**TcCom开发：接口指针实现PLC与C++模块的无缝集成**

|  |  |
| --- | --- |
|  | 作者：余洋  职务：技术支持工程师  公司：BECKHOFF中国  邮箱：yang.yu@beckhoff.com.cn  日期：2025-02-19 |
| **摘 要：**  TcCom开发和Windows COM组件开发，基本原理一致。借助COM（组件对象模型），TcCom模块能够在TwinCAT3环境中实现高效的模块间通信。 TcCom的主要特点包括线程安全的数据交换、灵活的内存管理以及对多任务环境的支持，使其成为复杂自动化系统中模块化通信的理想选择。本文介绍了在TwinCAT3平台上，通过接口指针实现PLC与C++模块之间的高效通信方法。文档一共分为三部分：如何使用TwinCAT3创建TcCom模块、使用PLC库文件进行不同场景下TcCom代理功能块的开发以及应用例程演示。 | |
| **附 件：**   |  |  |  | | --- | --- | --- | | 序 号 | 文件名 | 备注 | | 1. | Driver.zip | TcCom组件 | | 2. | TwinCAT TcCom Proxy Application Example | 例程 | | 3. | TcComProxyApplication | PLC库文件 | | 4. | EventloggerApplication | PLC库文件 | | 5. | PLCTcComProvider | PLC库文件 | | |
| **历史版本：**   |  |  |  | | --- | --- | --- | |  |  |  | |  |  |  | |  |  |  | | |
| **免责声明：**  我们已对本文档描述的内容做测试。但是差错在所难免，无法保证绝对正确并完全满足您的使用需求。本文档的内容可能随时更新，如有改动，恕不事先通知，也欢迎您提出改进建议。  **参考信息：** | |

目 录

[1. 软硬件版本 3](#_Toc191989653)

[1.1. 控制硬件 3](#_Toc191989654)

[1.2. 控制软件 3](#_Toc191989655)

[2. TcCom Object的创建 3](#_Toc191989656)

[2.1. C++ 创建 TcCom Object 3](#_Toc191989657)

[2.2. PLC创建TcCom Object 7](#_Toc191989658)

[2.3. 附件文件讲解 9](#_Toc191989659)

[3. 库文件TcComProxyApplication 10](#_Toc191989660)

[3.1. 功能块FB\_ProxyTcComModule 11](#_Toc191989661)

[3.2. 功能块FB\_DynProxyTcComModule 16](#_Toc191989662)

[3.3. 功能块FB\_ProxyPLCTcComModule 20](#_Toc191989663)

[4. 库文件其他函数和功能块 23](#_Toc191989664)

[4.1. 功能块FB\_TcCom 23](#_Toc191989665)

[4.2. 函数PVOID\_TO\_ITcomObject 23](#_Toc191989666)

[4.3. 函数ComQueryInterface 23](#_Toc191989667)

[4.4. 接口IProxyManagement 24](#_Toc191989668)

[5. 库文件Event解析 25](#_Toc191989669)

[6. 常见问题 26](#_Toc191989670)

# 软硬件版本

## 控制硬件

测试电脑（Windows11专业版）

## 控制软件

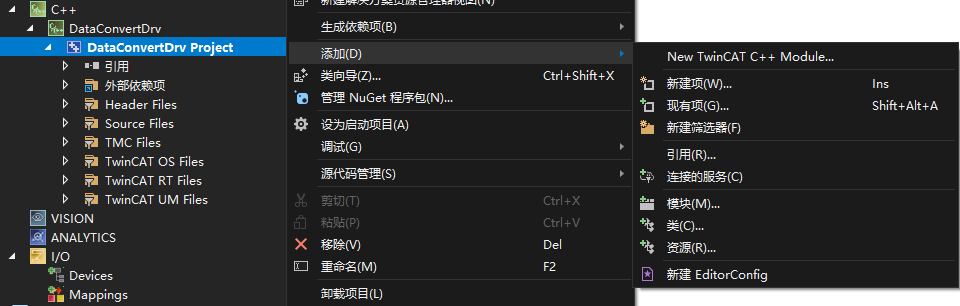
TwinCAT3（文档测试版本为v 3.1.4024.62）、Visual Studio 2019 C++开发组件。

（详情见：<https://tr.beckhoff.com.cn/mod/resource/view.php?id=2710>）

# TcCom Object的创建

## C++ 创建 TcCom Object

1. 新建一个TwinCAT C++项目，模板选择为带实时上下文的C++模板：



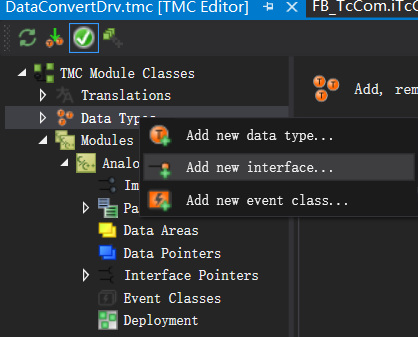


1. 双击进入C++项目里的TMC编辑器：

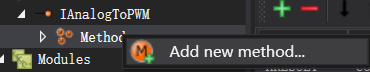
图形用户界面, 网站

描述已自动生成

（3）选择Data Types，右键选择Add new interface：



（4）本文创建的接口其作用是对模拟量数据进行转化至数字量数据，将模拟量信号(-100,100)的值转化对应占空比的数字量信号。创建接口底下的方法，并编辑方法的参数。该接口一共有两个方法，一个是方法Convert，输入为模拟量信号，类型为LREAL类型，还有两个BOOL类型指针的输入，这是我想从接口方法中处理完返回的值，所以这里用到了指针。（TestCounter仅为测试用，没有实际意义）另外一个方法是InitParam，用来初始化一个周期的PWM信号代表的时间。



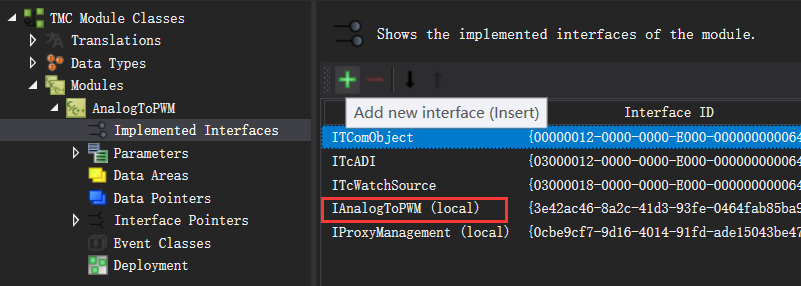
图形用户界面

AI 生成的内容可能不正确。

图片包含 图形用户界面

AI 生成的内容可能不正确。

（5）右键Implemented Interfaces，添加新的接口，选择刚刚创建的接口类型，该步骤代表当前的C++ TcCom模块是实现这个接口的对象实例类型：



（6）右键C++项目根据TMC文件自动生成代码：

电脑萤幕的截图

AI 生成的内容可能不正确。

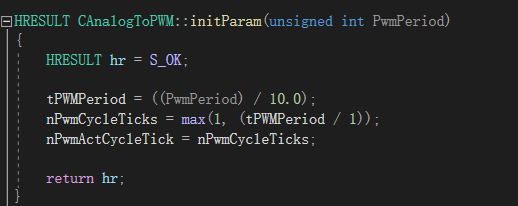
（7）对于该场景的应用，C++编写的TcCom模块是实现这个接口的对象实例，所以下一步是在CPP文件中编写对应实现接口的代码。这里算法具体的实现细节就不展示了，本文档在此只做参考。使用以实际情况为主。

以该模块为例，接口方法一：Convert的实现：

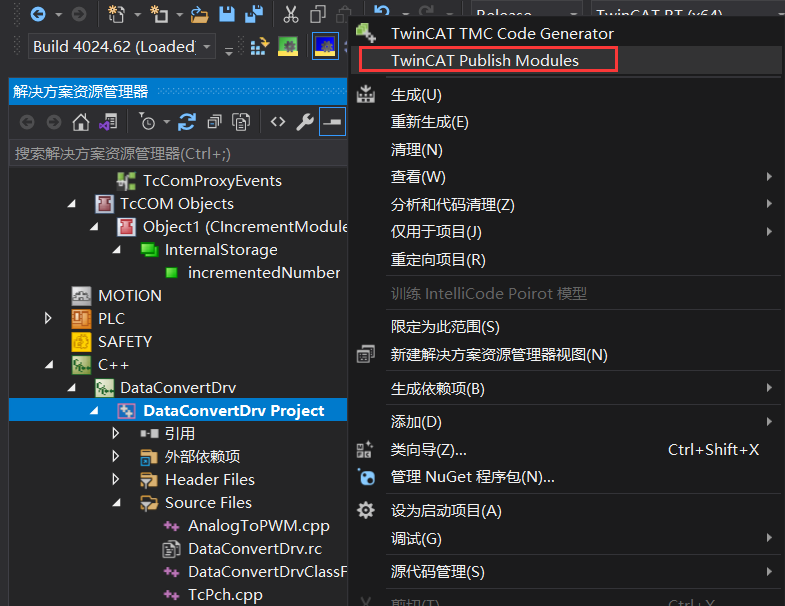
文本

AI 生成的内容可能不正确。

接口方法二：InitParam的实现：



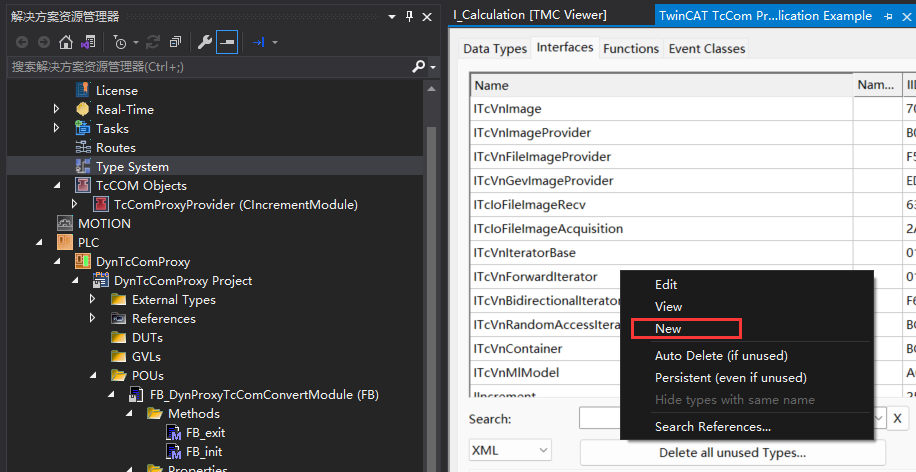
（8）发布TcCom模块：

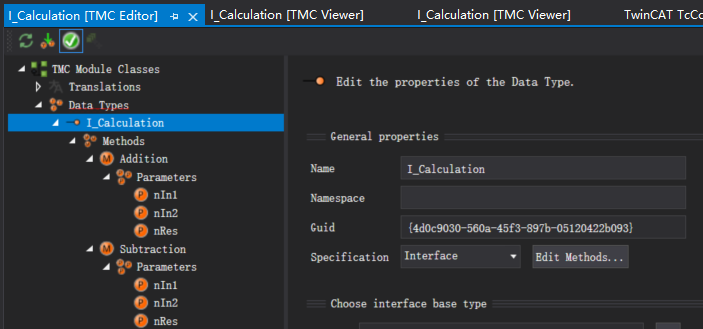


## PLC创建TcCom Object

TcCom Object不仅仅可以通过C++创建，使用PLC程序也可以创建TcCom模块：

（1）在在选择系统的Type System，在Interface中创建一个接口I\_Calculation，接口的编辑方式和2.1使用C++项目中的TMC编辑器一致，这里就不赘述了，接口中带一个加法和减法的方法：





（2）在PLC项目中创建一个功能块，拓展Tc3\_Module库文件中的TcBaseModuleRegistered功能块，再实现接口I\_Calculation，并添加C++兼容的属性：

图形用户界面, 文本

AI 生成的内容可能不正确。

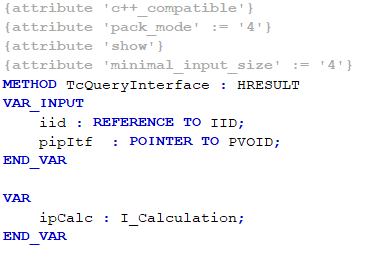
编写实现接口加减法实现的方法代码：

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。 图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

（3）重写TcQueryInterface方法，在内部先进行接口IID的比较，如果相同，将方法内部的I\_Calculation接口指向实现的实例，也就是功能块本体，再将指针的指针的所指向的内容通过接口转化PVOID指针的函数进行转化，最后增加引用计数器。如果接口ID不一致，调用父类的TcQueryInterface方法（该部分无须懂原理和为何这样写，固定搭配，抄写即可）：

 文本

AI 生成的内容可能不正确。

（4） 加一个属性，将父类的TcBaseModuleRegistered功能块中的OTCID与之Get绑定：



如果在同一个TwinCAT项目中，到这一步就可以了。但如果想要在其他的TwinCAT项目中使用这个功能块，推荐将功能块和其对应的接口，一并打包到库文件中，进行导出，接口需要Pin External Types以确保接口类型不会丢失。且建立一个全局变量区域，将TcCom功能块（例程中为FB\_Calculation）实例化在全局区域内：

图形用户界面, 应用程序, Word

AI 生成的内容可能不正确。

使用该PLC创建的TcCom Object见章节3.3。

（注：生成库文件的方式见：入门教程：[TwinCAT 3 自定义库的创建与使用 | 倍福虚拟学院](https://tr.beckhoff.com.cn/mod/hvp/view.php?id=2274)）

## 附件文件讲解

本文的配套的附件Driver中，有两个文件夹，分别对应两个TcCom模块的驱动文件，两个驱动文件中的TcCom模块分别只暴露了一个接口。一个是上述模拟量转数字量模块AnalogToPWM实现的接口IAnalogToPWM，（后续引用见章节3.2）另一个模块CIncrementModule中的接口IIncrement方法doIncrement可以进行自增累加，累加量为用户自定义的值（后续引用见章节3.1）。

在进行下一章节之前，需要将这两个文件夹放入“C:\TwinCAT\3.1\CustomConfig\Modules”该路径中。



本文档还配套的PLCTcComProvider库文件，可以看作PLC的TcCom驱动包，使用该库作为TcCom Object的载体，必须建立一个PLC项目作为TcCom模块的依赖环境（可以单独放在一个PLC项目中，也可以与调用代理功能块放在同一个PLC项目里，取决于用户的实际需求）。

# 库文件TcComProxyApplication

**库文件免责声明：**

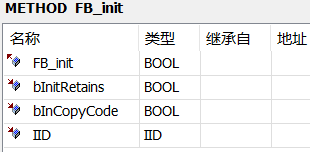
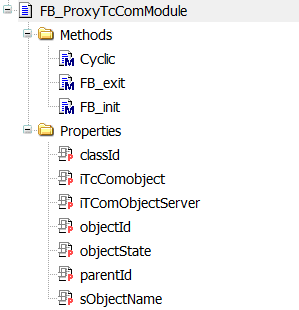
* 使用此版本的库是免费的。
* 客户有责任检查并决定该库和提供的功能是否适用于他的特定应用。
* 该库按原样提供，既不是TwinCAT3的一部分，也不是Beckhoff Automation直接支持的。
* 使用该库是客户自己的风险。
* 不保证该库和与该库一起提供的源代码是完整或准确的。
* 在任何情况下，作者或Beckhoff Automation对由于使用该库而引起的错误和/或遗漏不负任何索赔、损害或其他责任。
* 不能指望对该库的维护、支持、增强或修改，以及对该库或TwinCAT3的更新或与库的更新兼容性的期望权。
* 如有反馈或问题，[请联系yang.yu@beckhoff.com.cn](mailto:请联系yang.yu@beckhoff.com.cn)。
* 倍福中国或库文件作者未经明确授权，禁止复制、分发和利用此库或其部分，以及向他人传达其内容，对其进行未经授权的传播将受到追究赔偿责任。
* 在授予专利、实用新型或设计的情况下，保留所有权利。

该库文件专门中包含了一些函数和功能块对象。函数可以对一些种类的接口、接口指针、PVOID类型指针进行转化。而更重要的是库文件的功能块，在这些功能块中，以及对一些TcCom模块代理的步骤作了内部处理，如果想要创建自定义接口的代理功能块，只需要拓展对应的功能块类型，再做一些简单的处理，即可完成。本章节会对这些功能块进行讲解分析，以及如何在此基础之上进行灵活的运用。该库文件中还有一个引用库文件EventloggerApplication，该库是用来处理TcComProxyApplication中的事件用的。两个库都需要用户在PLC项目中添加。添加方式见入门教程：[TwinCAT 3 自定义库的创建与使用 | 倍福虚拟学院](https://tr.beckhoff.com.cn/mod/hvp/view.php?id=2274)

## 功能块FB\_ProxyTcComModule

1. 功能块说明：

该功能块的作用是与在TwinCAT3中已经存在的TcCom模块做交互，作为目标TcCom的代理。功能块创建完成之后就可以调用目标TcCom模块中的接口。该功能块需要关注的点是FB\_init构造函数，FB\_ProxyTcComModule重载之后的FB\_init增加了一个输入IID，IID（接口标识符，Interface Identifier）是一个唯一的标识符，用于标识特定的接口。在COM（组件对象模型）编程中，IID用于区分不同的接口，以便客户端可以请求和使用特定的接口功能。每个接口都有一个唯一的IID，在TwinCAT里面调用的方式为：TC\_GLOBAL\_IID\_LIST.。

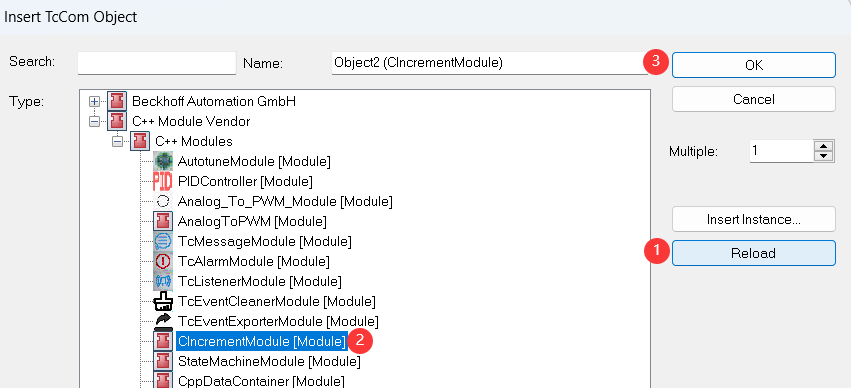


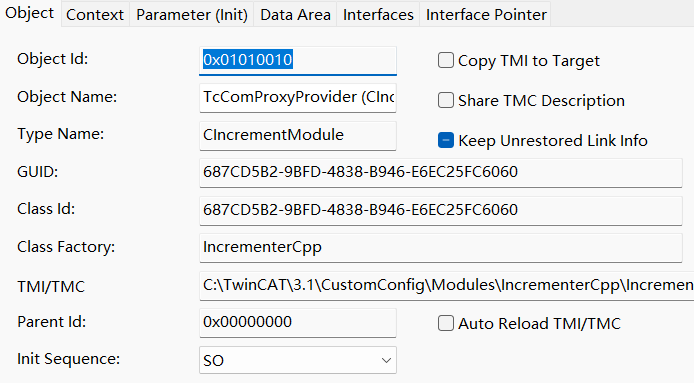
（2）使用方式：

将章节2.1提到的驱动文件放入对应的文件夹之后，在TwinCAT3 System中的TcCom Objects中，添加新项目，选择右侧的Reload，加载TcCom模块。记住该模块的ObjectID，后续会用到：

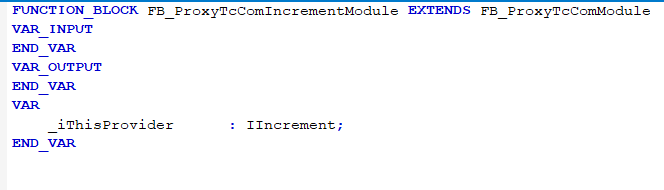
文本

AI 生成的内容可能不正确。





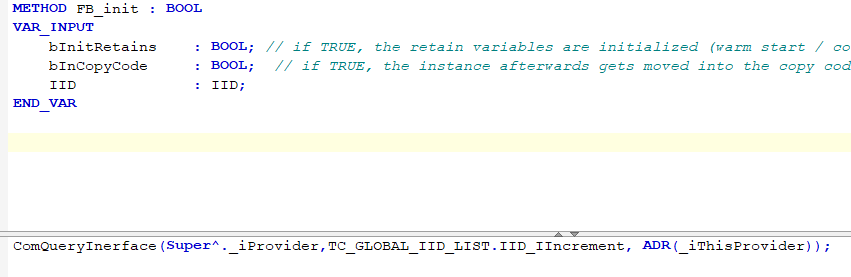
在PLC项目中创建一个功能块，拓展FB\_ProxyTcComModule，并实例化需要实现TcCom模块中的接口：



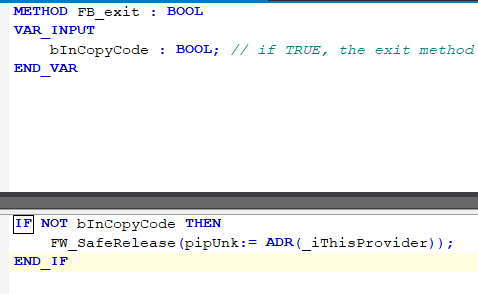
重写FB\_init方法：在方法中，新增函数ComQueryInerface，对父类FB\_ProxyTcComModule中的本地接口\_iProvider与本地的\_iThisProvider进行指定类型的接口转化，例程中，对应的类型就是IIncrement，而由于FB\_init方法的特殊性，隐式自动会调用父类的FB\_init,所以这里可以省略Super^.FB\_init(bInitRetains,bInCopyMode);：

图形用户界面, 应用程序, 表格

AI 生成的内容可能不正确。



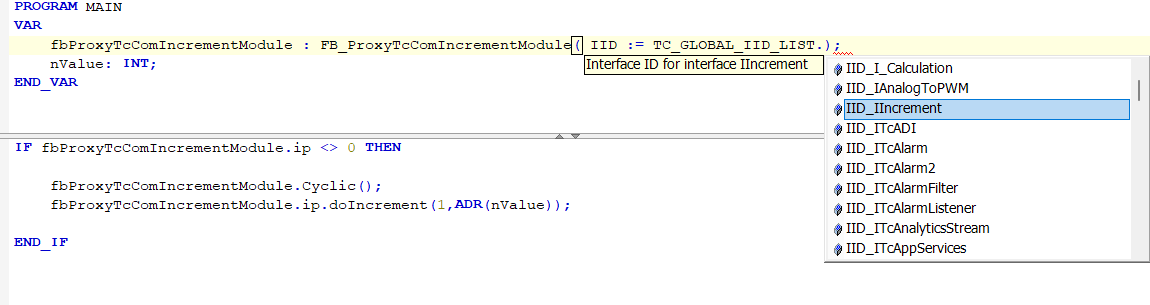
析构函数同理，再FB\_exit中使用FW\_SafeRelease释放新增的接口



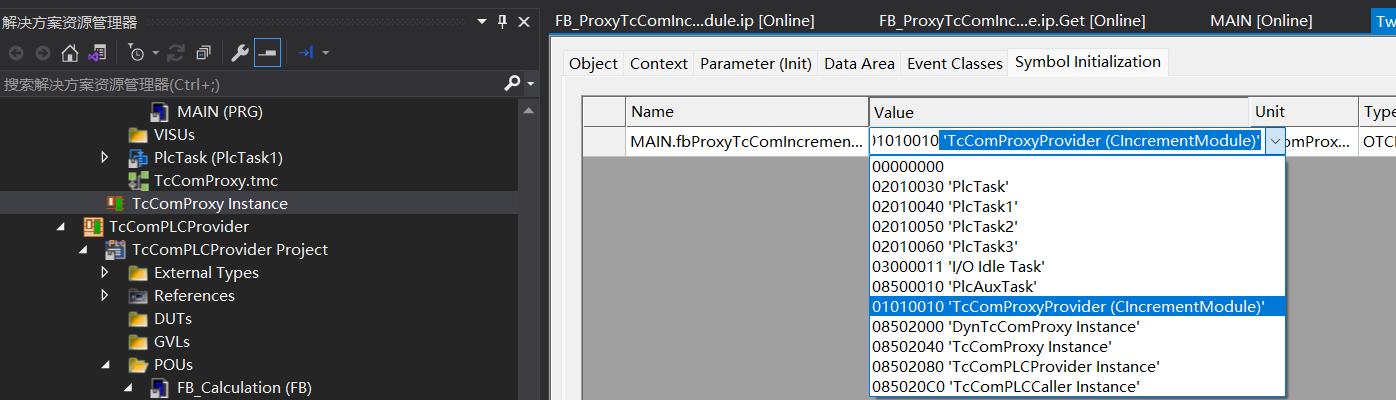
最后再新增一个属性ip，方便在主程序当中调用该代理功能块所执行的接口：



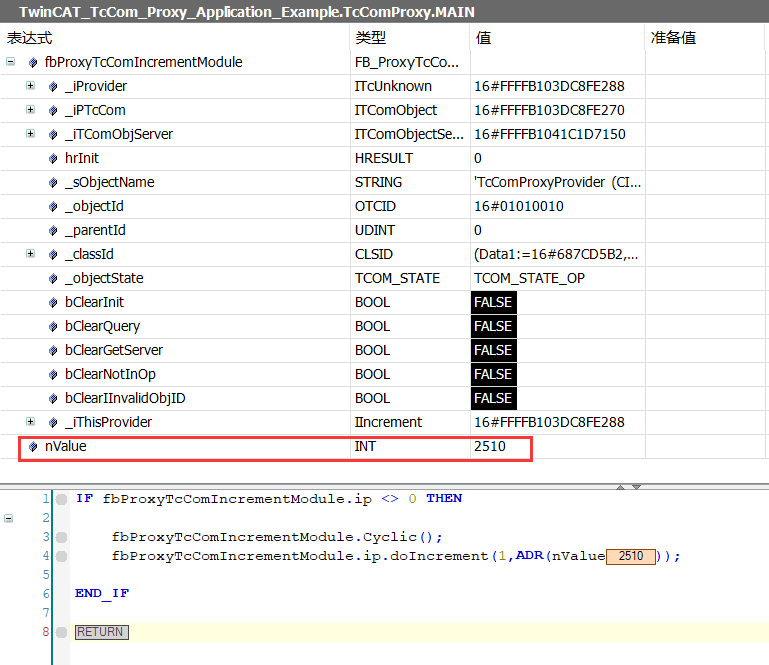
在主程序当中，实例化拓展之后的自定义代理功能块，并在声明的构造函数中填入对应接口ID，程序区域先判断指针是否为0，避免Page Fault，再调用功能块的Cyclic方法和属性接口中的doIncrement方法。填入对应的输入变量：



Build一下解决方案，在PLC的映射实例中，选择Symbol Initialization，在分配的ObjectID Value与TcCom模块的ObjectID进行绑定：



激活配置运行程序，我们会发现，nValue在以每个Task周期1的速度进行自加递增：



我们不妨再观察一下代理功能块内部的的变量：其中，由于\_iProvider和\_iThisProvider通过指令进行了接口转化，即使他们的类型不同，前者为ITcUnknown，后者为iIncrement（后者拓展了前者，TwinCAT所有的接口都是ITcUnknown的衍生物），但他们的值都是相同的，代表指向的是同一个实例。\_iPTcCom对应的是TcCom对象所实现的ITcomObject接口类型，可以获得TcCom对象属性，如\_sObjectName、\_objectId、\_parentId、\_classId、\_objectState。外围程序可以功能块的属性可以进行Get权限的获取这些信息。\_ITComObjServer指向的则是TwinCAT3内部管理所有TcCom模块的服务器实例。由于其功能过于底层，不建议在PLC程序中调用该接口所具备的方法。在该功能块中只是作为一个实例的地址供ITComObject方法中修改TcCom模块的状态机的输入引脚使用：

文本

AI 生成的内容可能不正确。

## 功能块FB\_DynProxyTcComModule

1. 功能块说明：

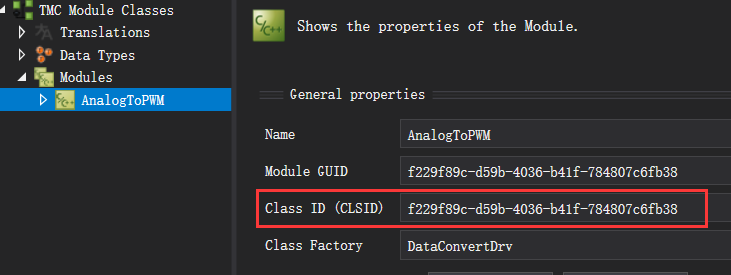
该功能块在构造上和3.1中的FB\_ProxyTcComModule功能块类似，但它是用来处理需要在TwinCAT或者是PLC启动之后，动态创建TcCom模块的需求，该功能块的FB\_init构造函数与前者有所区别，FB\_DynProxyTcComModule构造函数新增了四个输入引脚，classID、IID、sObjectName和eObjectState：

图形用户界面, 应用程序

AI 生成的内容可能不正确。表格

AI 生成的内容可能不正确。

classID可以在TMC或者是TcCom模块的信息中获取：

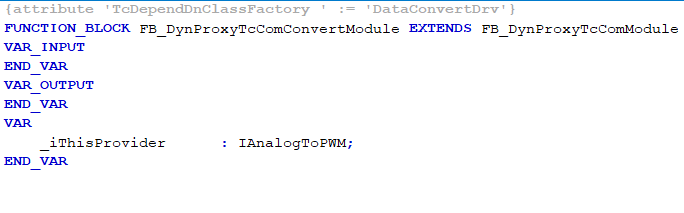


IID可以在TC\_GLOBAL\_IID\_LIST中，全局索引到，sObjectName由用户自定义进行命名，而eObjectState则是在构造函数创建TcCom模块时，将模块启动之后的模式，没有特殊需求，填入TCOM\_STATE.TCOM\_STATE\_OP即可。

1. 使用方式：

在PLC项目中创建一个功能块，拓展FB\_DynProxyTcComModule，并实例化需要实现TcCom模块中的接口，在功能块的抬头中，添加对应类工厂的名称：

（注：即使该功能块可以动态创建模块，但是为了接口类型被识别到，应该先添加一次该接口所在的TcCom模块去预加载TMC文件中的接口，即 Driver的AnalogToPWM Module）

在构造函数中，添加对应接口转化的代码：

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

在析构函数中，添加对应释放接口的代码：

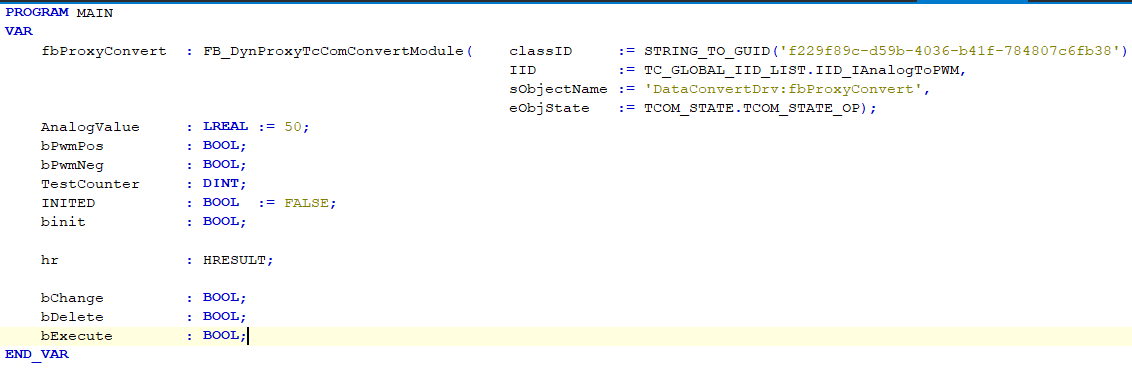
文本, 应用程序

AI 生成的内容可能不正确。

最后再新增一个属性ip，方便在主程序当中调用该代理功能块所执行的接口：

在主程序当中，实例化拓展之后的自定义代理功能块，并在声明的构造函数中填入对应ClssID、IID、sObjectName和eObjState：

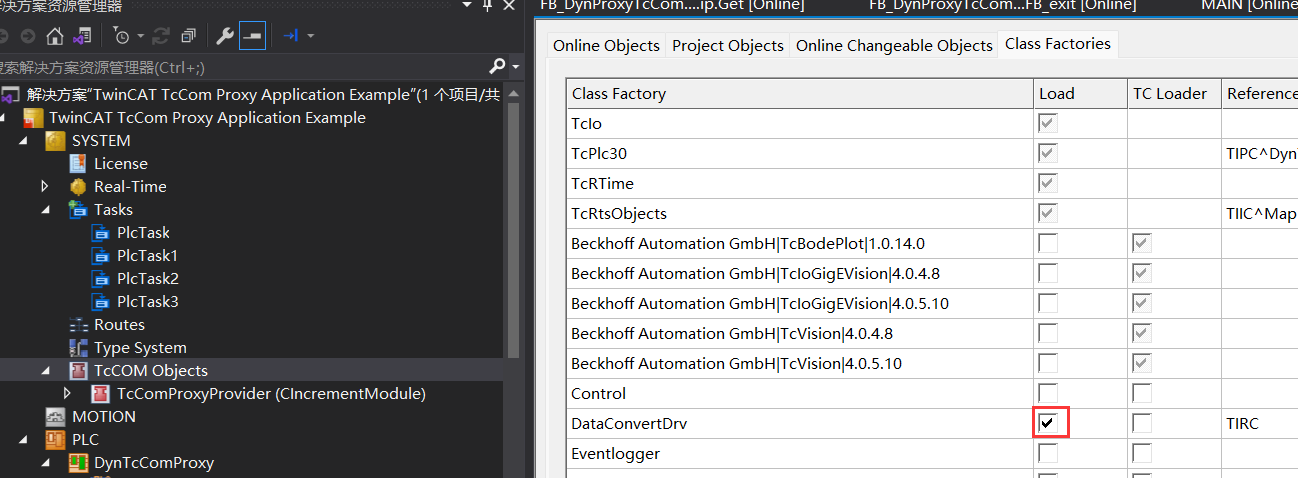


程序区域也是先进行ip非空的，判断，再进行代理功能块中各种接口的调用，包含了初始化PWM周期、数据转化、更改TcCom模块的状态，以及删除创建的TcCom模块：

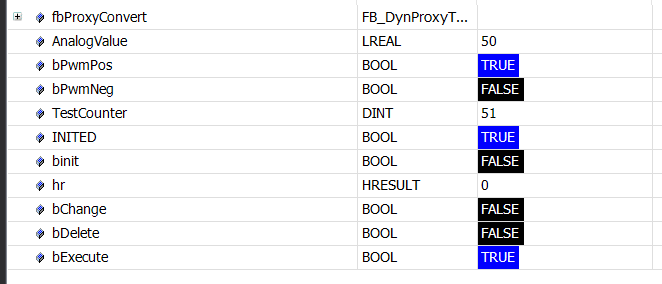
图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

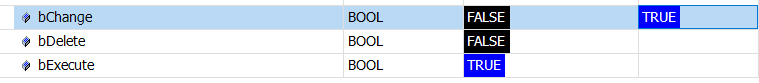
在TcCOM Object的Class Factries中，勾选对应的Load：

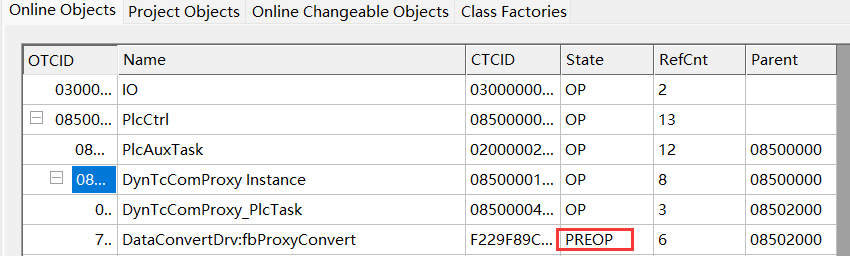


激活配置运行程序，PLC会根据构造函数自动隐式创建一个TcCom模块（在TwinCAT XAE TcCom配置界面无需创建），将bInit置为TRUE，之后设置PWM周期为1s，再将bExecute置为TRUE，接口方法就会调用，根据输入的模拟量值50，会将bPwmPos以0.5s TRUE，0.5s FALSE的规律进行交互变化：

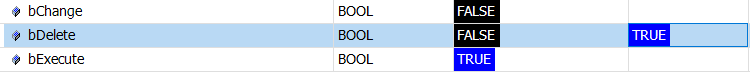


如果将bChange置为TRUE，模块会切换至PREOP状态，由于在TcCom模块的C++代码中，进行了非OP状态不执行该方法的判断，所以bPwmPos不会再改变：





如果将bDelete置为TRUE，则代理功能块可以动态地释放所有的接口指针，由于在TwinCAT3中。接口指针都是一种智能指针，会根据RefCnt为0，自动删除该TcCom模块的实例。同时在Online Objects中，也看不到原来的对象了。



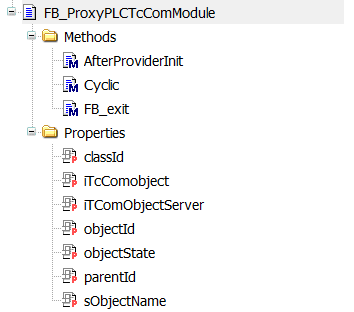
图形用户界面, 应用程序

AI 生成的内容可能不正确。

## 功能块FB\_ProxyPLCTcComModule

1. 功能块说明：

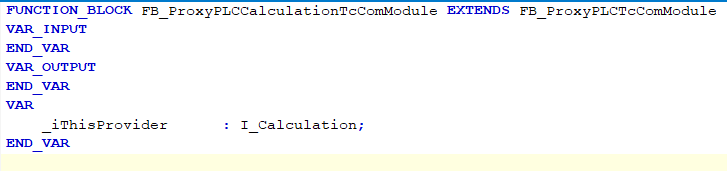
该功能块在构造上和前两者有所区别，并没有用到构造函数。取而代之的是方法AfterProviderInit。该功能块是用来在代理功能块所处的PLC模块中，调用另一个PLC模块所提供的接口实例。那由于在TwinCAT3运行之后，PLC模块被创建，不能保证先后顺序（比如提供接口的PLC还没有生成对应的ID，后者的构造函数就已经执行，那就会出现无效ObjectID），所以必须手动或者依靠程序进行代理功能块的初始化，输入变量仅有一个IID：

 表格

AI 生成的内容可能不正确。

1. 使用方式：

在PLC项目中创建一个功能块，拓展FB\_ProxyPLCTcComModule，并实例化需要实现TcCom模块中的接口：



重写AfterProviderInit方法：

图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。

新增一个属性ip，方便在主程序当中调用该代理功能块所执行的接口：



（注：I\_Calculation接口不是通过C++实现的，所以不需要驱动文件，但是需要该TwinCAT项目中引用过PLCTcComProvider库文件，如果在同一个TwinCAT项目，不需要依赖库文件）

实例化拓展之后的自定义代理功能块，用程序进行AfterProviderInit的调用，填入IID，判断接口非0的情况下，调用Cyclic方法和加减方法即可。Build解决方案，将实现对象功能块所在的PLC中实例的输出ObjID绑定到代理功能块所在的PLC中实例的输入ObjID：

文本

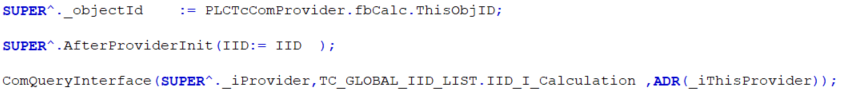
AI 生成的内容可能不正确。

图形用户界面, 文本, 应用程序

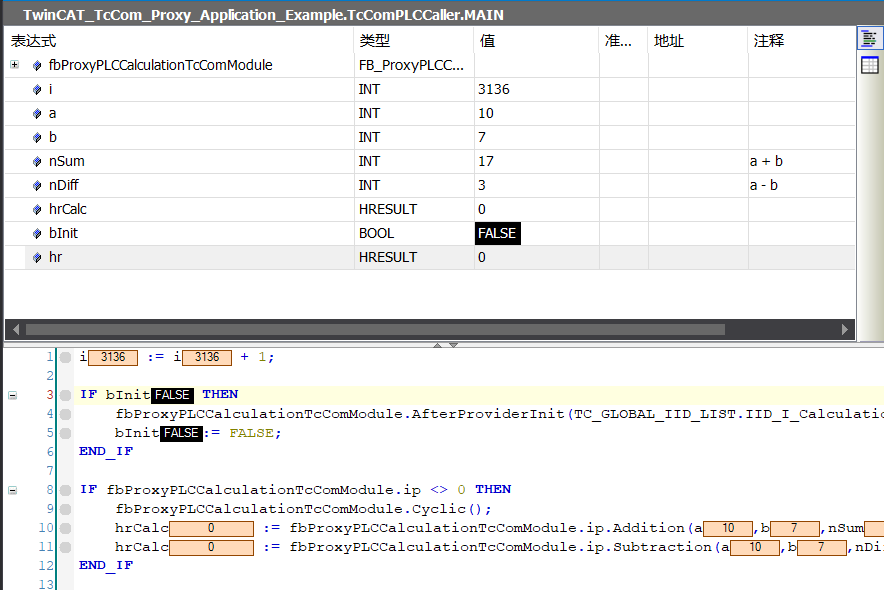
AI 生成的内容可能不正确。文本

AI 生成的内容可能不正确。

如果使用库文件的方式且在同一个PLC项目，那就无需理会输入输出绑定，直接在功能块中，在AfterProviderInit方法中调用库文件全局变量中TcCom功能块实例的属性ThisObjID即可：



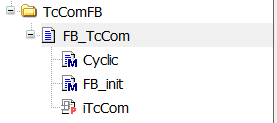
（4）激活配置运行程序，确保TcCom模块所对应的功能块所在的PLC程序运行，以及代理功能块所在的PLC运行，将bInit置为TRUE，ip会因为初始化自动指向实例，运行Cyclic方法，以及对应接口的加减法方法：



# 库文件其他函数和功能块

## 功能块FB\_TcCom

该功能块代表一个TcCom模块实例本身，它的原理是通过构造函数中的ipInterface接口指针传递到功能块当中再通过函数PVOID\_TO\_ITcomObject将接口指针转化为ITcomObject接口类型给到本地的接口\_iTcCom，这样该功能块就可以通过属性iTcCom对代表的TcCom模块进行一些属性的读取甚至修改了。

 表格

AI 生成的内容可能不正确。

## 函数PVOID\_TO\_ITcomObject

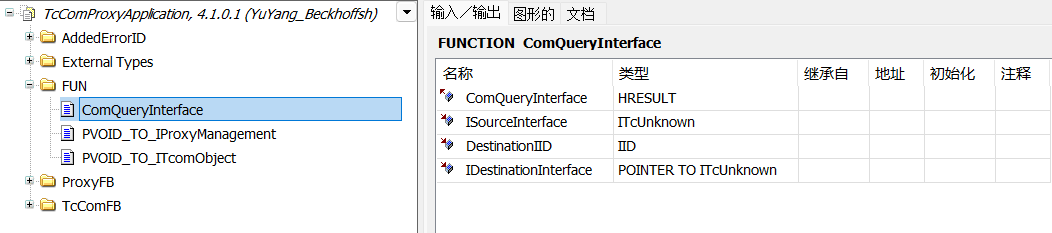
该函数可以将一个PVOID任意类型的指针（此处代表任意类型接口的指针）转化为接口I\_TcomObject接口类型：

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

## 函数ComQueryInterface

该函数是一个用于查询接口的函数。它的主要功能是将一个源接口转换为目标接口，目标接口类型为IID所代表的接口。目标接口通过ITcUnknown接口指针类型传递



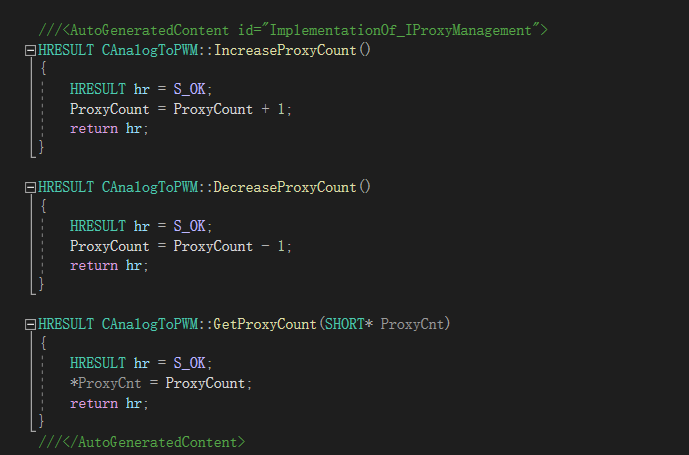
## 接口IProxyManagement

该接口是库文件4.0之后在内部拥有的一个接口类型，在附件的TcCom模块IAnalogToPwm中，实现了这个接口，该接口的作用是对TcCom Object的代理数进行管理。如果一个TcCom Object实例被多个功能块进行代理，那会出现冲突，和数据不一致的问题，实现了该接口的TcCom Object能更好地配合功能块对其代理进行管理。

文本

AI 生成的内容可能不正确。

库文件中，功能块FB\_ProxyTcComModule以及FB\_ProxyPLCTcComModule会在代理初始化之后，检查代理的TcCom Object是否实现了该接口，如果实现了该接口，那就会执行IncreaseProxyCount，TcCom Object的代理数就会+1；如果删除了代理的TcCom Object，对应代执行DecreaseProxyCount，那么TcCom Object的代理数就会-1。同时，两者会在循环方法中，通过GetProxyCount获取当前的代理目标TcCom Object的代理数：



一旦大于1的话，EventLogger会发送一个报警：



# 库文件Event解析

1. CreateTcComFailure：FB\_DynProxyTcComModule功能块创建TcCom模块失败
2. QueryInterfaceTcomFailure：代理功能块所代理的对象未实现接口ITcomObject。
3. GetTcComServerFailure：获取当前TwinCAT的TcCom服务器接口指针失败。
4. TcComObjectNotInOP：当前代理的TcCom对象不在OP状态。
5. InvalidObjectID：代理功能块没有填写参数ObjectID（TcCom对象实例的ID）。
6. InvalidClassID：代理功能块没有填写参数ClassID（TcCom对象的类工厂ID）。
7. InvalidIID：代理功能块没有填写参数IID（TcCom对象实现的接口ID）。
8. EmptyInterface：使用函数ComQueryInerface没有填入有效的接口（空接口）。
9. TcComProxyCountWarning：当前的TcCom对象正在被超过一个功能块代理。
10. TcComPLCModuleActive：提示PLC TcCom对象被创建在程序中，不要使用断点功能。
11. QueryInterfaceProxyManagementFailure：代理功能块所代理的对象未实现接口IProxyManagement。
12. GetTcComFailure：FB\_ProxyTcComModule/FB\_ProxyPLCTcComModule功能块获取TcCom模块实例失败。
13. CyclicMethodInvalid：FB\_ProxyPLCTcComModule功能块没有执行过AfterProviderInit方法，因此Cyclic方法是无效的。

# 常见问题

Q：如果需要在PLC中多次调用一个类型的接口咋办？

A：以AnalogToPWM为例，如果一个模拟量是30，一个模拟量是50，调用了一个TcCom对象的同一个接口，会导致数据紊乱，有此类需求，建议创建两个TcCom实例分别调用。

Q：使用PLC功能块生成TcCom Object直接在调用代理功能块的项目中填入ID值不行吗？

A：如果提前记下PLC功能块生成的TcCom Object ID值，再后一次激活前使用代码填入，库文件中TcCom功能块自动生成的TcCom Object ID会避开这个值，无法进行同步。因为TwinCAT中的OTCID类型变量的值是不会重复的。只有两种方法，针对两个PLC项目，通过输入输出绑定；针对一个PLC项目，通过全局变量调用实例的属性。

Q：在PLC功能块作为实现接口的TcCom对象内部加入断点导致蓝屏

A：绝对不可以在实现接口的的TcCom对象功能块中加入断点，会导致线程的不安全，进一步导致系统驱动层崩溃，具体细节未知。使用编译过的库文件的话就更安全一点，因为编译过的库文件内部代码是无法加入断点的

**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32号利通广场1303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 室

电话: 028-86202581 邮箱: chengdu@beckhoff.com.cn

|  |  |
| --- | --- |
| 请用微信扫描二维码  通过公众号与技术支持交流 | 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 |
|  |