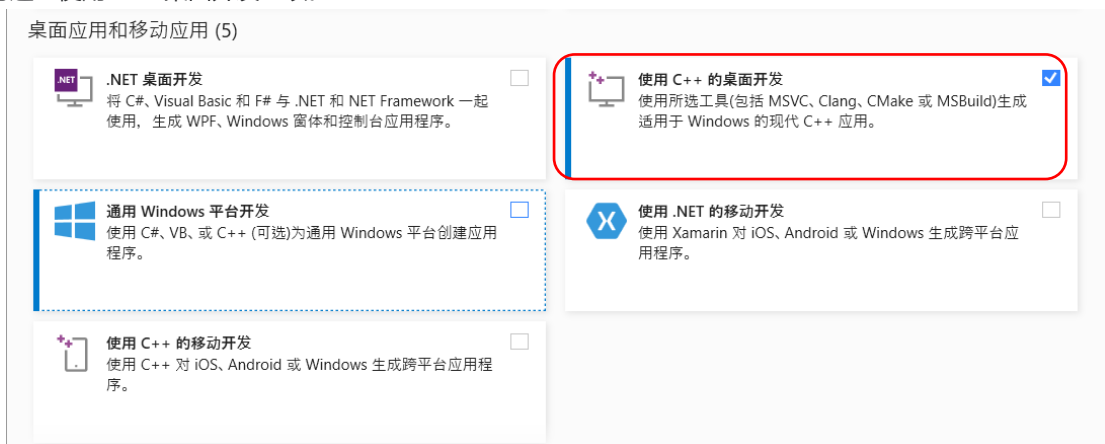
支持的操作系统有：Windows 7, Windows 10, Windows Server (32-bit and 64-bit), TwinCAT/BSD。

TwinCAT XAR 版本为 3.1.4024.7 或者更高。

四、安装步骤

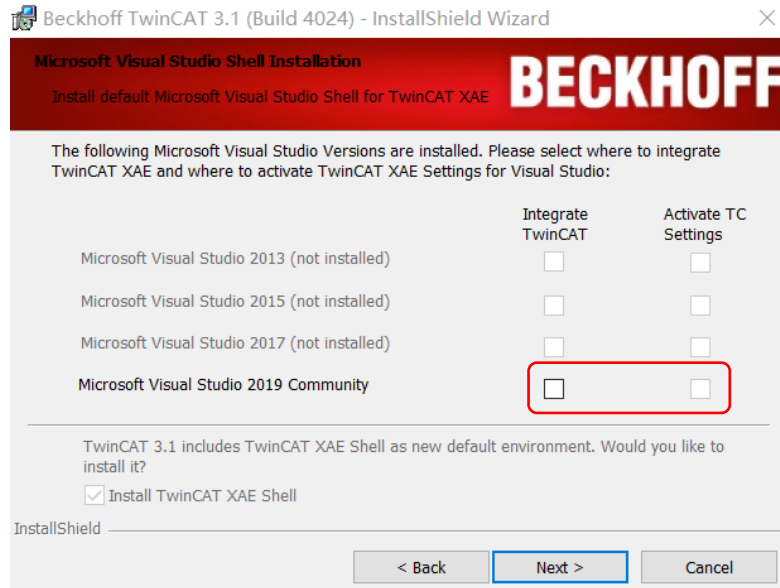
首先确保先安装 Visual Studio，并且在安装过程中勾选 Desktop development with C++ 项目。如果您已经安装过 Visual Studio，但是没有勾选相关项，请先安装好后再进行下一步操作。

1. 安装合适的 Visual Studio 版本，在本步骤中我们选择安装 Visual Studio 2019 Community，在安装过程中注意勾选“使用 C++ 桌面开发”项。



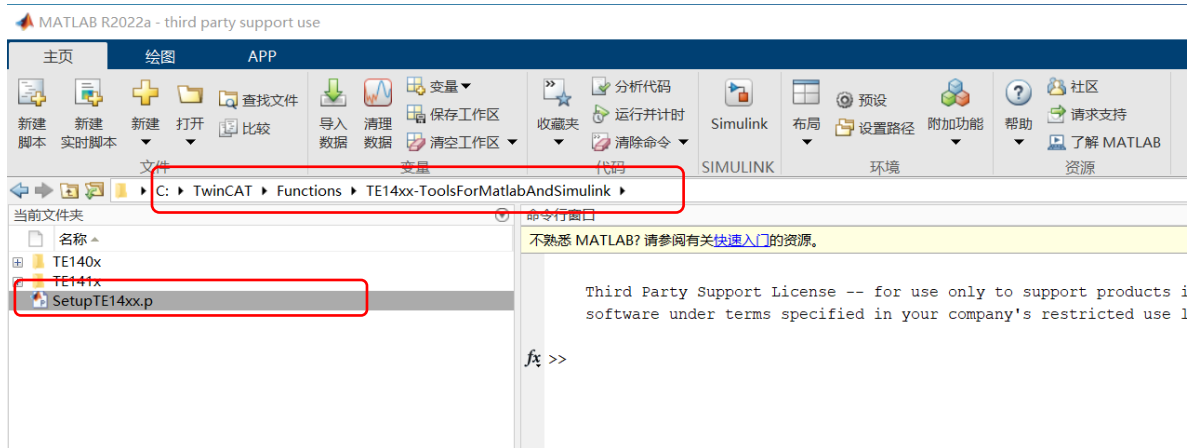
图一 勾选 C++ 桌面开发

2. 随后安装 TwinCAT3 Full 版本，和单独安装 TwinCAT3 Full 版本不同的是，在安装的过程中软件会自动检测当前已经安装的 Visual Studio 版本，勾选下图红色框图圈起的项，点击“Next”安装即可。



图二 嵌入到安装好的 VS 框架中

3. 安装 MATLAB 软件，它的安装次序无关紧要。
4. 安装 TE1400 软件包。TE1400 安装完成后放置在 C:\TwinCAT\Functions 路径下，因此，它的安装是独立于 MATLAB 软件的。
5. 以管理员身份打开 MATLAB 软件，在 MATLAB 软件中找到 TwinCAT 安装目录下 SetupTE14xx.p 文件，点击右键运行，如下图所示：



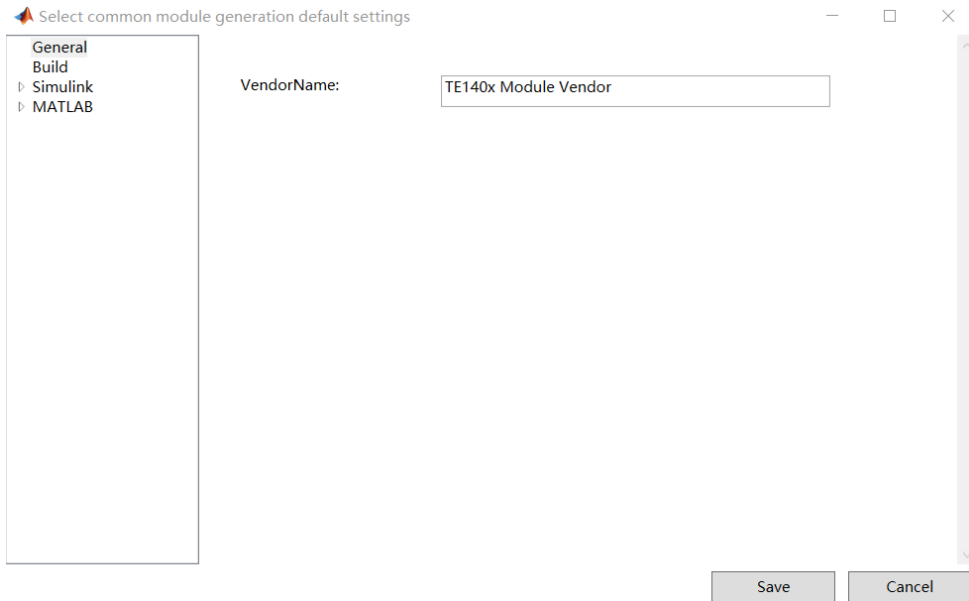
图三 在 MATLAB 软件中运行 TE14XX.P 文件

这时会出现一个关于 TE1400 的对话框，在这个对话框中可以设置若干参数。

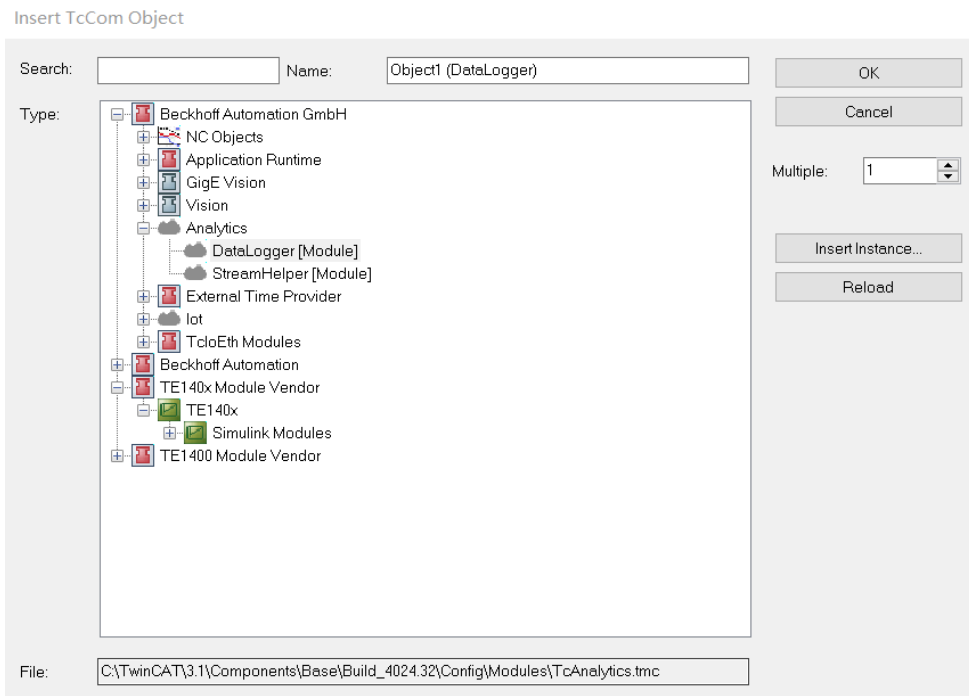
- VendorName
- GroupName (MATLAB) and
- GroupName (Simulink)

这几个参数的设置内容会影响到调用 TcCOM 组件时 TwinCAT 中的层级结构。

- VendorName "TE140x Module Vendor"
- GroupName "TE140x\MATLAB Modules" for MATLAB and
- "TE140x\Simulink Modules" for Simulink.

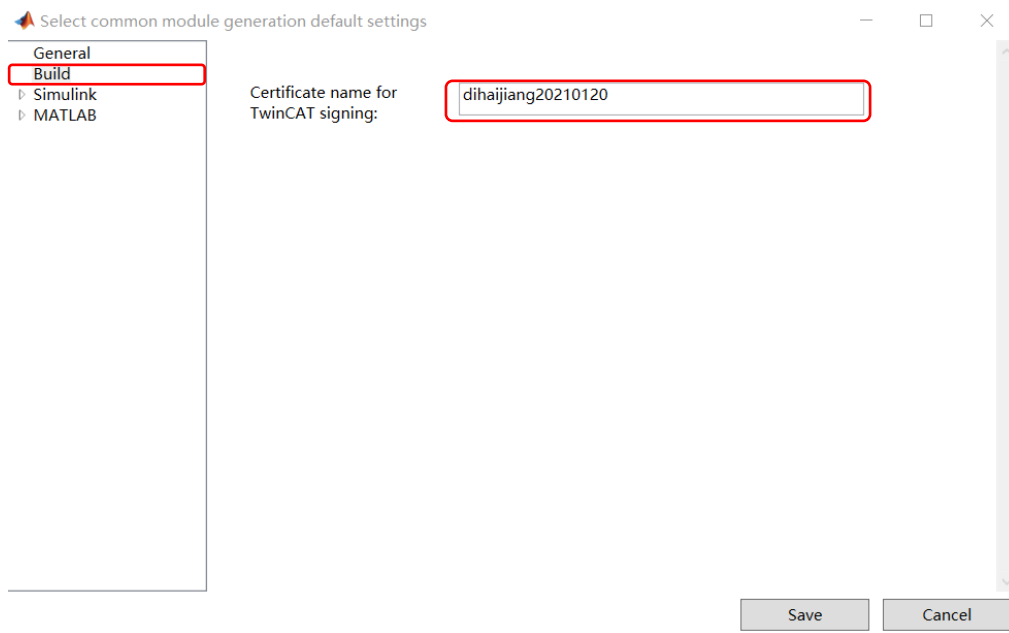


图四 打开配置对话框



图五 配置参数直接影响调用时的层级结构名称

在 Build 选项卡下，您可以设置用来做驱动程序签名的证书。创建和使用驱动程序签名证书的过程，在后面的章节中会重点介绍。



图六 设置用来封装的证书名称

五、授权

在使用 TwinCAT Target for Simulink 功能时需要关注两个授权。首先，在编程 PC 上需要 TE1400 授权对 Simulink 模型进行封装操作，一般我们选择将此授权绑定到 Dongle 中使用。其次，在运行时 PC 上需要激活能够支持 Simulink 语言的授权，例如：TC1220 或者 TC1320。

在不购买永久授权的情况下，上述三个授权均可通过激活临时授权的方式进行试用。需要注意的是 TE1400 授权如果激活的是试用版授权，则封装的 Simulink 模型最大不能超过 100 个功能模块，并且输入不超过 5 个，输出也不超过 5 个。TC1220 和 TC1320 如果激活的是临时授权，则功能上没有任何限制，只是试用版的临时授权每激活一次只能运行 7 天。

第二章 设置驱动程序签名

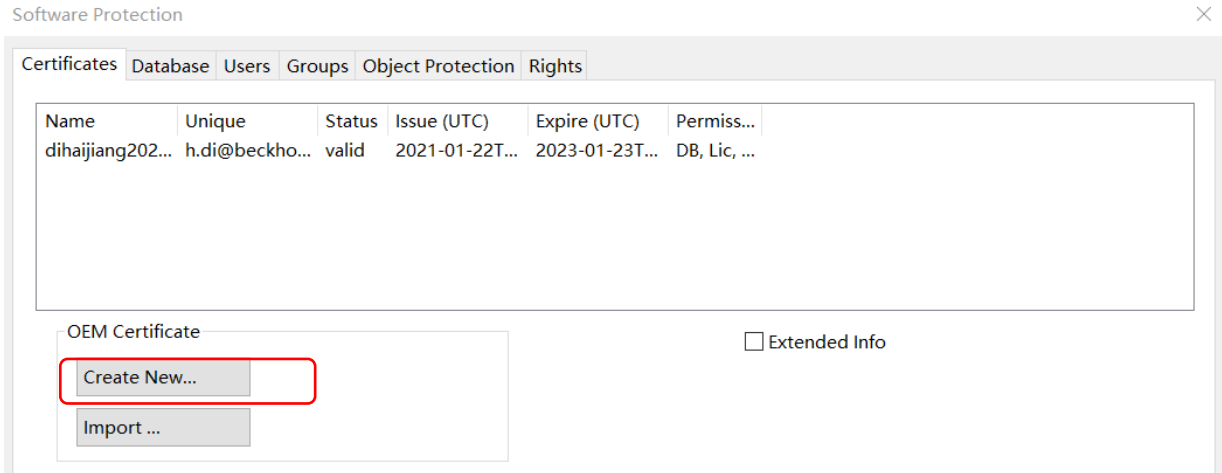
到现在为止，我们已经安装好了相关的软件，也激活了相关的授权，是否就可以进行 Simulink 模型的封装了呢？在新版本的 TwinCAT Target for Simulink 中，从 Simulink 中封装生成的 TwinCAT 对象（如：TcCOM）和 TwinCAT C++ 对象一样是基于一个叫做 tmx（TwinCAT Module Executable）的驱动的。而这种驱动程序必须使用 OEM certificate level 2 证书进行签名才能下载到 TwinCAT 运行时中执行。

那么我们如何才能创建一个 OEM 证书来对驱动程序进行签名呢？请参考下面两个链接相关的文档说明。

- [General documentation on OEM certificates](#)
- [Application-related documentation for tmx driver signing](#)

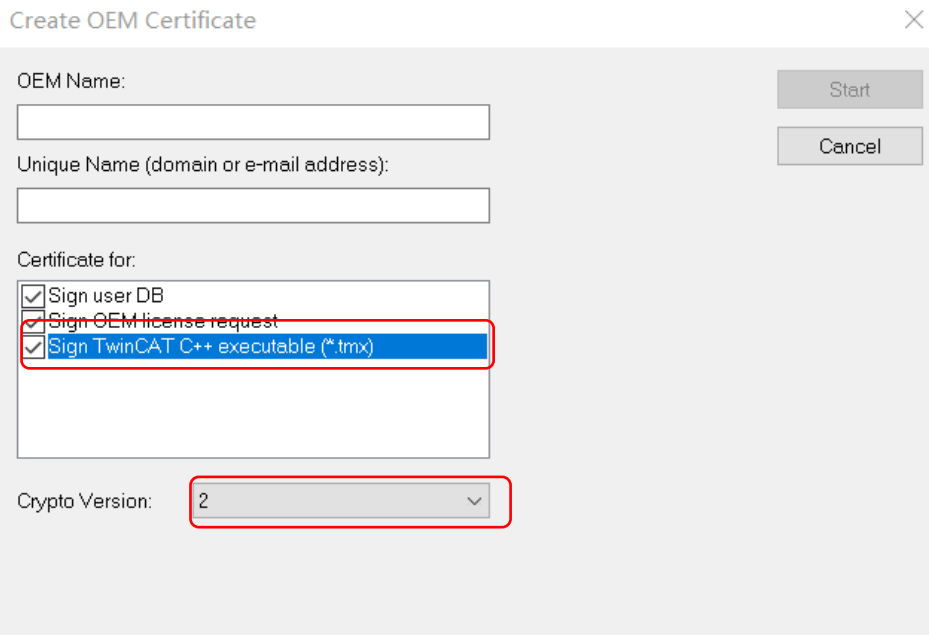
一、简言之，我们最需要关注如下几点：

第一：在 Visual Studio 菜单栏中选择“扩展”->TwinCAT->Software Protection...即可打开一个对话框，在这个对话框中可以创建自己的 OEM 证书。



图七 创建一个新的 OEM 证书

第二：创建 OEM 证书时请务必勾选 “Sign TwinCAT C++ executables” 项，确保 Crypto Version 设置项为 2。



图八 必须勾选的项目

第三：32 位操作系统和 64 位操作系统中的驱动程序都必须签名才能使用。

第四：驱动程序可以在没有签名的情况下创建，以后再签名。

第五：在开发阶段出于测试的目的，客户可以使用没有会签的证书（未经德国授权）对程序进行签名。

第六：如果客户希望得到会签后的证书，则可以向 Beckhoff 订购 TC0008。

二、如何使用 Level2 等级的 OEM 证书对驱动程序签名呢？

对 TMX 驱动程序进行签名有四种不同的方式，这四种方式请任选一种使用即可。

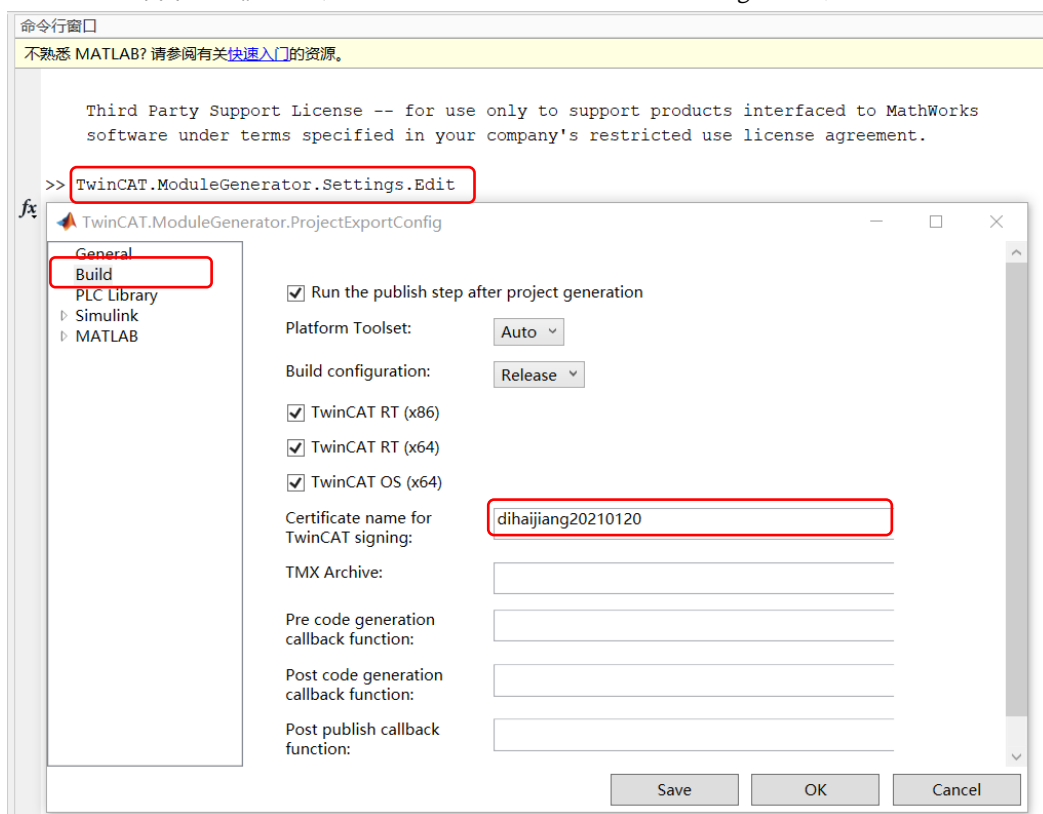
1. 在编程 PC 上为 TwinCAT C++、MATLAB 函数和 Simulink 模型的封装设置一个默认的证书。



图九 添加新的系统变量

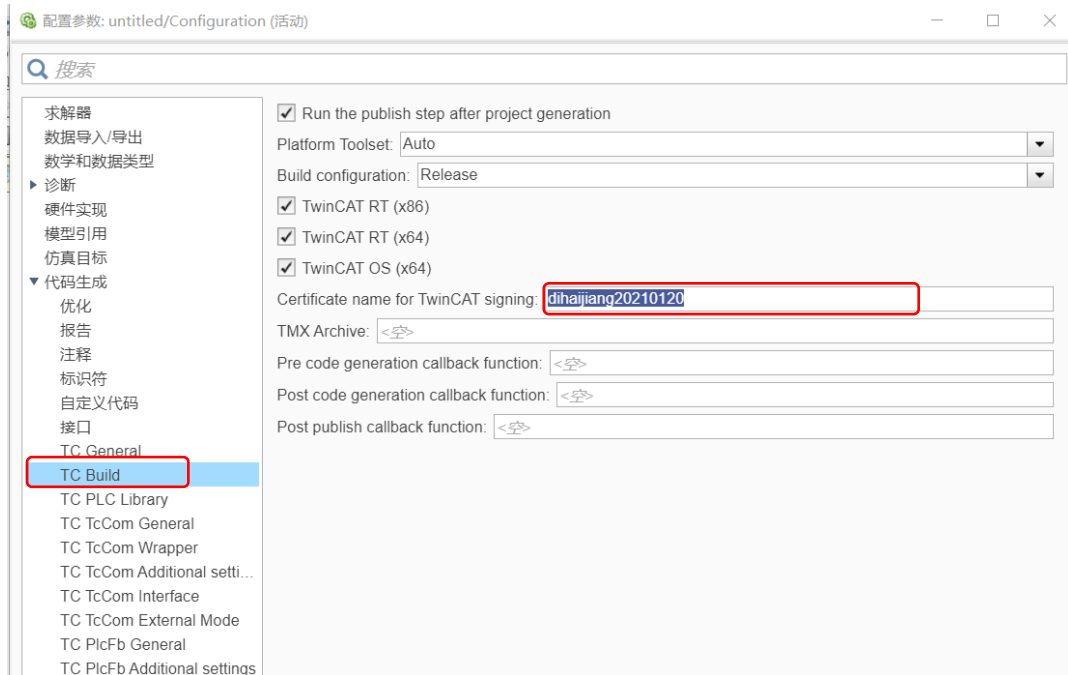
OEM 证书的放置路径为: *TwinCAT\3.1\CustomConfig\Certificates*

2. 在编程 PC 上为 MATLAB 函数和 Simulink 模型的封装设置一个默认证书，具体设置内容请参考图六。对话框可以在 MATLAB 工作台中通过输入指令: *TwinCAT.ModuleGenerator.Settings.Edit* 调出。



图十 设置封装模型的证书

3. 事先并不做任何设定，直到模型封装时再指定 OEM 证书。



图十一 在封装 Simulink 模型时才指定 OEM 证书

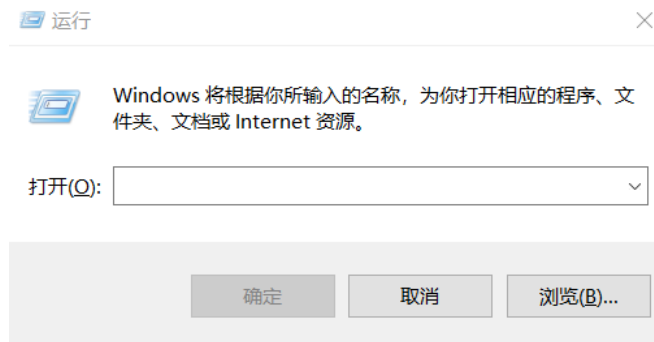
4. 在不使用证书的情况下进行封装，使用 TcSignTool 工具进行签名。TcSignTool 是一个命令行程序，它存放在 `C:\TwinCAT\3.x\sdk\Bin` 这个路径下。使用 `tcsigntool /?` 或者 `tcsigntool sign /?` 命令可以得到更多关于这个工具的帮助信息。

```
TcSignTool sign /f "C:\TwinCAT\3.1\CustomConfig\Certificates\ MyCertificate.tccert" /p MyPassword
"C:\TwinCAT\3.1\Repository\TE140x Module Vendor\ModulName\0.0.0.1\TwinCAT RT (x64)\MyDriver.tmx"
```

对于第一种、第二种和第三种方式，必须使用 TcSignTool 工具将指定证书的密码存储在系统中。出于安全的考虑，密码不应该在 Simulink 模型的源代码或 MATLAB 代码中出现。使用 TcSignTool 工具，可以在 Windows 操作系统的注册表中加密存储属于您的证书的密码。

下面我们以存储密码为例，演示如何使用 TcSignTool 工具。

- a. 按下 Win+R 键，调出运行对话框。



图十二 调出运行对话框

- b. 在运行对话框中输入 `cmd`，然后回车。



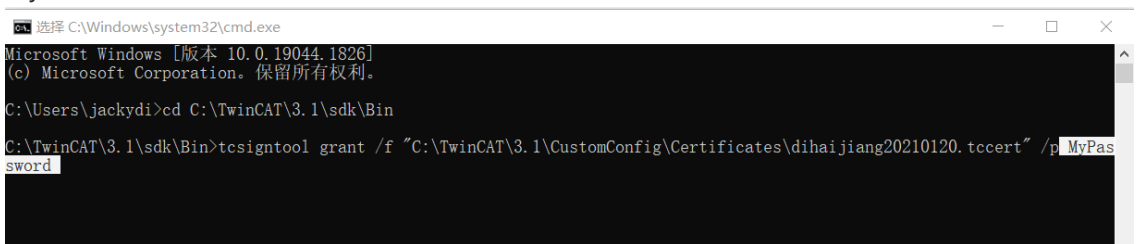
图十三 cmd 指令打开对话框

- c. 输入指令 `cd C:\TwinCAT\3.1\sdk\Bin`，回车。



图十四 修改路径

- d. 输入指令 `tcsigntool grant /f "C:\TwinCAT\3.1\CustomConfig\Certificates\dihaijiang20210120.tccert" /p MyPassword`，然后回车即可。



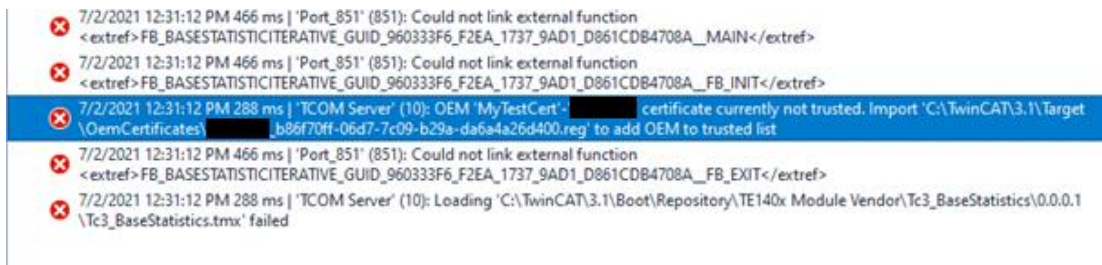
图十五 通过指令存储 OEM 证书的密码

这里注意指令内容中 `dihaijiang20210120` 部分请用您自己的证书名称替换，`MyPassword` 部分需要填写您申请 OEM 证书时设定的密码。

三、TwinCAT 运行时中需要做什么操作呢？

经过 OEM 证书签名且正常完成封装的 Simulink 模型，通过 TwinCAT3 软件激活到运行时中执行时，必须遵守如下规定：

每个 TwinCAT 运行时(XAR)都有自己的受信任证书白名单。如果用于签名的证书不包含在此白名单中，则不会加载驱动程序。这时将输出相应的错误消息：



图十六 激活到运行时报错

根据提示信息，在运行时 PC 的 C:\TwinCAT\3.1\Target\OemCertificates 这个路径下可以找到相关的注册表文件，右键单击，以管理员身份运行，重启运行时 PC 即可。

如果在驱动程序签名时使用的是倍福没有会签过的 OEM 证书，那么运行时 PC 需要将测试模式打开，请先在 BIOS 中确认“SecureBoot”是关闭状态，然后在运行时 PC 即目标系统上以管理员身份运行命令：bcdedit /set testsigning yes，最后重启 PC 即可。

如果驱动程序签名时使用的是倍福会签的 OEM 证书，则上述步骤就不需要了。

四、如何得到 TwinCAT user certificate 的会签签名？

在4024版本的TwinCAT3平台上，倍福为现有客户提供了一个“TwinCAT3 OEM user certificate”，这个证书可以对TwinCAT3上C++程序的TMX文件进行签名操作。

- 该证书要求对申请人的信息进行安全验证，因此，客户必须正式订购，以便验证地址和联系方式等信息。需要注意的是该证书只能颁发给Beckhoff现有的客户。
- 订购产品：TC0008。
- 签发该证书是免费的。
- 客户得到会签后的证书请保存在C:\TwinCAT\3.1\CustomConfig\Certificates路径下。

TwinCAT3 OEM Certificate只用于TMX文件的一次性签名操作，使用它签名的TMX文件时不需要这个证书。因此，TwinCAT3 OEM Certificate只需要部署在签署TMX文件的编程电脑而不需要放置在运行代码的运行时系统上。

出于安全的考虑，TwinCAT3 OEM user certificate的有效期为2年。过期后不能再使用它来签名新的TMX文件，但是，仍然可以不受任何限制地使用已经签名过的TMX文件。

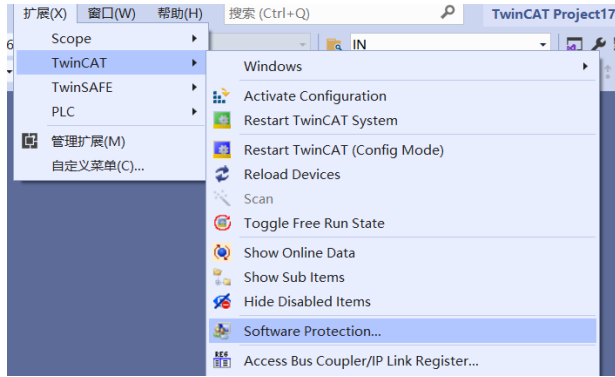
客户可以在两年有效期满前(甚至之后)申请证书的延期。

订购和得到会签证书的步骤如下：

1. 联系倍福的销售工程师，订购“TC0007”或“TC0008”。
2. 请提供联系人姓名、电子邮件地址、公司名称、公司地址等信息。
3. 倍福销售将核实订单中提供的联系方式等信息。
4. 按照步骤生成一个OEM证书的请求文件，详细步骤后续内容会专门讲解。
5. 订购TC0008的客户需要在生成OEM证书的时候确认指纹文件。
6. 将OEM证书的请求文件以及指纹信息、公司信息、联系人信息等按要求发送给倍福销售工程师。
7. 倍福德国总部授权后，对应的倍福销售工程师会将授权后的文件发送给客户。

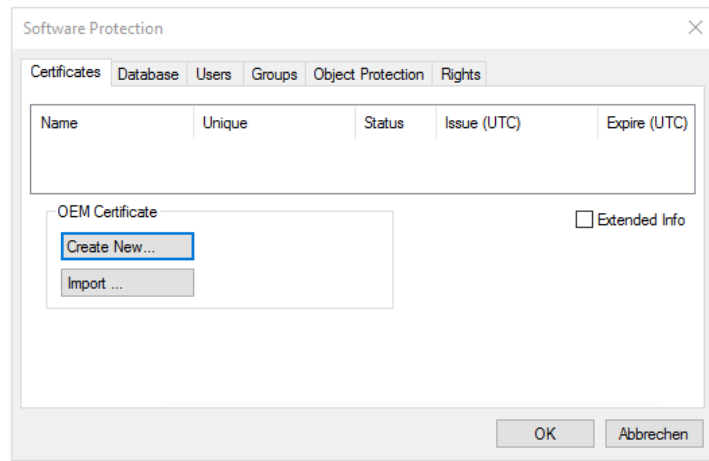
五. OEM 证书请求文件的创建步骤

1. 以VS2019为例，在“扩展(X)”主菜单中选择“TwinCAT”菜单项，然后选择“Software Protection...”



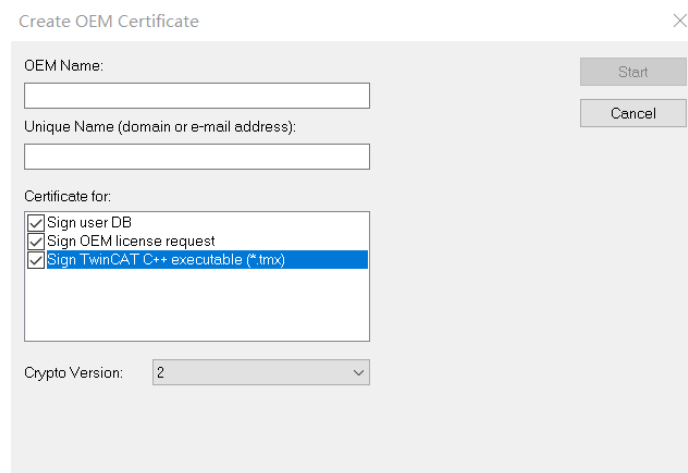
图十七 选择Software Protection项

2. 在打开的窗口中选择“Certificate”标签页，点击“Create New...”



图十八 创建一个新的证书

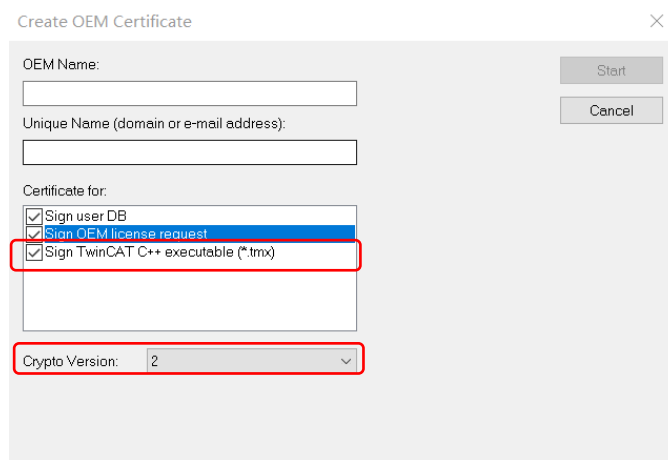
3. 在打开的窗口中输入相关信息



图十九 输入相关信息

- 在OEM Name文本框中输入公司名称。这个名称必须可以很清楚地说明你的公司或者你的业务单元。
- 在Unique Name文本框中输入公司域名或者你的公司邮箱。这个名称必须是证书所有者的唯一标识。
- 根据你的实际需求勾选服务项目：第一项和第二项与PLC程序的软件保护功能有关（TC0007），第三项与

TwinCAT驱动程序的签名有关，如果你的这个证书只用于TwinCAT驱动程序的签名，可以只勾选第三项，此时Crypto Version项必须设置为2。



图二十 确保红色框图中的项的设置

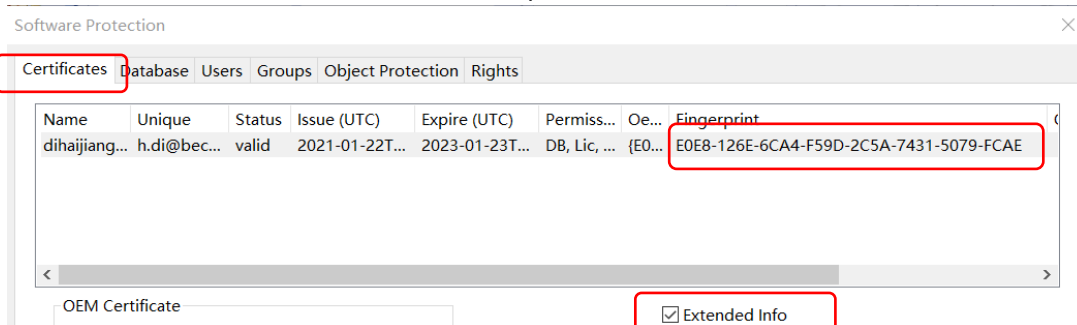
4. 输入相关数据后点击“Start”选择保存文件，默认的保存路径是：“c:\twincat\3.1\customconfig\certificates”。
5. 在随后打开的对话框中输入密码，并再次确认密码。**请妥善保存此密码，切勿忘记，在使用证书的时候需要这个密码。**

六. 确定 OEM 证书请求文件的指纹码

OEM证书的请求文件默认保存在“c:\twincat\3.1\customconfig\certificates”路径下。

当客户订购TC0008功能的时候，需要向倍福提供TwinCAT OEM Certificate证书的指纹码信息，具体操作步骤如下：

1. 在TwinCAT3工程文件中打开TwinCAT software protection配置窗口



图二十一 指纹码信息

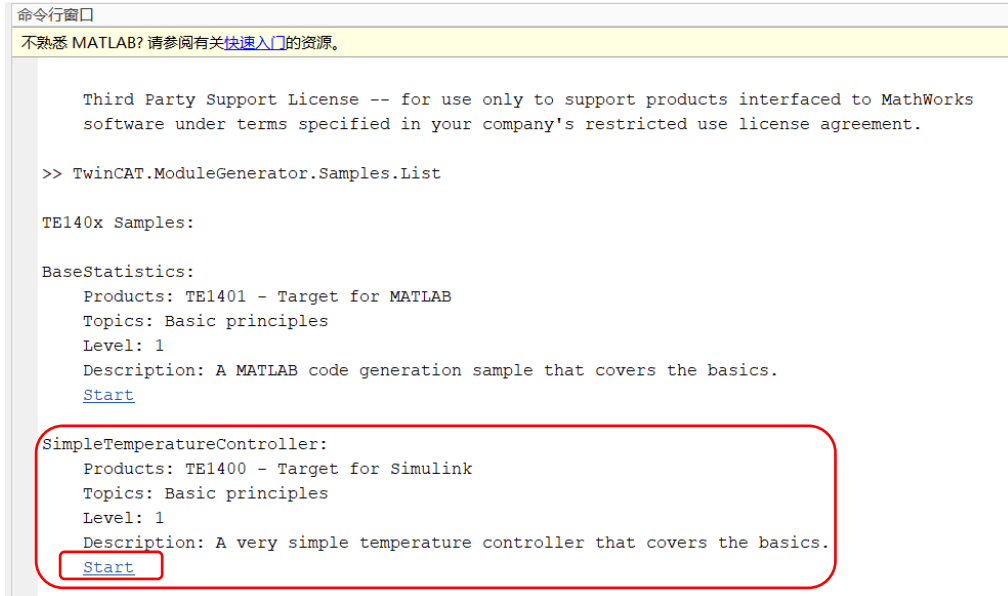
2. 选择“Certificate”标签。
3. 勾选“Extended Info”。
4. 在窗口中向右滚动，直到看到Fingerprint列。
5. 或者双击证书所在的行，可弹出一个对话框显示 Fingerprint 信息。通过快捷键“Ctrl + C”可以将消息窗口中的指纹数据拷贝到 Windows 剪贴板。

第三章 快速入门操作

一. 打开一个简单的 Simulink 模型

为了让用户能更快地熟悉TwinCAT Target for Simulink的使用方法，我们可以通过在MATLAB指令窗口中输入TwinCAT.ModuleGenerator.Samples.List来查看倍福提供的案例程序，这些案例程序中有详细的操作步骤演示。我们以SimpleTemperatureController案例为例演示。

1. 在Matlab命令行窗口中找到该案例，点击“Start”按钮。



```
命令行窗口
不熟悉 MATLAB? 请参阅有关快速入门的资源。

Third Party Support License -- for use only to support products interfaced to MathWorks
software under terms specified in your company's restricted use license agreement.

>> TwinCAT.ModuleGenerator.Samples.List

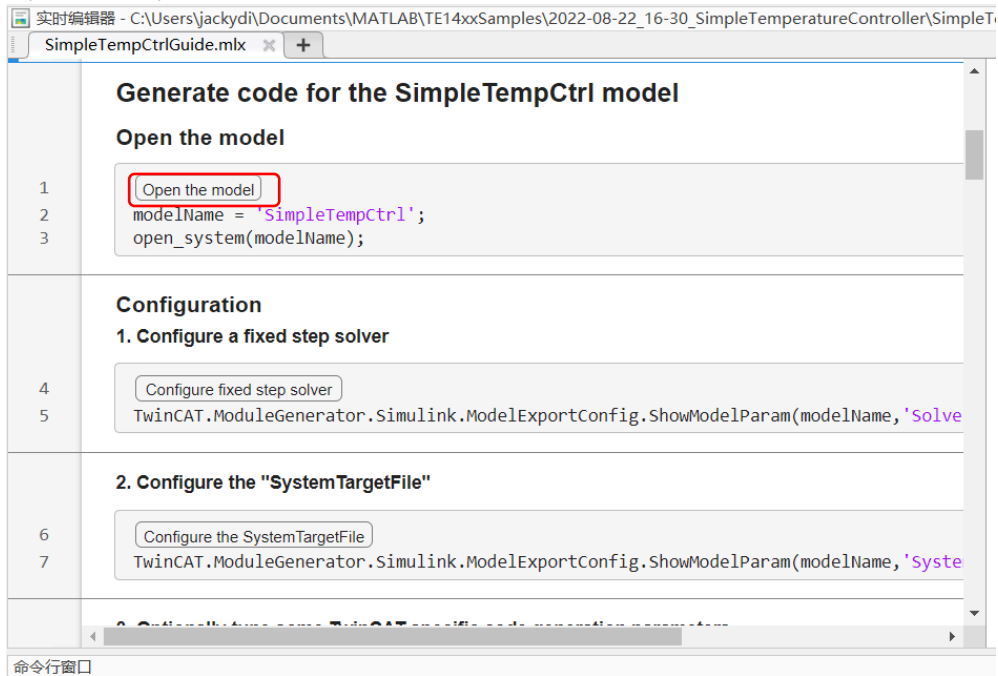
TE140x Samples:

BaseStatistics:
  Products: TE1401 - Target for MATLAB
  Topics: Basic principles
  Level: 1
  Description: A MATLAB code generation sample that covers the basics.
  Start

SimpleTemperatureController:
  Products: TE1400 - Target for Simulink
  Topics: Basic principles
  Level: 1
  Description: A very simple temperature controller that covers the basics.
  Start
```

图二十二 在MATLAB命令行窗口选择演示案例

2. 在Live Script中选择“open the model”按钮，开始演示整个封装步骤。



```
实时编辑器 - C:\Users\jackydi\Documents\MATLAB\TE14xxSamples\2022-08-22_16-30_SimpleTemperatureController\SimpleT
SimpleTempCtrlGuide.mlx x +

Generate code for the SimpleTempCtrl model

Open the model

1 [Open the model]
2 modelName = 'SimpleTempCtrl';
3 open_system(modelName);

Configuration

1. Configure a fixed step solver

4 [Configure fixed step solver]
5 TwinCAT.ModuleGenerator.Simulink.ModelExportConfig.ShowModelParam(modelName, 'solve

2. Configure the "SystemTargetFile"

6 [Configure the SystemTargetFile]
7 TwinCAT.ModuleGenerator.Simulink.ModelExportConfig.ShowModelParam(modelName, 'system

命令行窗口
```

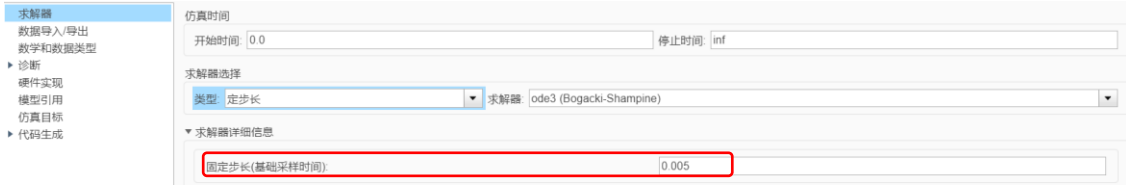
图二十三 打开案例程序窗口

当然，你也可以自己编写一个简单的simulink程序。

二. 在 Simulink 中的配置步骤

如果仅仅是演示整个操作步骤，我们可以在Live Script中逐个点击案例程序提供的按钮即可，这里不再赘述。假设你打开的simulink模型是自己开发完成的，那么请按照如下步骤设置simulink中的参数。

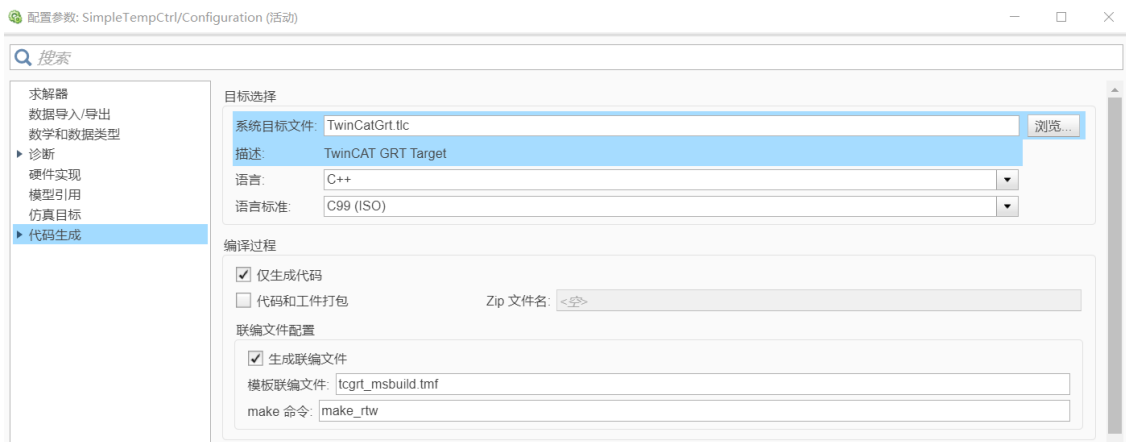
1. 在“求解器”中选择“Fixed-step”。



图二十四 求解器中的参数配置

在“求解器详细信息”中可以设定固定步长的数值，单位s。

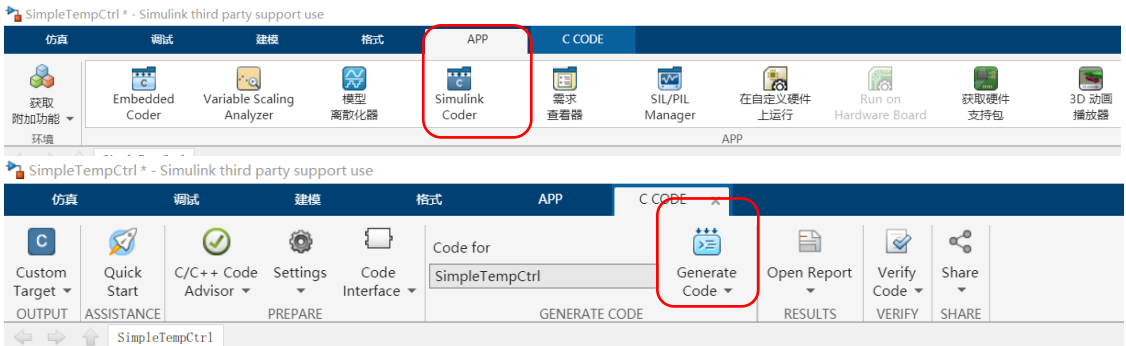
2. 在“系统目标文件”中选择“TwinCatGrt.tlc”。



图二十五 代码生成中参数的设置

3. 点击“应用”和“确定”按钮保存上述设置。

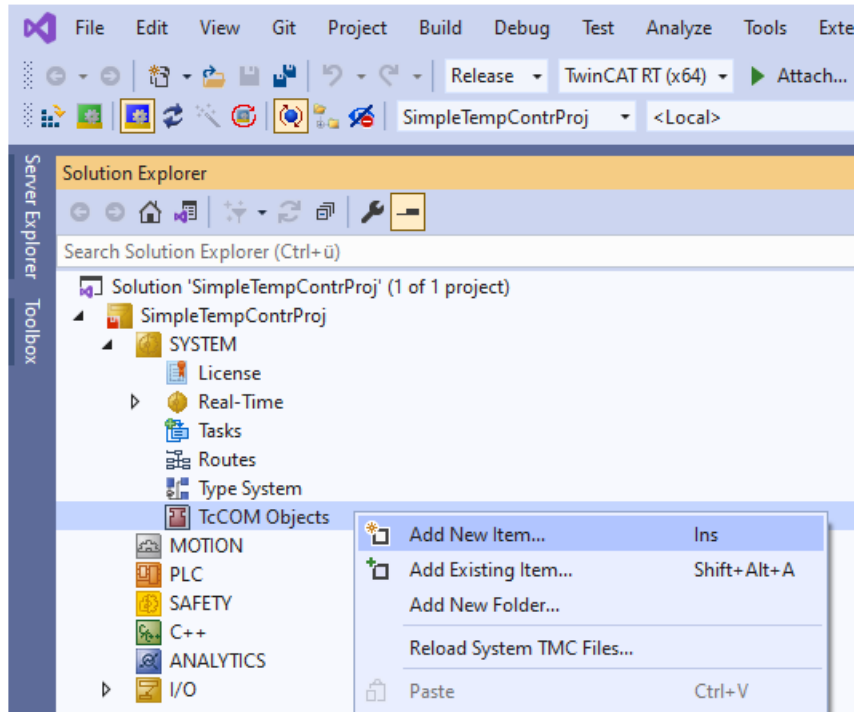
4. 在simulink窗口中的APP中选择“Simulink code”启动代码编译“Generate code”。



图二十六 Simulink窗口中的操作

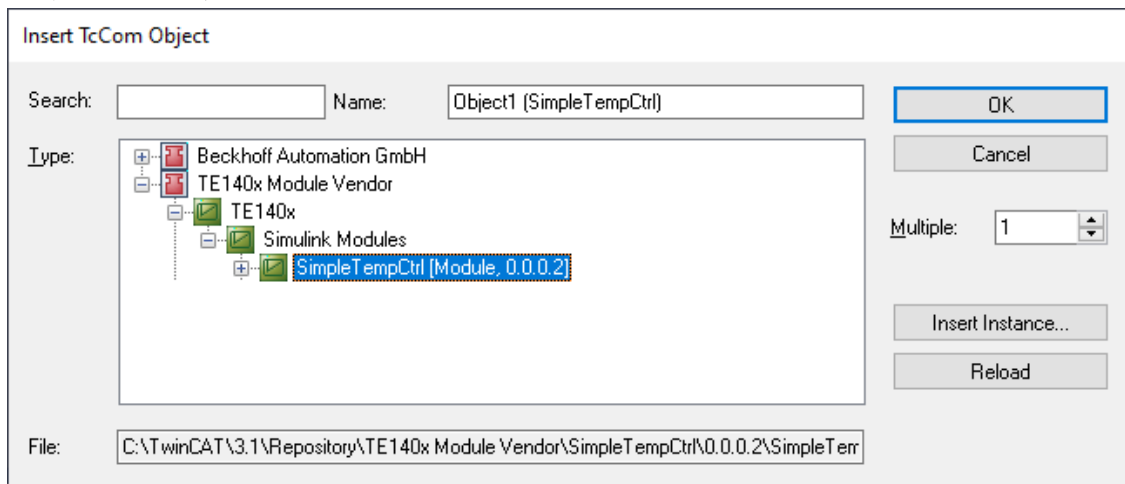
三. 在 TwinCAT 中的操作

1. 打开TwinCAT工程，创建一个项目。
2. 实例化一个TcCOM组件。



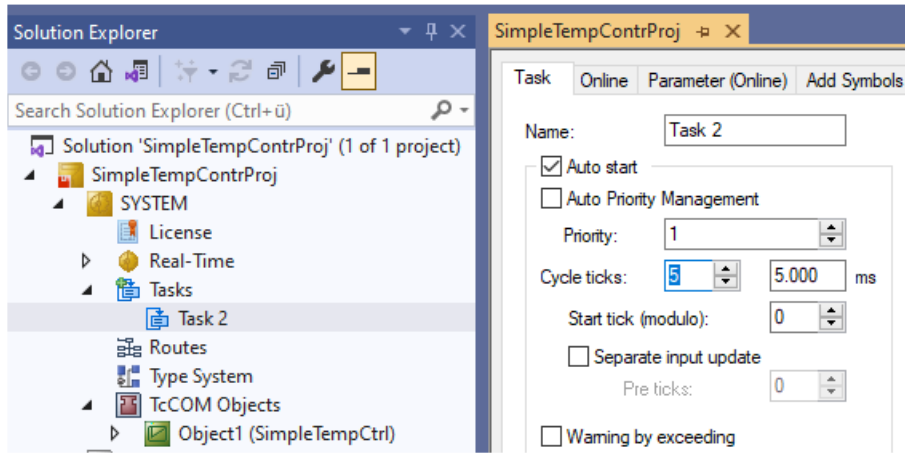
图二十七 添加TcCOM组件

3. 添加Simulink封装后的TcCOM组件。



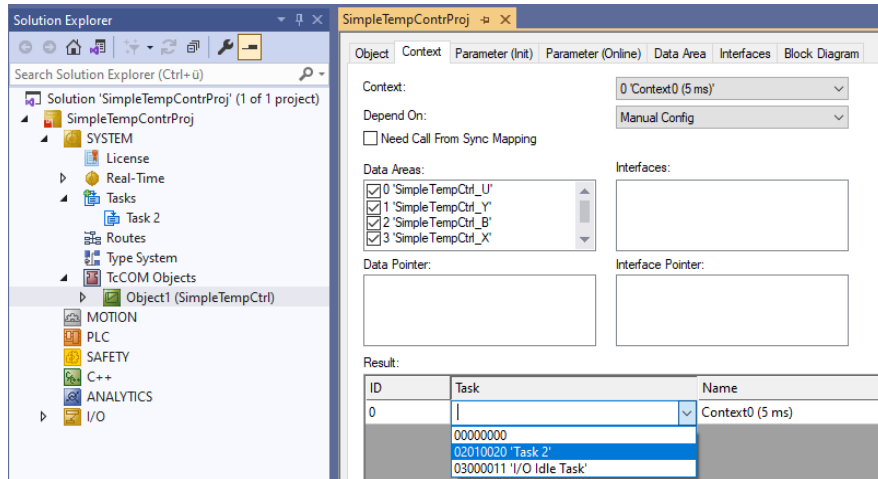
图二十八 选择所需要的模型

4. 创建一个任务，并设定任务周期。请注意周期的数值要与封装Simulink时求解器固定步长时的设定相同。



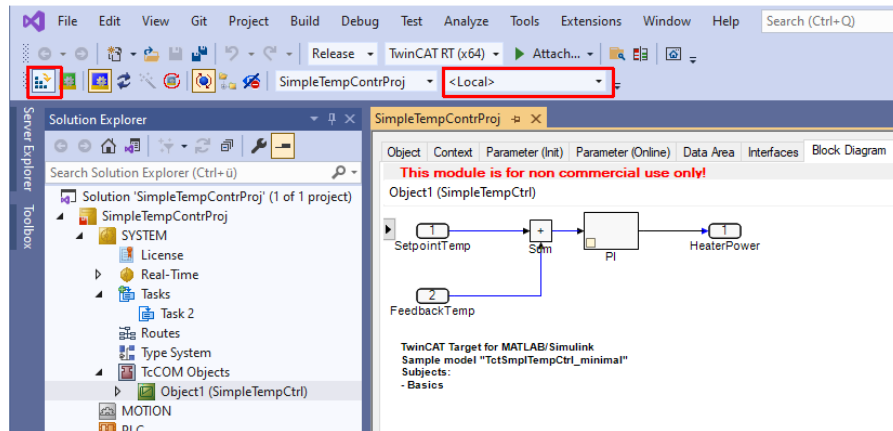
图二十九 创建任务

5. 将该任务分配给TcCOM组件。



图三十 分配任务

6. 激活配置。



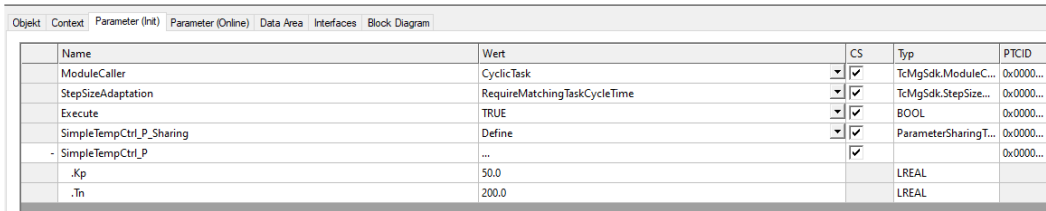
图三十一 激活配置

四. TcCOM 组件的参数设置及链接

TcCOM实例的数据交换是通过过程映像的映射进行的。Simulink输入和Simulink输出被自动映射为过程映像中

的输入或输出，并可与I/O或其他对象相链接。

在TcCOM实例的“Parameter (Init)”标签页下，你可以有选择地对参数进行初始化的设置。例如，你可以在这里将参数Kp设置为“52”，TcCOM模块将使用这个值作为这个实例的初始化值。



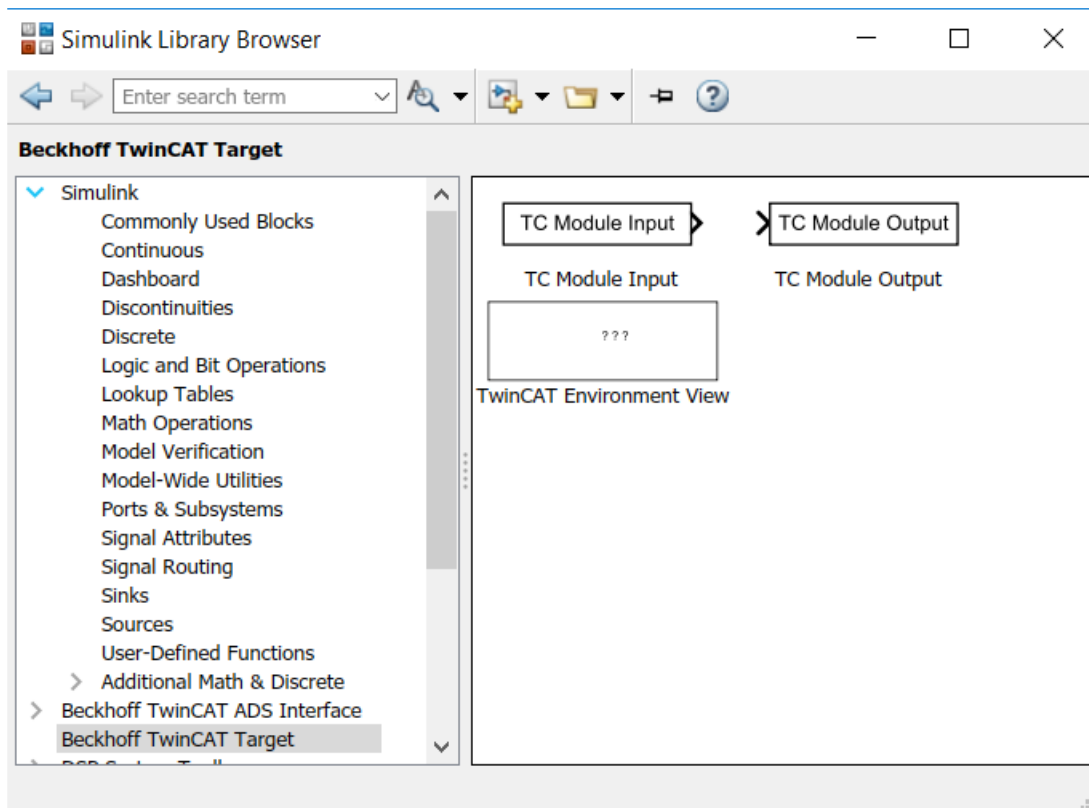
| Name | Wert | CS | Typ | PTCID |
|--------------------------|------------------------------|-------------------------------------|----------------------|-----------|
| ModuleCaller | CyclicTask | <input checked="" type="checkbox"/> | TcMgSdk.ModuleC... | 0x0000... |
| StepSizeAdaptation | RequireMatchingTaskCycleTime | <input checked="" type="checkbox"/> | TcMgSdk.StepSize... | 0x0000... |
| Execute | TRUE | <input checked="" type="checkbox"/> | BOOL | 0x0000... |
| SimpleTempCtrl_P_Sharing | Define | <input checked="" type="checkbox"/> | ParameterSharingT... | 0x0000... |
| - SimpleTempCtrl_P | ... | <input checked="" type="checkbox"/> | | 0x0000... |
| .Kp | 50.0 | | LREAL | |
| .Tn | 200.0 | | LREAL | |

图三十一 参数的初始化

第四章 Simulink 中的 TwinCAT 库

在Simulink的工具箱中倍福提供了一个直接供TwinCAT输入输出模块使用的功能块。这些输入输出功能块可以提供给TwinCAT作为输入输出使用。以往通常的做法是直接使用Simulink自带的标准的输入和输出。

打开Simulink的“Library Browser”，在“Beckhoff TwinCAT Target”下可以找到TwinCAT指定的输入输出功能块。



图三十二 Beckhoff在Simulink工具箱中增加的库

相对于使用Simulink提供的标准的输入输出端口而言，使用Beckhoff提供的TwinCAT指定输入输出模块会有如下好处：

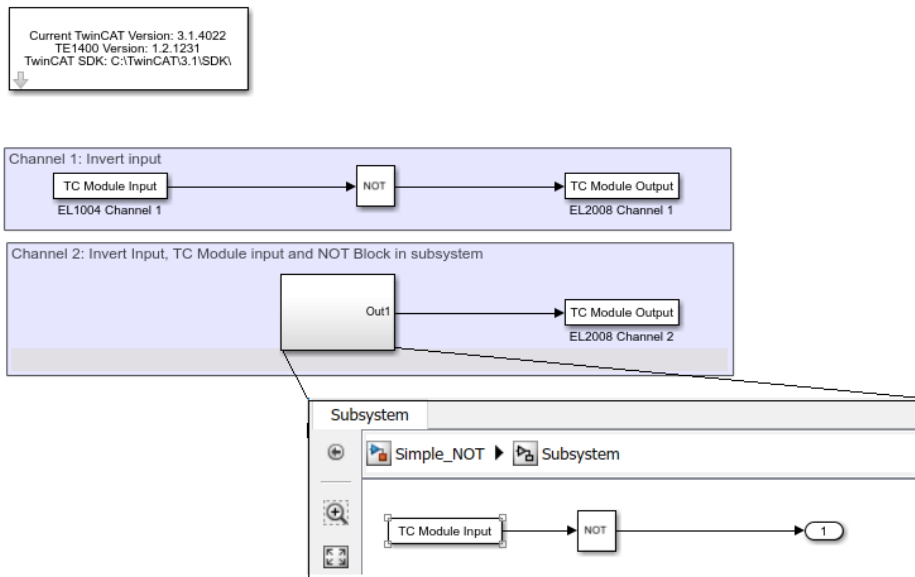
- 可以直接在子系统中定义TcCOM的输入和输出信号，而不再需要先将信号从子系统中传输到上层系统。
- 在使用Beckhoff提供的Simulink输入输出模块时，可以直接在模块参数上设置其链接和映射，这样在实例化TcCOM组件的时候，变量就自动链接到相关的I/O模块或者其他TcCOM组件了。（非强制使用）

在使用自动映射的时候需要注意，如果该TcCOM组件被多次实例化，则存在地址冲突的问题，这时就需要手动地将链

接关系重新设置。

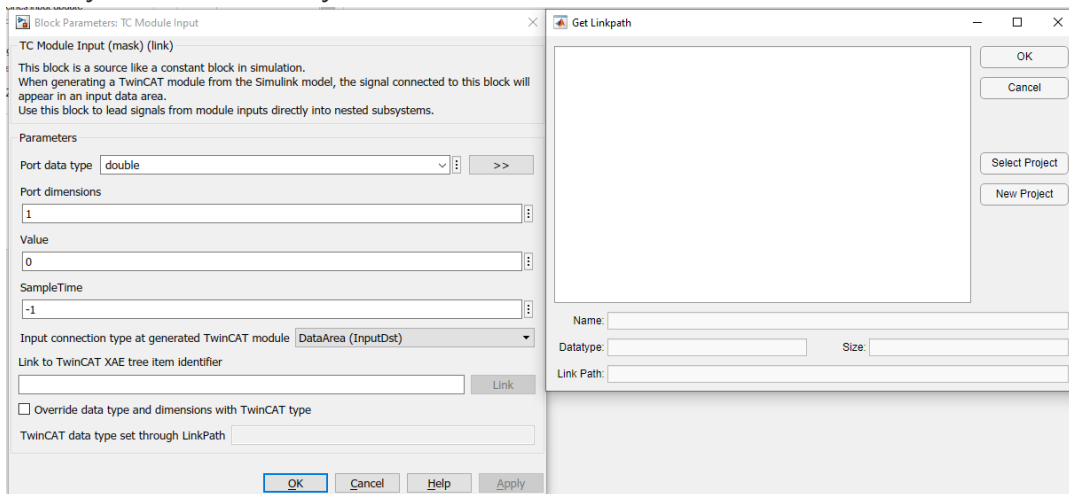
实现步骤：

创建一个Simulink模型，它的两个输出是对各自输入的取反。其中一个输入被放置在子系统中，根据上述的第一个属性， 子系统内的Beckhoff提供的输入输出功能块可以出现在TwinCAT的过程映像中直接链接外部I/O模块。



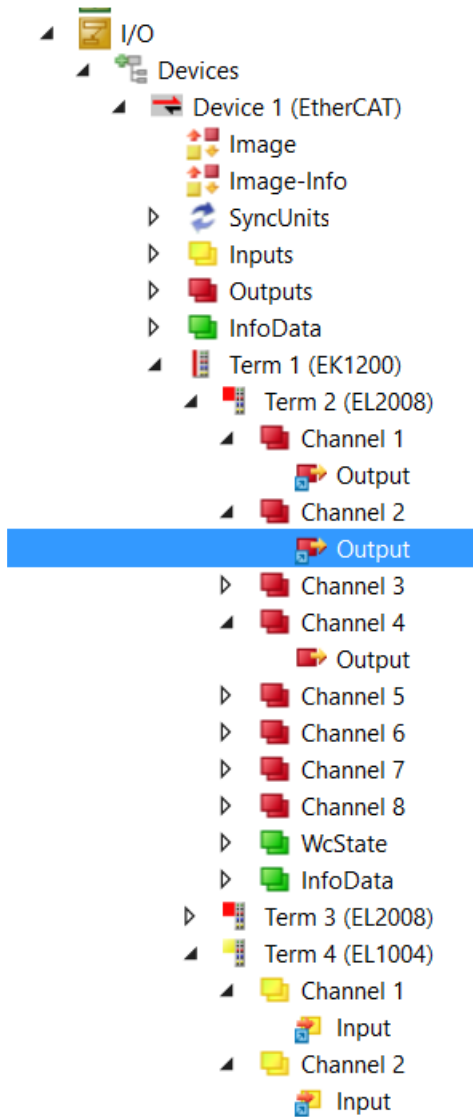
图三十三 Simulink中Beckhoff提供的输入输出功能块

通过TC Module Input和TC Module Output功能块的属性可以使得Simulink模型的输入输出可以直接自动地映射到I/O模块。鼠标左键双击选中的TC Module Input模块，在弹出的对话框中选择Link，然后弹出一个新的对话框，在新的对话框中选择“Select Project”或者“New Project”然后选择该信号对应的输入输出通道。



图三十四 通过属性设置自动链接关系

如果上述Simulink模型被编译成TcCOM并集成到TwinCAT 3解决方案中，则会自动创建Simulink中选择的输入和输出与对应I/O模块的映射。自动生成的映射被标记为蓝色，而手动映射的则为白色。

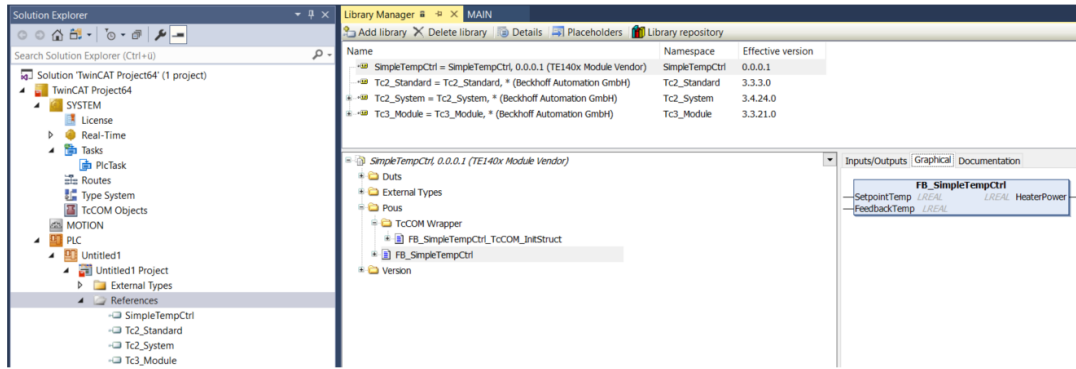


图三十五 自动链接的变量和模块通道

第五章 将 Simulink 模型直接封装成 PLC 库的操作

一. 创建并安装 PLC 库的步骤及参数设置

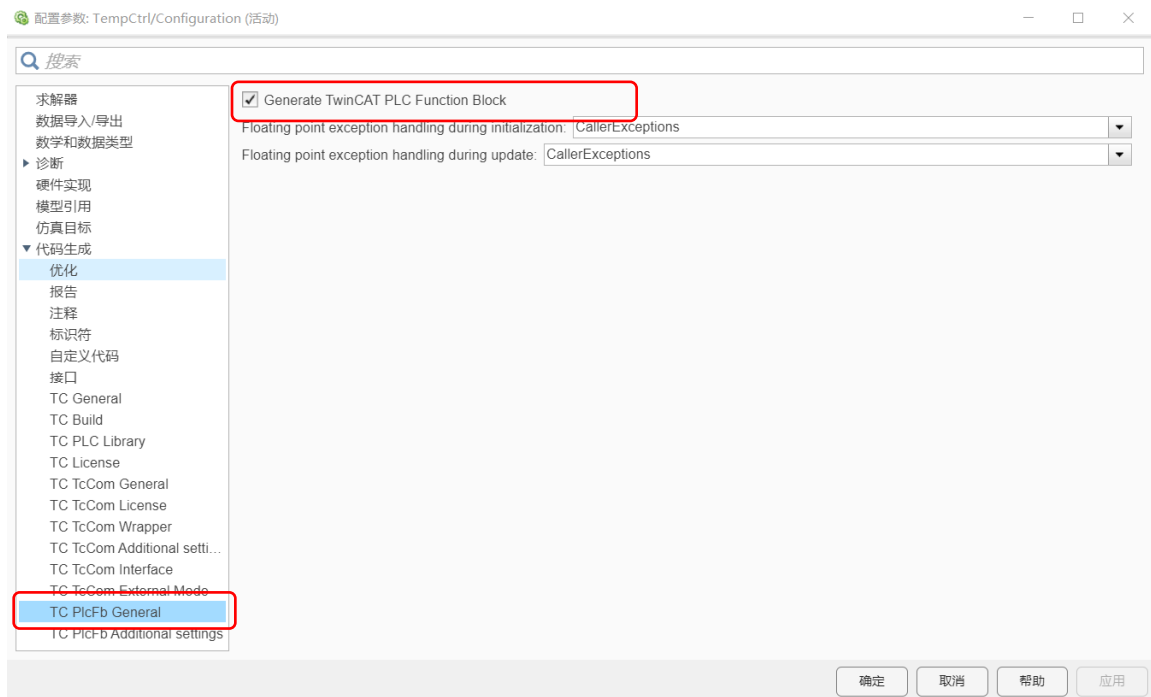
Simulink模型除了可以封装为TcCOM组件外，还可以封装成可以在PLC中直接调用的库，在库文件中包含了两种可以调用的功能块FB。第一种功能块会被命名为FB_<modelName>，它包含了Simulink模型的全部功能，源代码被直接封装在这个功能块中，我们把这种功能块称为**PLC-FB**；第二种功能块存放在库文件POU文件夹下面TcCOM Wrapper子文件夹下，它是TcCOM组件的FB封装函数，即：功能并不在FB功能块中，而是外包给TcCOM的一个实例。我们把这种功能块叫做**TcCOM-Wrapper-FB**。



图三十六 安装库文件后产生的功能块

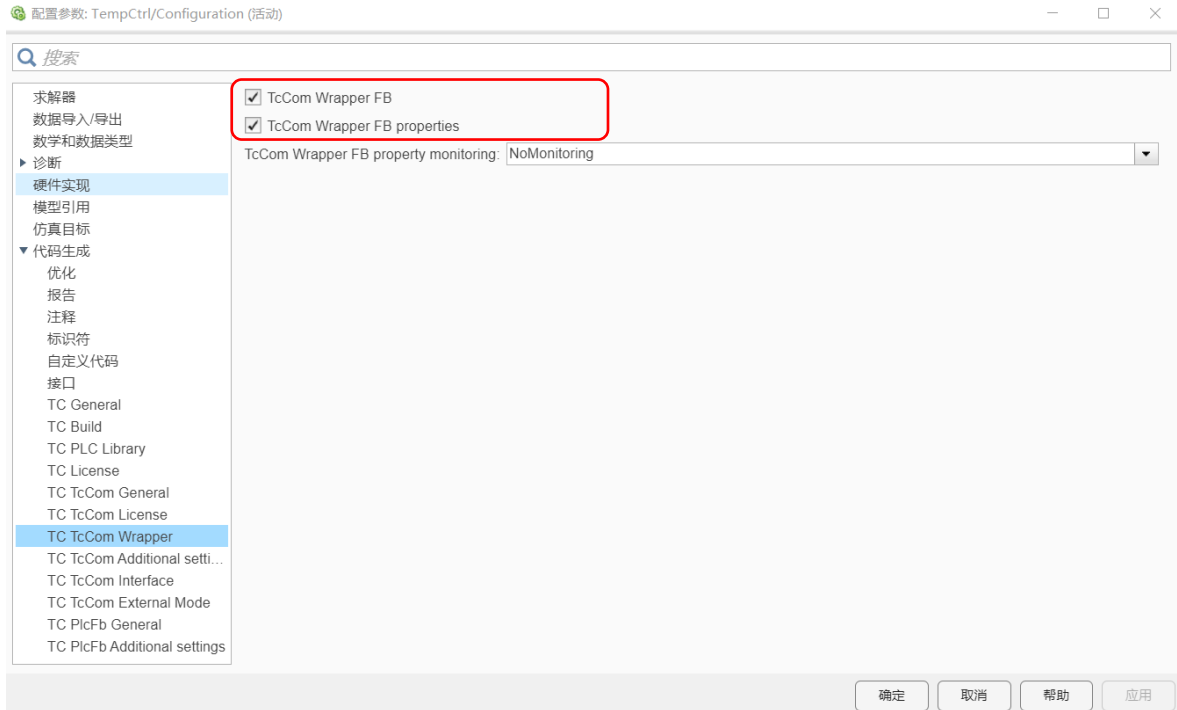
Simulink模型封装成库文件并安装在PLC中的步骤如下：

1. 打开一个Simulink模型。
2. 在配置参数中导航到**Configuration Parameters > Code Generation > TC PlcFb General**，勾选如下选项，可生成**PLC-FB**。



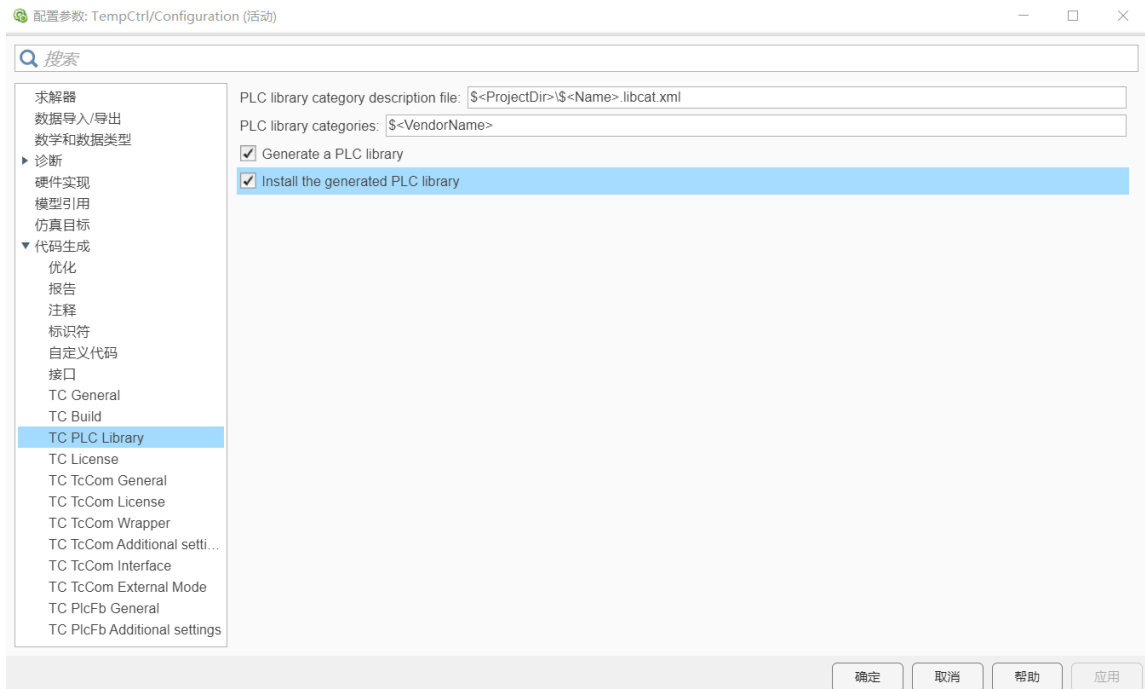
图三十七 TC PLCFb General接口参数设置

3. 在配置参数中导航到**Configuration Parameters > Code Generation > TC TcCom Wrapper**，勾选如下选项可生成**TcCOM Wrapper FB**。



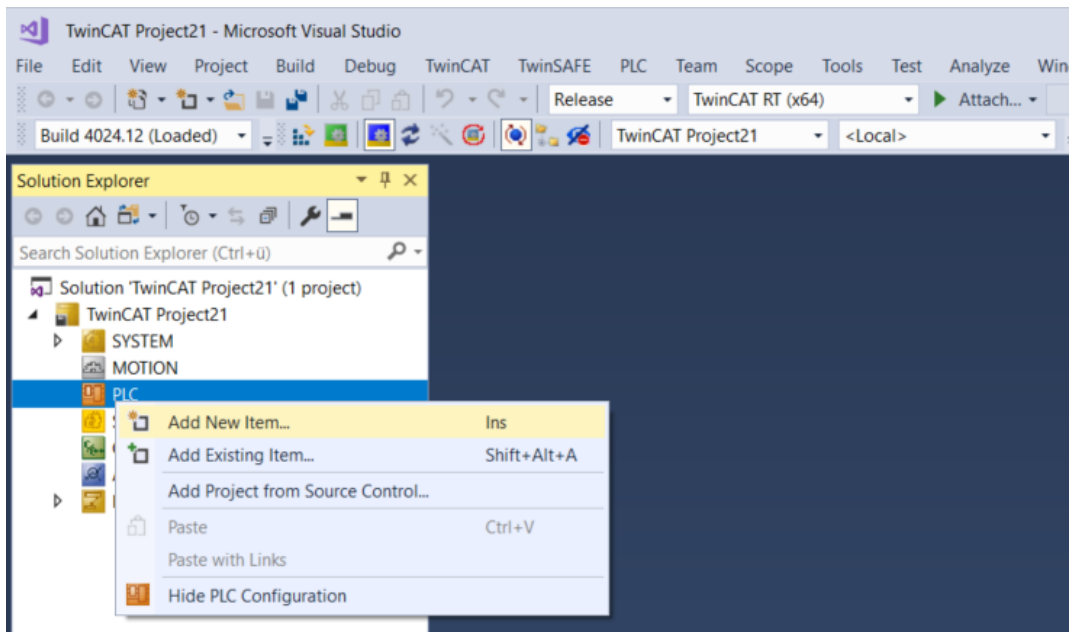
图三十八 TC TcCOM Wrapper接口参数的设置

4. 导航到Configuration Parameters > Code Generation > TC PLC Library, 勾选如下选项。



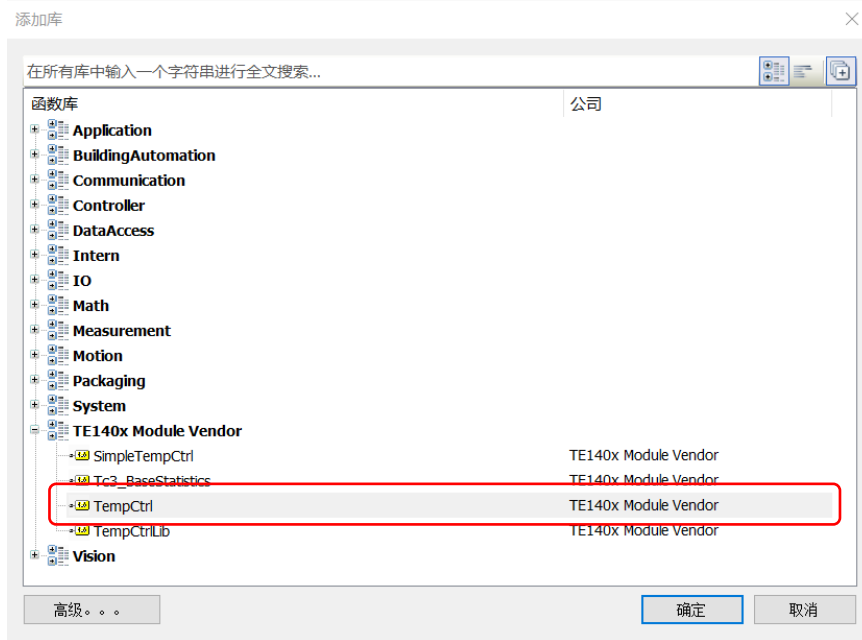
图三十九 TC PLC Library接口参数的设置

5. 参数设置完成后对Simulink模型进行封装。
6. 首先确保封装Simulink模型的电脑和对PLC进行编程的电脑是同一台。
7. 创建一个PLC工程。



图四十 创建PLC工程

8. 导入生成的PLC库。



图四十一 安装PLC库

9. 如果封装好的库文件也要分享给其他工程师使用，可在如下路径找到生成的.library文件：

C:\TwinCAT\3.1\Repository\TE140x Module Vendor\TempCtrl。这个库文件就可以安装到其他开发电脑上了。

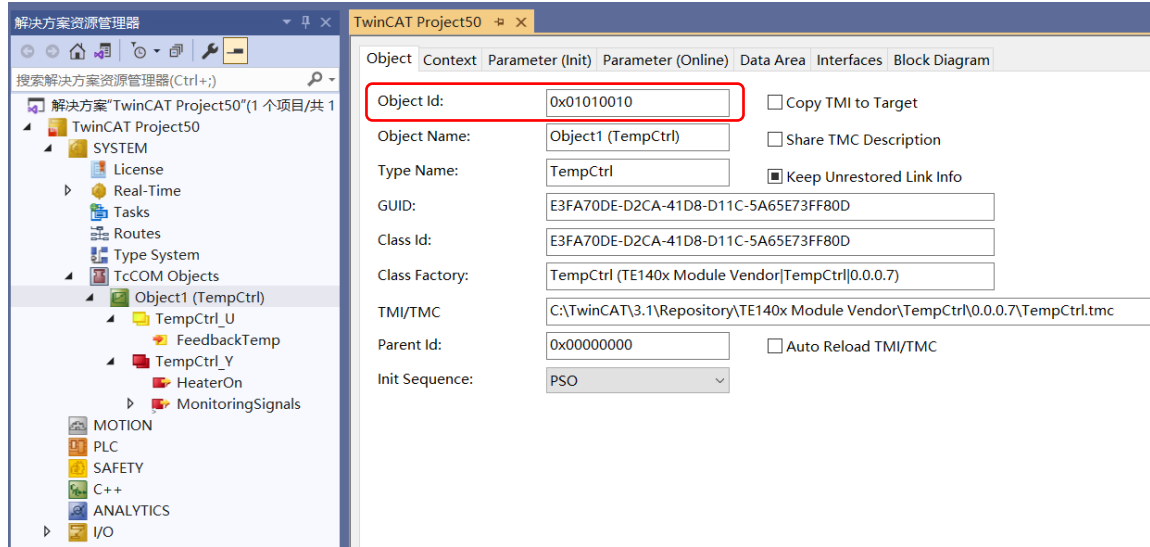
二. TcCOM Wrapper FB 的应用

有两种方法可以实现PLC中调用TcCOM组件，第一种是静态调用，第二种是动态调用。不论用那种方法都会用到已经安装的PLC库中的TcCOM Wrapper FB。

第一种方式：引用静态模型实例

实现步骤如下：

1. 在System > TcCOM Objects点击右键添加封装好的TcCOM组件。TcCOM Wrapper FB功能块就可以访问该模块实例了。

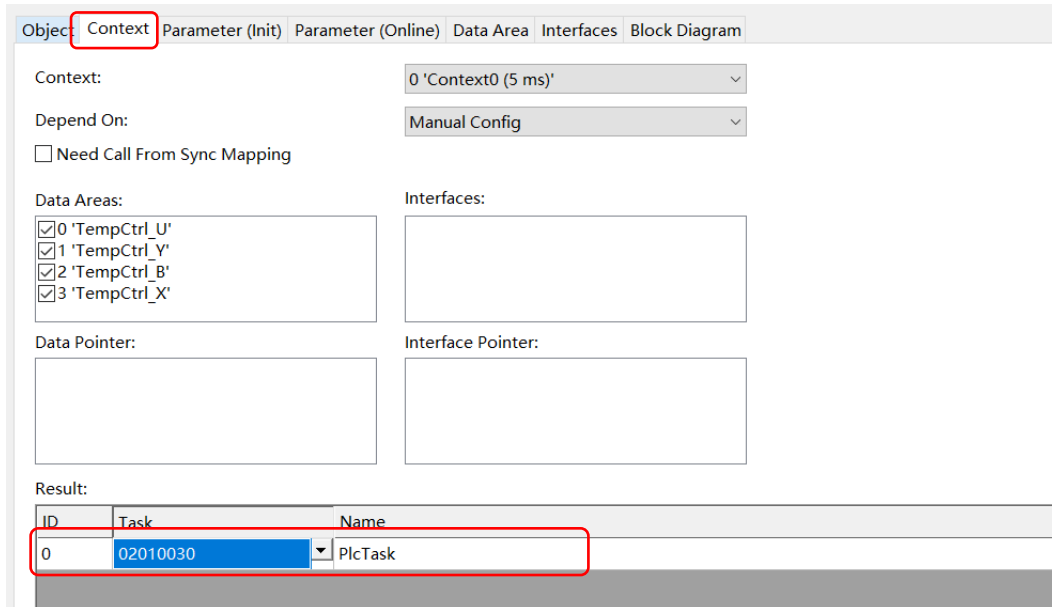


图四十一 添加TcCOM组件

2. 对于静态调用的情形，在功能块实例化声明过程中必须设置对应模块实例的对象ID。

```
// link wrapper with a static instance
InitStrStatic      : ST_FB_TempCtrl_TcCOM_InitStruct := (noid:= 16#01010010); // OID from objeot1
fbTempCtrStatic    : FB_TempCtrl_TcCOM_InitStruct(InitStrStatic);
Inputs             : ST_ExtU_TempCtrl_T; // data type defined in SimpleTempCtrl library
Outputs           : ST_ExtY_TempCtrl_T;
```

3. 设置TcCOM组件的“Context”与调用TcCOM Wrapper FB功能块的PLC程序为同一个任务（Task）。



图四十七 设置相同的任务

4. 设置TcCOM组件中“Parameter (init)”参数“Module Call”为：“Module”。

| Object | Context | Parameter (Init) | Parameter (Online) | Data Area | Interfaces | Block Diagram |
|--------------------|------------------------------|------------------|--------------------|-----------|------------|---------------|
| | | | | | | |
| Name | Value | CS | Type | PTCID | | |
| ModuleCaller | Module | ☑ | TcMgSdk.Mo... | 0x00... | | |
| StepSizeAdaptation | RequireMatchingTaskCycleTime | ☑ | TcMgSdk.Ste... | 0x00... | | |
| ExecutionSequence | UpdateBeforeOutputMapping | ☑ | TcMgSdk.Exe... | 0x00... | | |
| Execute | TRUE | ☑ | BOOL | 0x00... | | |
| TempCtrl_P_Sharing | Inherit | ☑ | ParameterSha... | 0x00... | | |
| + TempCtrl_P | ... | ☑ | | 0x00... | | |

图四十八 调用方式的设置

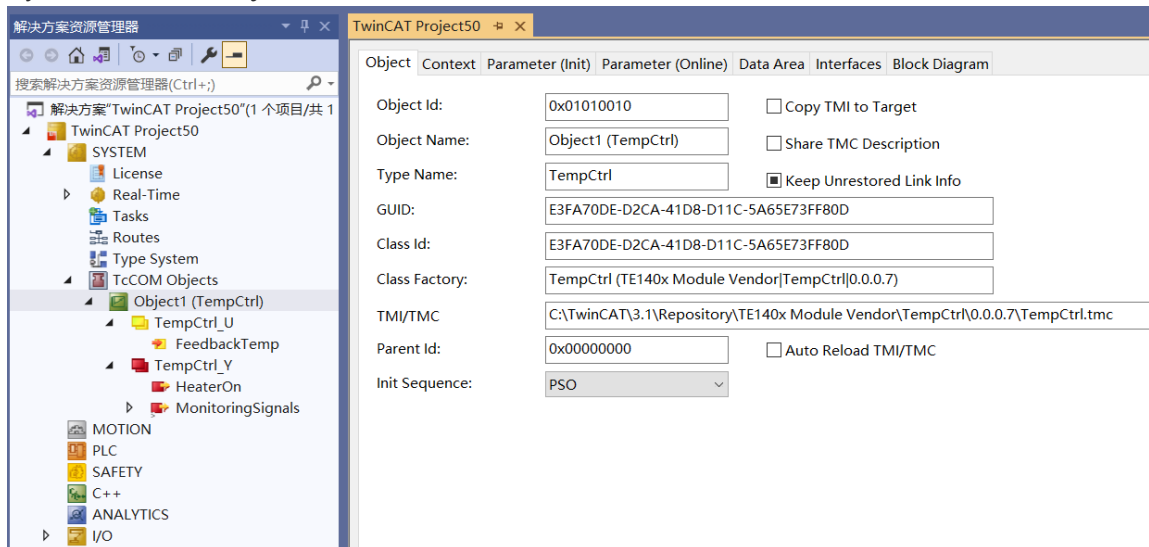
5. 使用这种调用方式，TcCOM所需的内存从系统的non paged pool中获得。

第二种方式：在PLC中动态实例化和应用

TcCOM Wrapper FB 通过自身在PLC中生成对TcCOM组件的实例并链接到Wrapper，在这种调用方式下，编程人员可以使用默认参数，也可以自定义TcCOM组件中的相关参数。

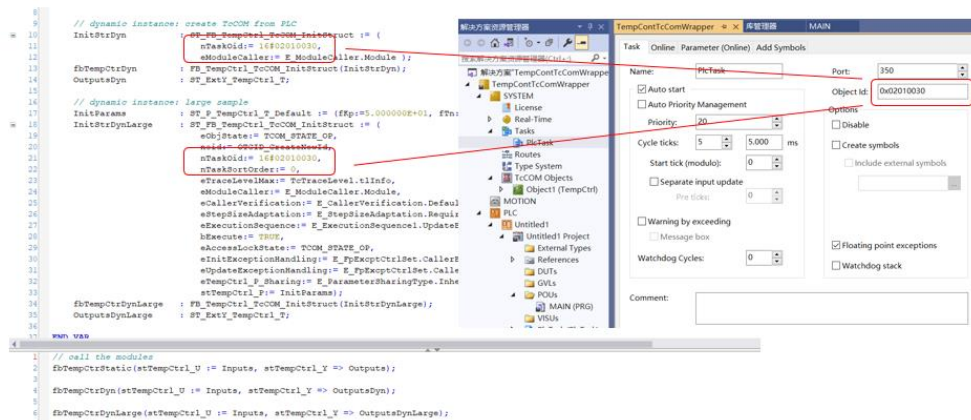
实现步骤如下：

1. 在System > TcCOM Objects点击右键添加封装好的TcCOM组件。



图四十九 添加TcCOM组件

2. 在PLC程序中需要将调用TcCOM Wrapper FB的任务的TaskOid分配给功能块的参数。



图五十 动态实例化调用的TaskOid分配

3. 设置TcCOM组件中“Parameter (init)”参数“Module Call”为：“Module”。
4. 在这种调用方式下TcCOM组件所需的内存是从ADS Router Memory中获得。

```

// dynamic instance: create TcCOM from PLC
InitStrDyn
: ST_FB_TempCtrl_TcCOM_InitStruct := (
  nTaskOid:= 16#02010030, // take TaskOID of PloTask
  eModuleCaller:= E_ModuleCaller.Module ); // set module caller to "call by module"
fbTempCtrlDyn
: FB_TempCtrl_TcCOM_InitStruct (InitStrDyn);
OutputsDyn
: ST_ExtY_TempCtrl_T;

// dynamic instance: large sample
InitParams
: ST_P_TempCtrl_T_Default := (fRp:=5.000000E+01, fFn:=2.000000E+02);
InitStrDynLarge
: ST_FB_TempCtrl_TcCOM_InitStruct := (
  eObjState:= TCOM_STATE_OP, // target object state is OP
  noid:= OTCID_CreateNewId, // create a new object
  nTaskOid:= 16#02010030, // task oid of PloTask
  nTaskSortOrder:= 0, // sort order of generated TcCOM
  eTraceLevelMax:= TcTraceLevel.tlInfo,
  eModuleCaller:= E_ModuleCaller.Module, // mandatory setting!
  eCallerVerification:= E_CallerVerification.Default, // next params: parameters as defined in T
  eStepSizeAdaptation:= E_StepSizeAdaptation.RequireMatchingTaskCycleTime,
  eExecutionSequence:= E_ExecutionSequence1.UpdateBeforeOutputMapping,
  bExecute:= TRUE,
  eAccessLockState:= TCOM_STATE_OP,
  eInitExceptionHandling:= E_FpExcpCtrlSet.CallerExceptions,
  eUpdateExceptionHandling:= E_FpExcpCtrlSet.CallerExceptions,
  eTempCtrl_P_Sharing:= E_ParameterSharingType.Inherit, // inherit, as dyn instance of fbTempCtrlDyn
  stTempCtrl_P:= InitParams ); // initialize model with these parameters
fbTempCtrlDynLarge
: FB_TempCtrl_TcCOM_InitStruct (InitStrDynLarge);
OutputsDynLarge
: ST_ExtY_TempCtrl_T;
  
```

图五十一 使用默认参数和用户自定义参数的Wrapper动态调用

三. PLC-FB 功能块的应用

按照第一部分的描述将Simulink模型封装成的PLC库安装到TwinCAT工程中，该库文件中有一个可以调用的PLC-FB。在PLC程序中可以对这个功能块进行一个或者多个的实例化调用。

```

1 PROGRAM MAIN
2 VAR
3   fpTempCtrl           : FB_TempCtrl;
4   nFeedbackTemp       : INT;
5   bHeaterOn           : BOOL;
6   stMonitoringSignals : ST_MonitoringSignalsType;
7
8 END_VAR
9
10 fpTempCtrl (nFeedbackTemp:=nFeedbackTemp , bHeaterOn=>bHeaterOn , stMonitoringSignals=>stMonitoringSignals );
  
```

图五十二 PLC-FB功能块的调用

当TwinCAT处于运行模式时，您可以在TwinCAT XAE中更新PLC库版本，并可通过在线更改将其加载到正在运行的应

用程序中。这意味着PLC库中的所有功能块都可以在不重启TwinCAT的情况下进行更新。

1. 使用TwinCAT Target for Simulink创建第一个版本的PLC库。
2. 在PLC工程中安装这个生成的PLC库。
3. 激活这个PLC库版本（例如Ver.0.0.0.1）下的TwinCAT工程。
4. 在Simulink中更新代码，重新封装创建新的PLC库（Ver.0.0.0.2）。
5. 选择**Build > Build Solution**重新编译整个工程。
6. 选择**Login > Login with online change**在线更新整个工程。

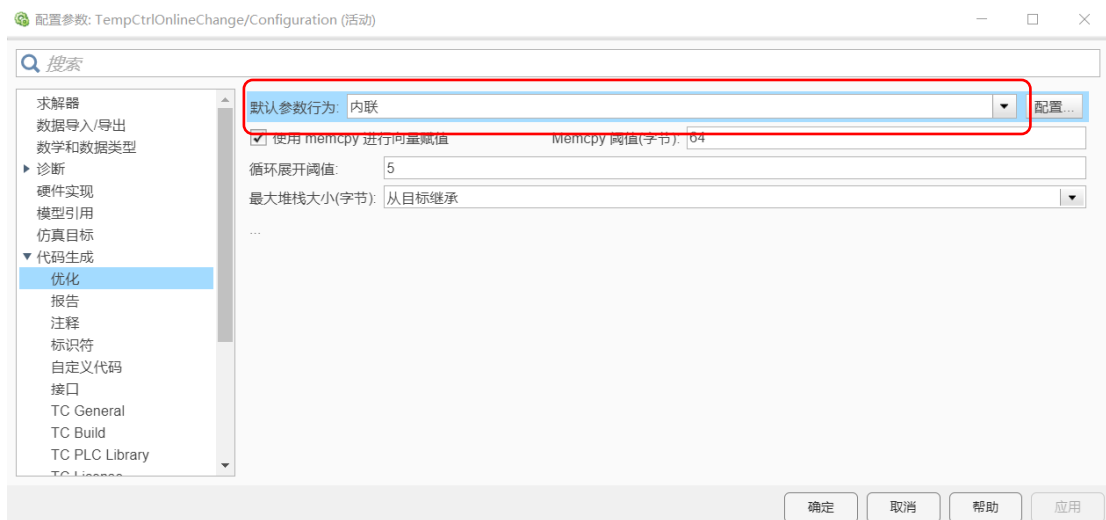
第六章 TcCOM 组件在线更新操作

从Simulink模型封装的每个TcCOM组件对象都包含版本信息。因此，您可以选择Simulink模型的任何版本，并在TwinCAT中有选择性地实例化。

当我们选择一个版本的TcCOM组件激活运行后，可否允许在线的情况下切换另外的版本呢？

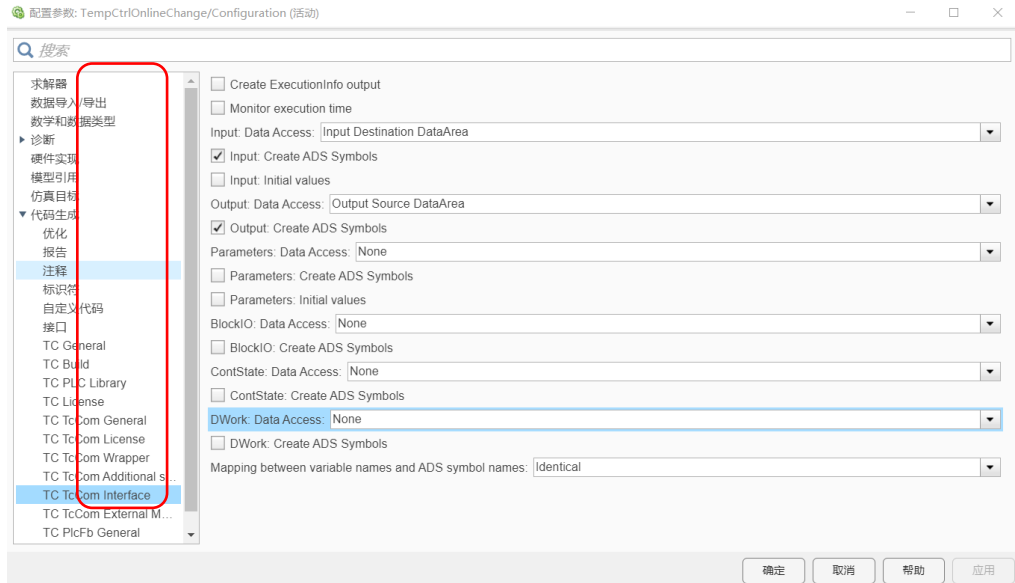
按照如下步骤操作可轻松实现TcCOM组件的在线更新：

1. 打开Simulink模型，导航到**Configuration Parameters > TC TcCom General**，勾选**Online Change Support**。
2. 为防止ADS访问修改模型内部的参数，可选择如下设置，但这个设置不是必须的。



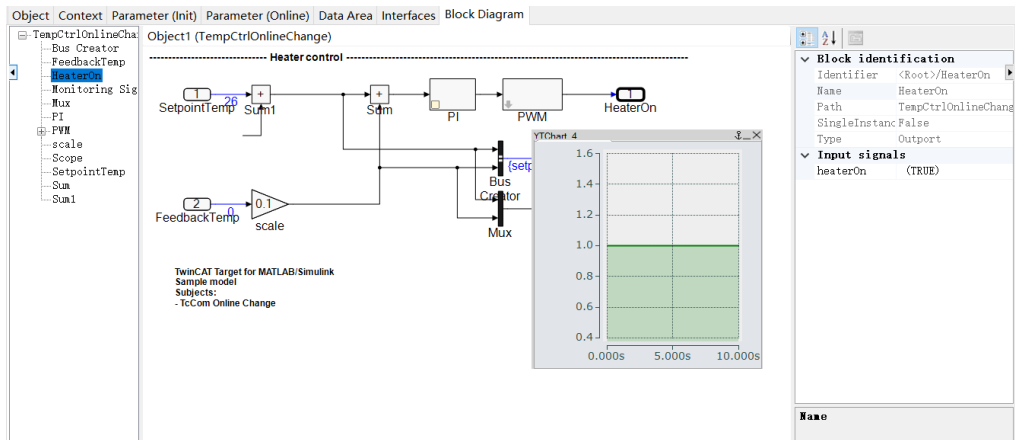
图五十三 优化接口参数设置

3. 根据实际需要设置模型接口参数的读写操作限制。



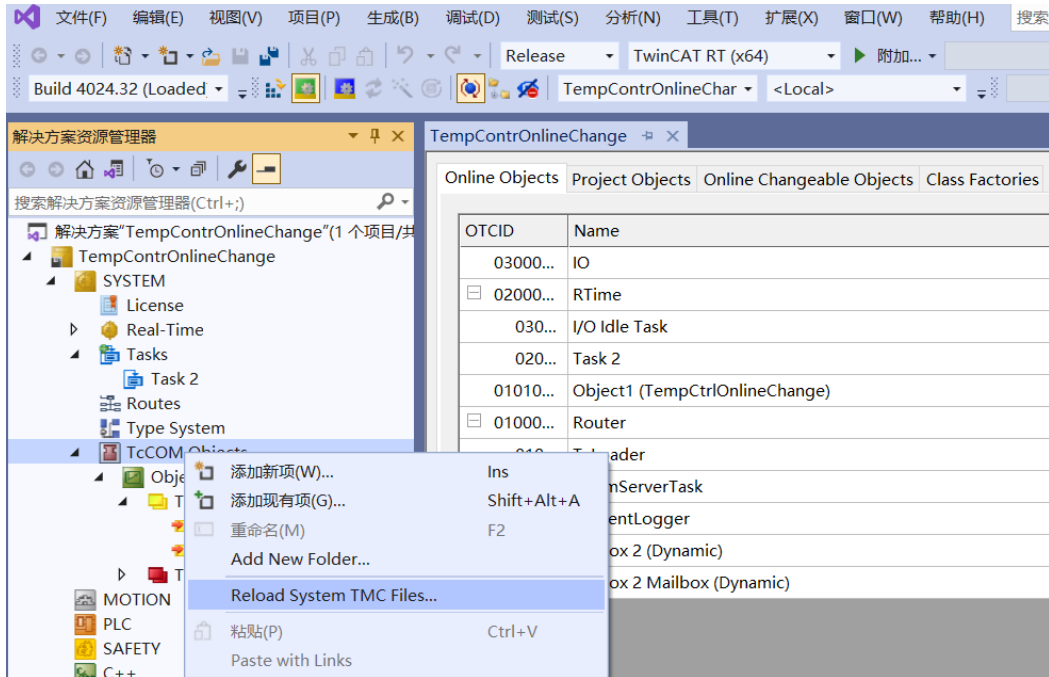
图五十四 设置接口参数的读写操作限制

4. 保存设置并封装Simulink代码。
5. 打开TwinCAT工程在TcCOM组件调用模型，分配任务后激活运行。



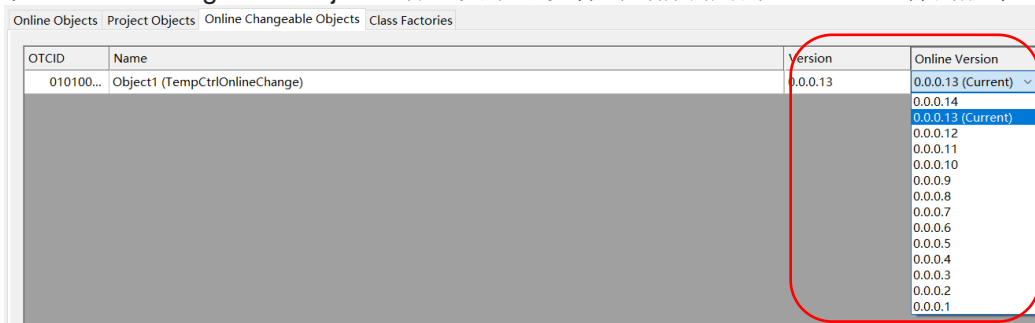
图五十四 激活运行第一个版本的TcCOM组件

6. 在TwinCAT工程中Reload TMC file.



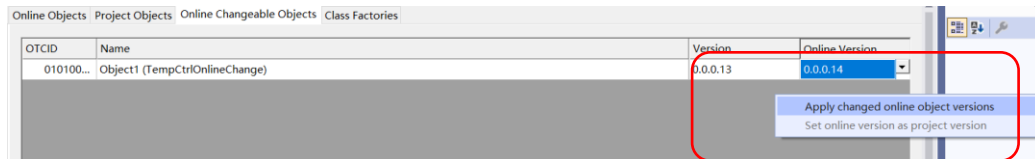
图五十五 重新导入系统TMC文件

7. 在右侧“Online Changeable Object”标签页下可以看到当前调用的 TcCOM 组件的版本。



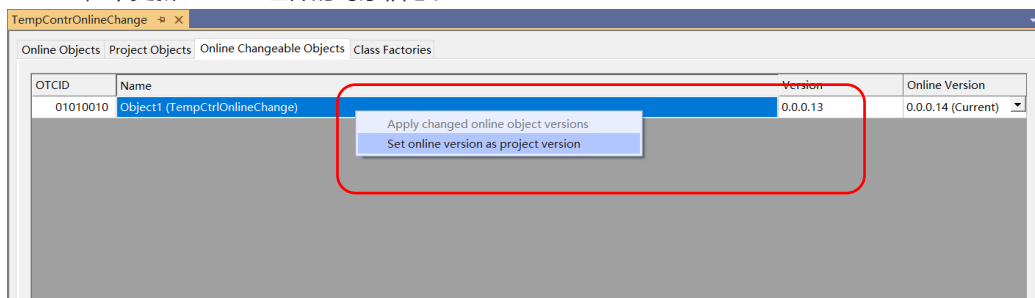
图五十六 在线更新组件标签

8. 选择新生成的版本并更新。



图五十七 更新到最新版本

9. 在TwinCAT工程中更新TcCOM组件的对象信息。



10. 组件对象的版本更新完成。



| OTCID | Name | Version | Online Version |
|----------|--------------------------------|----------|----------------------|
| 01010010 | Object1 (TempCtrlOnlineChange) | 0.0.0.14 | 0.0.0.14 (Current) ▾ |

图五十七 相关信息更新完成

11. 在“Object”和“Block Diagram”两个标签页都可以看到更新后的TcCOM组件相关信息。



请用微信扫描二维码
通过公众号与技术支持交流

上海（中国区总部）

中国上海市静安区汶水路 299 弄 9 号（市北智汇园）
电话: 021-66312666

北京分公司

北京市西城区新街口北大街 3 号新街高和大厦 407 室
电话: 010-82200036
邮箱: beijing@beckhoff.com.cn

广州分公司

广州市天河区珠江新城珠江东路 16 号高德置地 G2603 室
电话: 020-38010300/1/2
邮箱: guangzhou@beckhoff.com.cn

成都分公司

成都市锦江区东御街 18 号百扬大厦 2305 房
电话: 028-86202581
邮箱: chengdu@beckhoff.com.cn

倍福官方网站:

<https://www.beckhoff.com.cn>

在线帮助系统:

https://infosys.beckhoff.com/index_en.htm

倍福虚拟学院:

<https://tr.beckhoff.com.cn/>

招贤纳士: job@beckhoff.com.cn

技术支持: support@beckhoff.com.cn

产品维修: service@beckhoff.com.cn

方案咨询: sales@beckhoff.com.cn