**TwinCAT 3 PLC通过接口调用C++的函数**

|  |  |
| --- | --- |
|  | 作者：余洋  职务：技术支持工程师  邮箱：yang.yu@beckhoff.com.cn  日期：2023-08-07 |
| **摘 要：**  TwinCAT3中不同的模块（如PLC、C++、Matlab等等）可以通过TcCom以不同的方式交互从而完成整个项目对于高精度算法以及实时性的需求。一般交互的方式分为四种：IO变量直接映射、接口调用、IO数据指针以及ADS通讯。本文将在C++中创建一个简单的状态机，其他模块（PLC）可以从外部启动和停止该状态机，也可以设置或读取 C++状态机的特定状态。交互的方式通过接口来进行。 | |
| **附 件：**   |  |  |  | | --- | --- | --- | | 序 号 | 文件名 | 备注 | | 1 | TwinCAT CPP Interface Test | 例程 | |  |  |  | |  |  |  | |  |  |  | | |
| **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](#_Toc148305597)

[1.1. 控制硬件 3](#_Toc148305598)

[1.2. 控制软件 3](#_Toc148305599)

[2. 准备工作 3](#_Toc148305600)

[3. C++ TMC编辑器配置操作 3](#_Toc148305601)

[4. 修改C++ 头文件（.h）以及CPP文件代码 9](#_Toc148305602)

[5. PLC调用C++中的接口 12](#_Toc148305603)

# 软硬件版本

## 控制硬件

1. 测试电脑（Windows11专业版）

## 控制软件

1. TwinCAT 3（版本为4024.47，嵌入在Visual Studio2019中进行开发）
2. Visual Studio C++开发相关组件

# 准备工作

TwinCAT3 安装在 64 位操作系统上的时候，开发并运行 PLC 没什么问题，但如果要开发并运行 matlab-simulink 或者 C++则需要安装测试证书，具体操作步骤详见文档《[TwinCAT3 C++使用准备教程 V3.11](https://tr.beckhoff.com.cn/mod/resource/view.php?id=2710)》或者使用应用程序“[一键环境变量设置和64位数字签名](https://tr.beckhoff.com.cn/mod/resource/view.php?id=2711)”。

# C++ TMC编辑器配置操作

（1）首先新建一个TwinCAT Project：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

（2）新建一个C++ Project：

图形用户界面, 文本, 应用程序

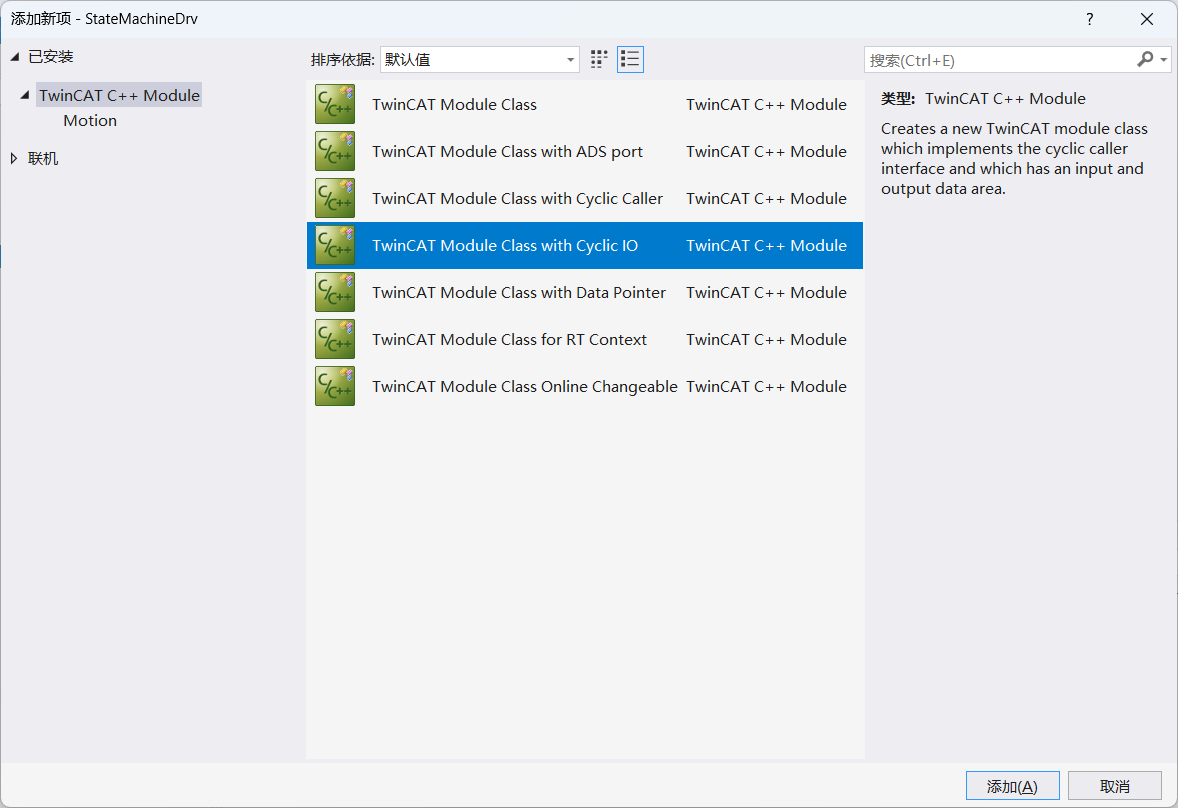
描述已自动生成

（3）选择“TwinCAT Driver Project”，输入项目名称，并单击Add：

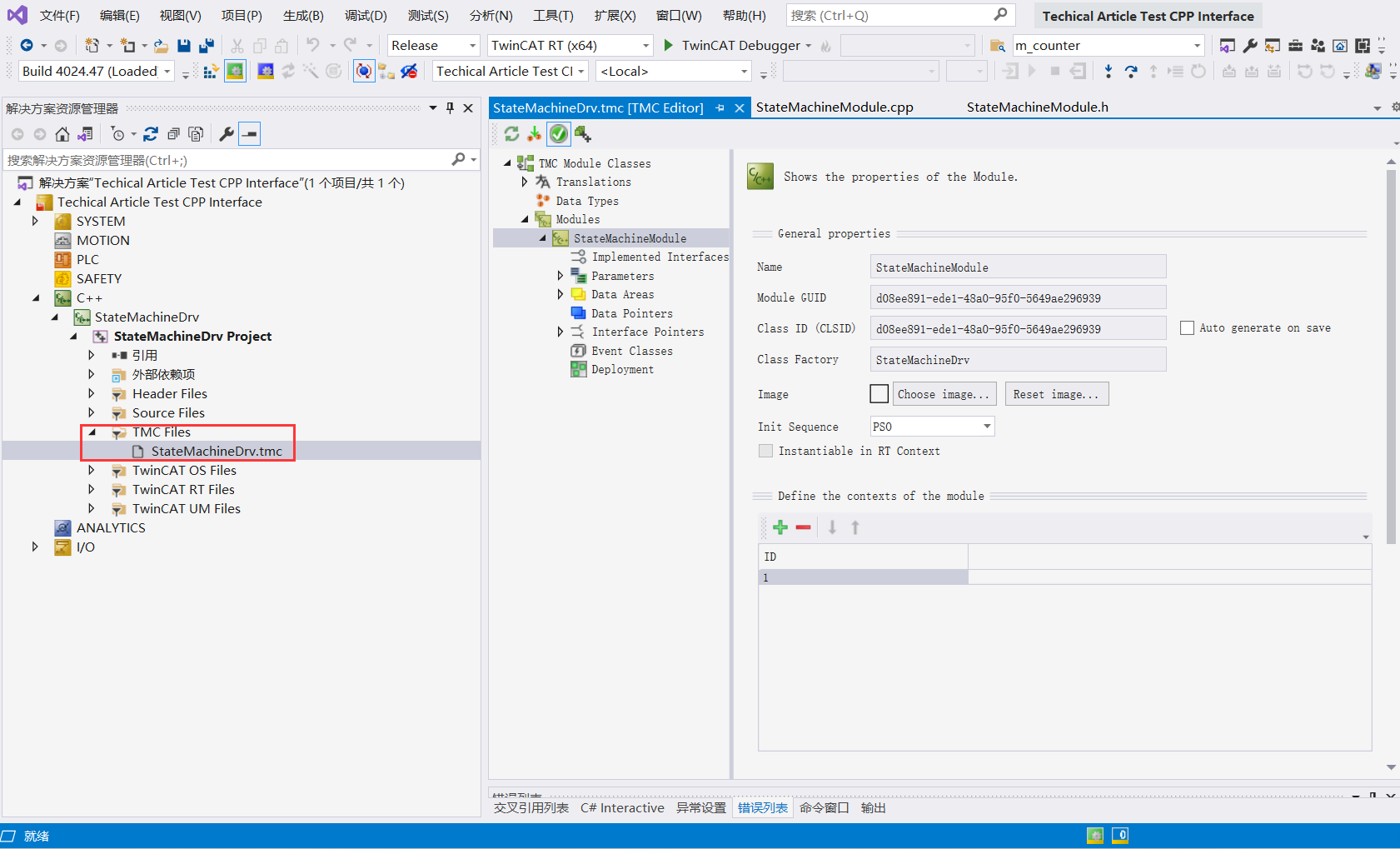
图形用户界面, 文本, 应用程序

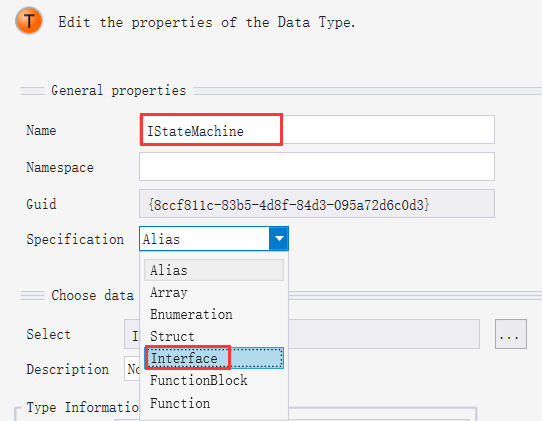
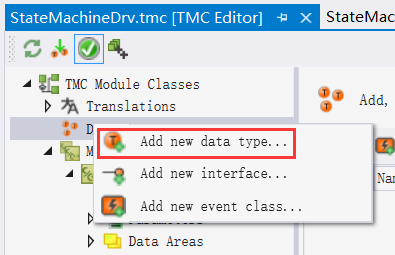
描述已自动生成

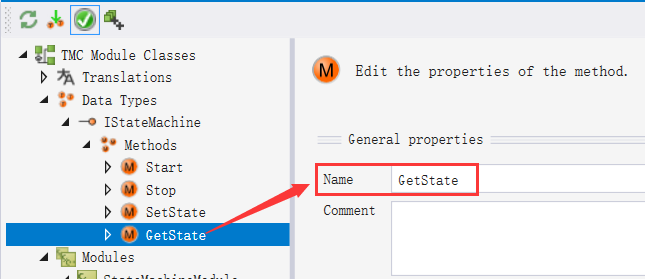
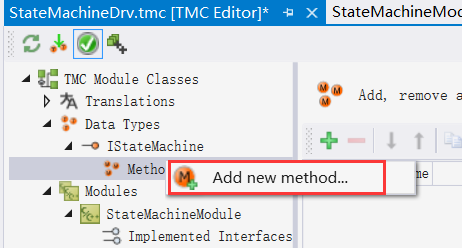
（4）选择用于此驱动项目的模块模板，此例程中，选择“TwinCAT Module Class With Cyclic IO”：



（5）创建完模块之后进入TMC编辑器：



（6）选中Data Type右键选择“Add new data type…”，在跳出的窗口中编辑名称并选择接口：

（7）在创建的接口中右键新建四个Method，分别是Start、Stop、SetState、GetState：

（8）展开SetState下方的Parameters，选择➕创建一个新的参数：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

（9）编辑参数名称，类型选择位普通INT类型，描述为普通种类：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

（10）同理，在GetState方法中新建一个参数，名字为pState，描述设置为指针：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

（11）在底下的模块中，选择Implemented Interfaces，在右侧选择➕：

图形用户界面, 文本, 应用程序

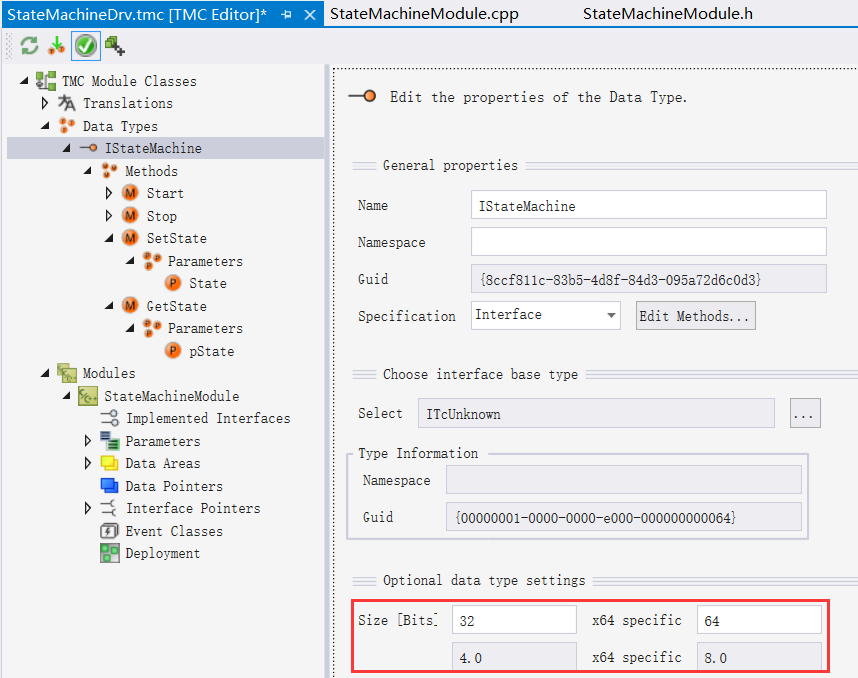
描述已自动生成

（12）选择本地新增的接口，并观察其他系统接口的大小：

图形用户界面, 表格

描述已自动生成

（12）观察到系统接口的大小为4（8），重新回到接口编辑页面进行大小的修改，确保一致：

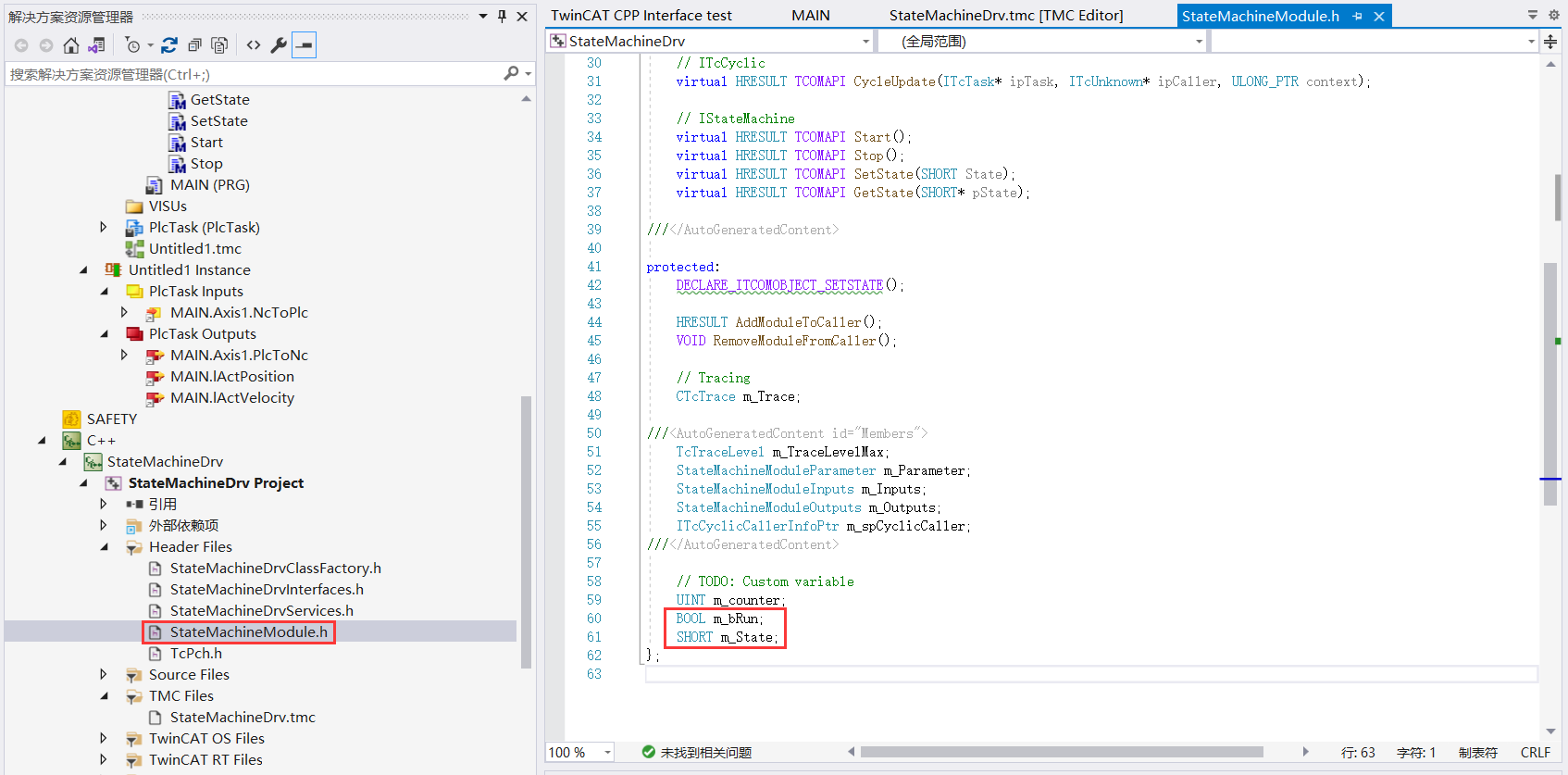


（13）右键C++程序项目，选择TMC Code Generator自动生成配置的代码：

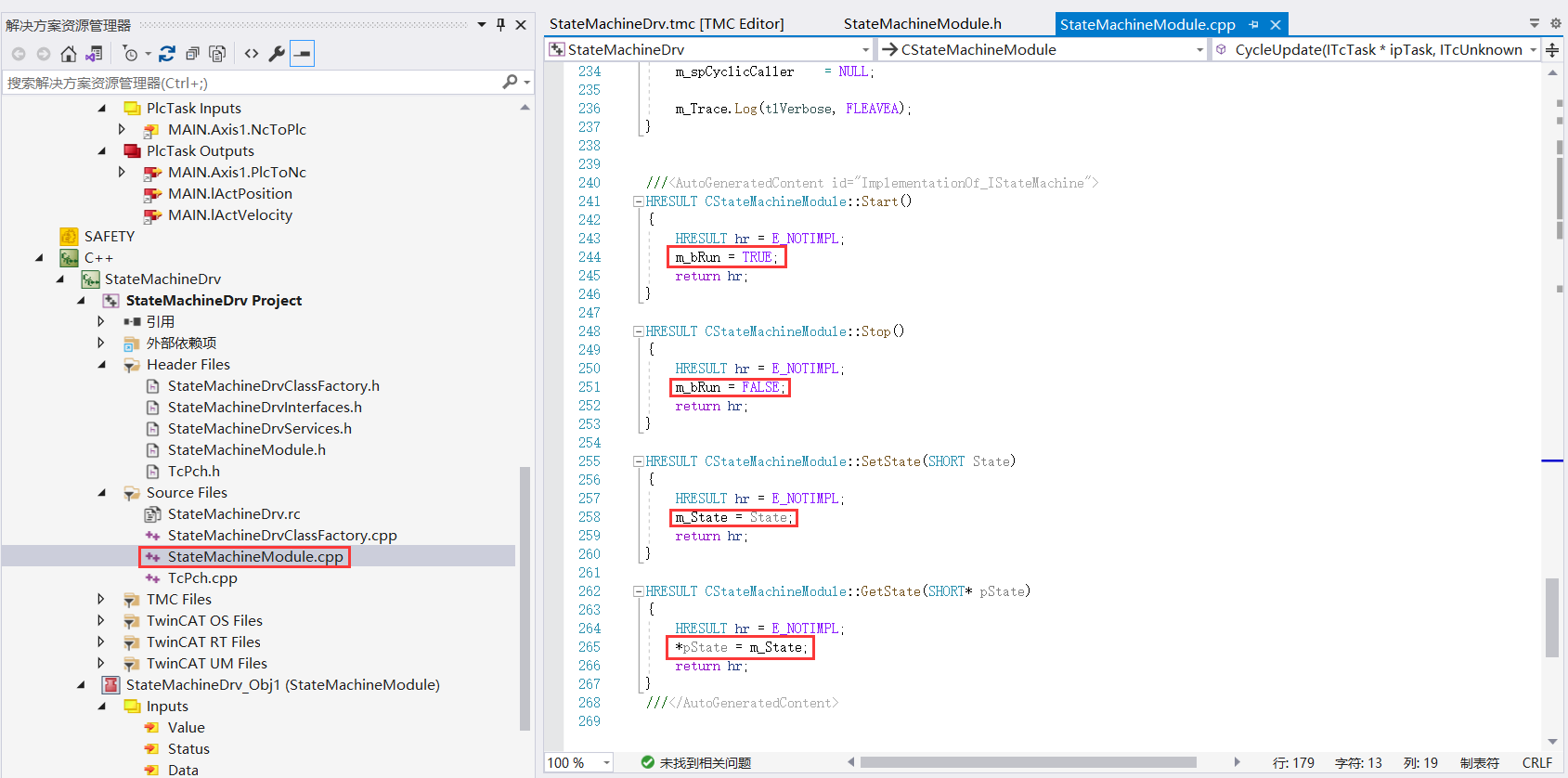


# 修改C++ 头文件（.h）以及CPP文件代码

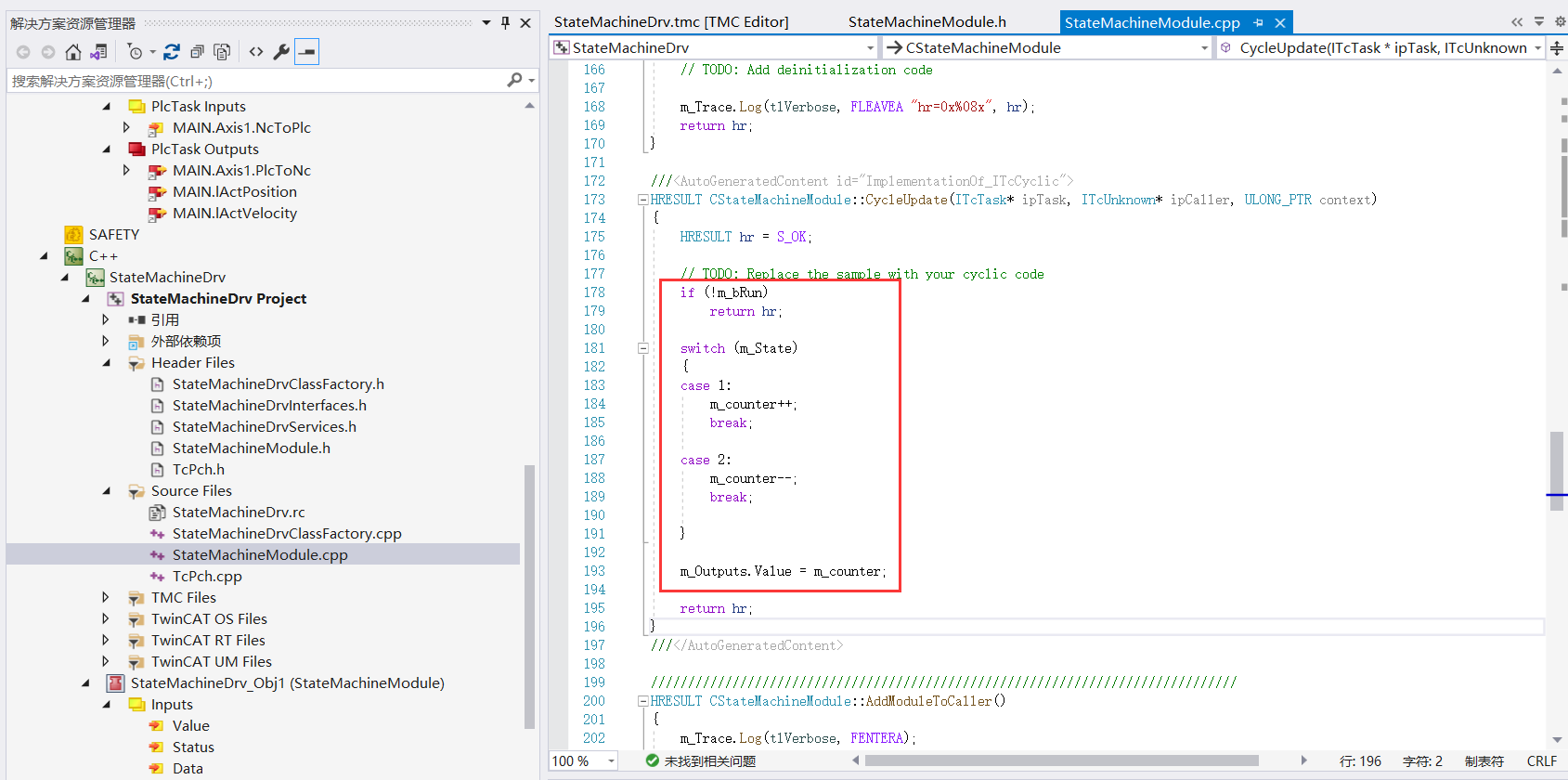
（1）将配置中的变量添加到头文件 StateMachineModule.h：



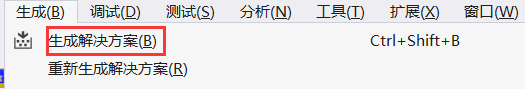
（2）进入CPP文件实现状态机模块四个方法的代码：



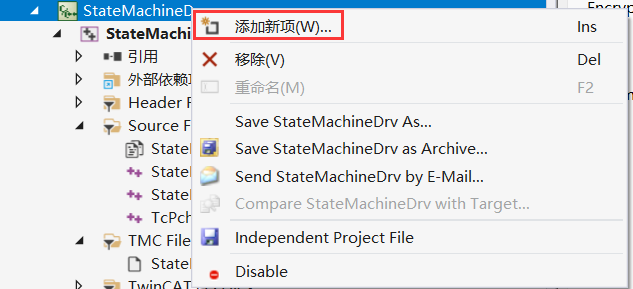
（3）在CPP文件中，编辑循环更新的函数，即如果状态为1，则递增计数器；状态为2，则递减计数器。同时，将生成的计数器的值分配给Value：



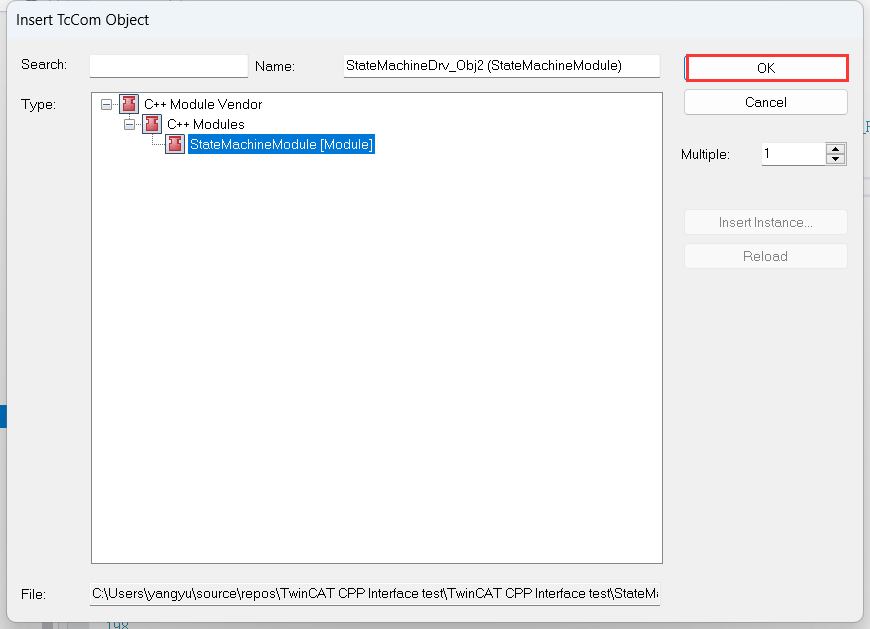
（4）生成（重新生成）解决方案：

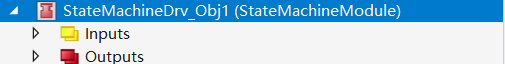


（5）编译无报错之后右键C++项目，选择Add New Item：

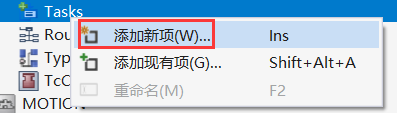


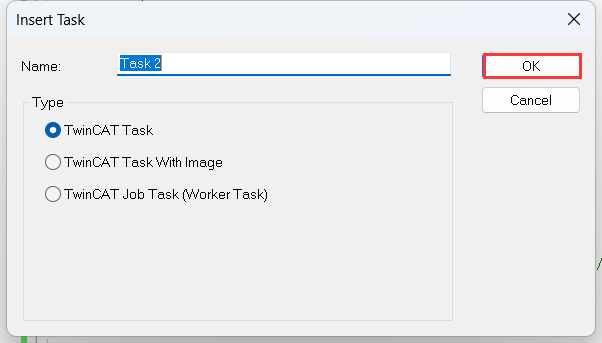
（6）选择对应的项目实例，资源管理器中就会出现对应C++项目的实例：



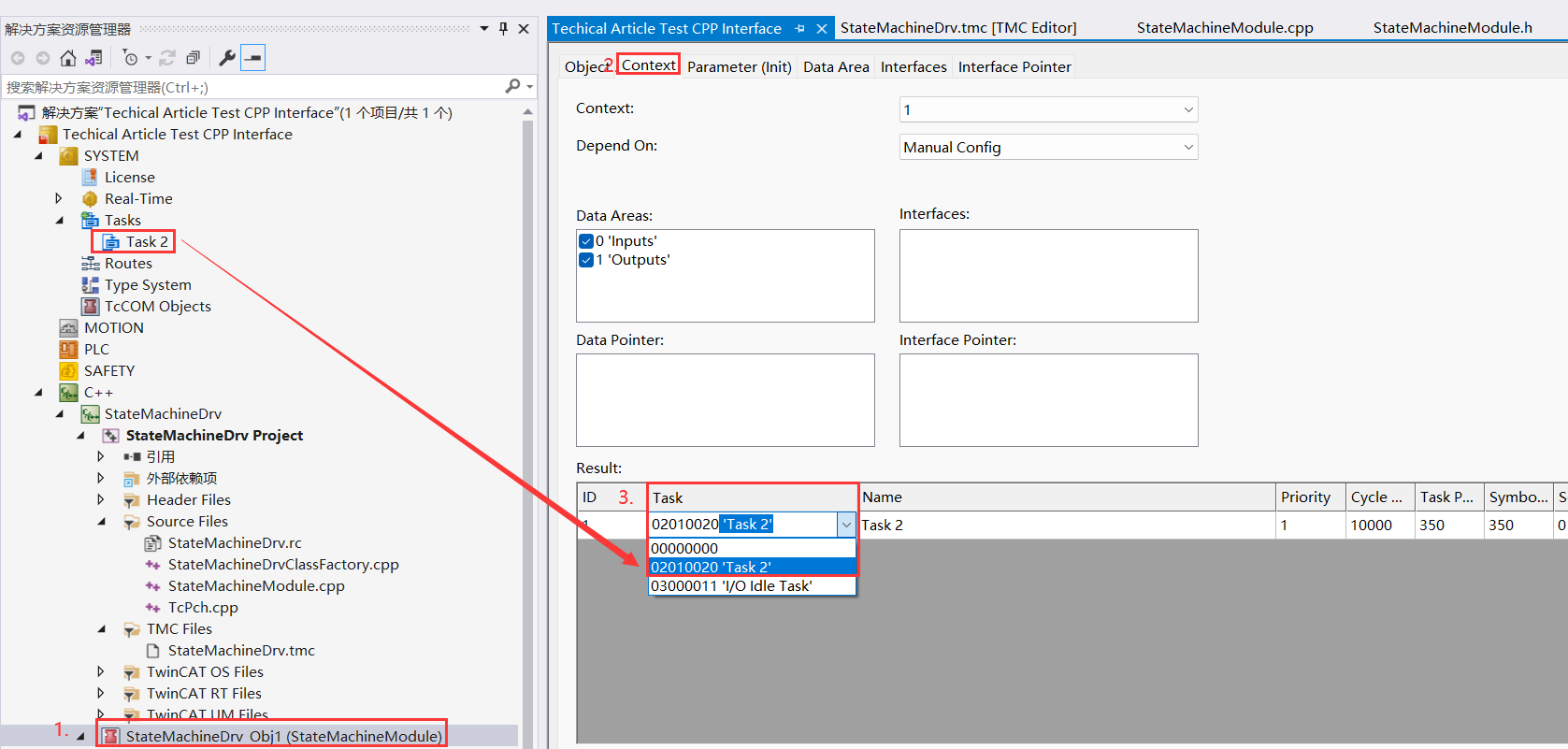


（7）右键System下方的Task，选择Add New Item：

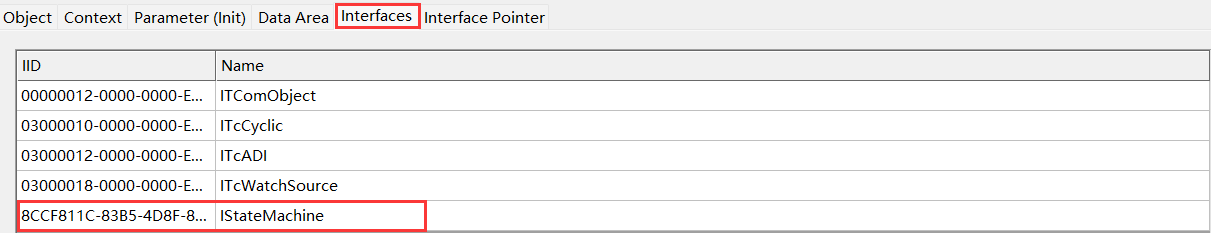




（8）在C++模块的实例处，选择Context选项卡，然后分配对应的Task（2）：

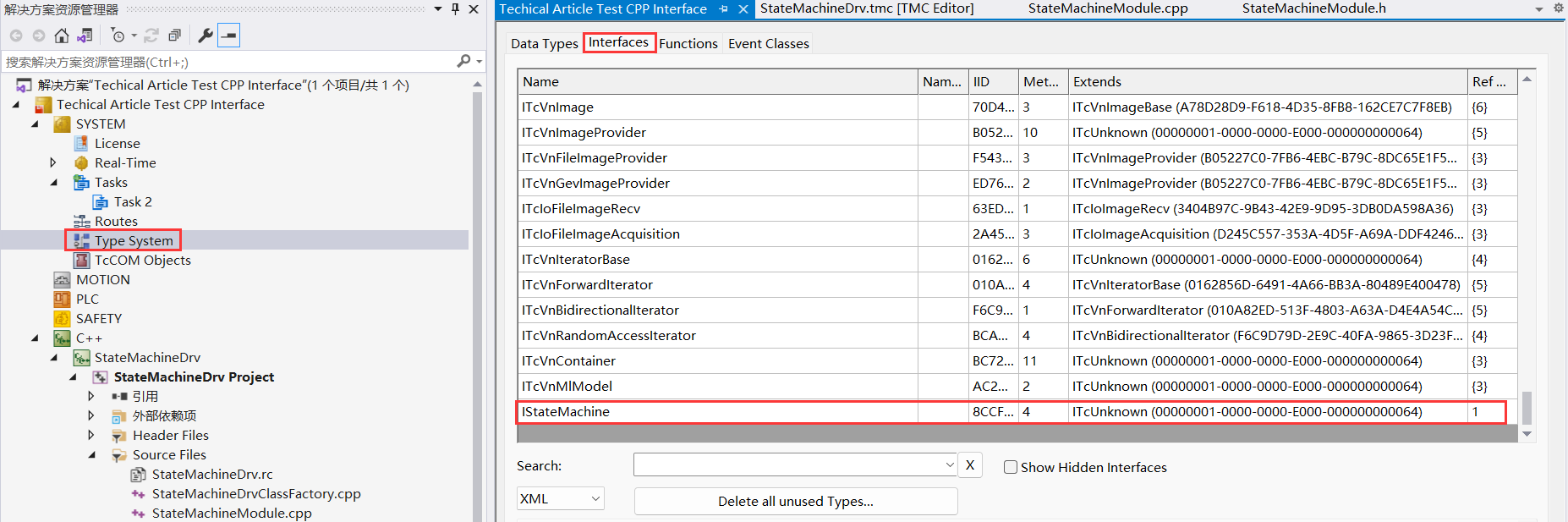


（9）检查实例的Interfaces选项卡，查看是否存在对应的自定义接口：

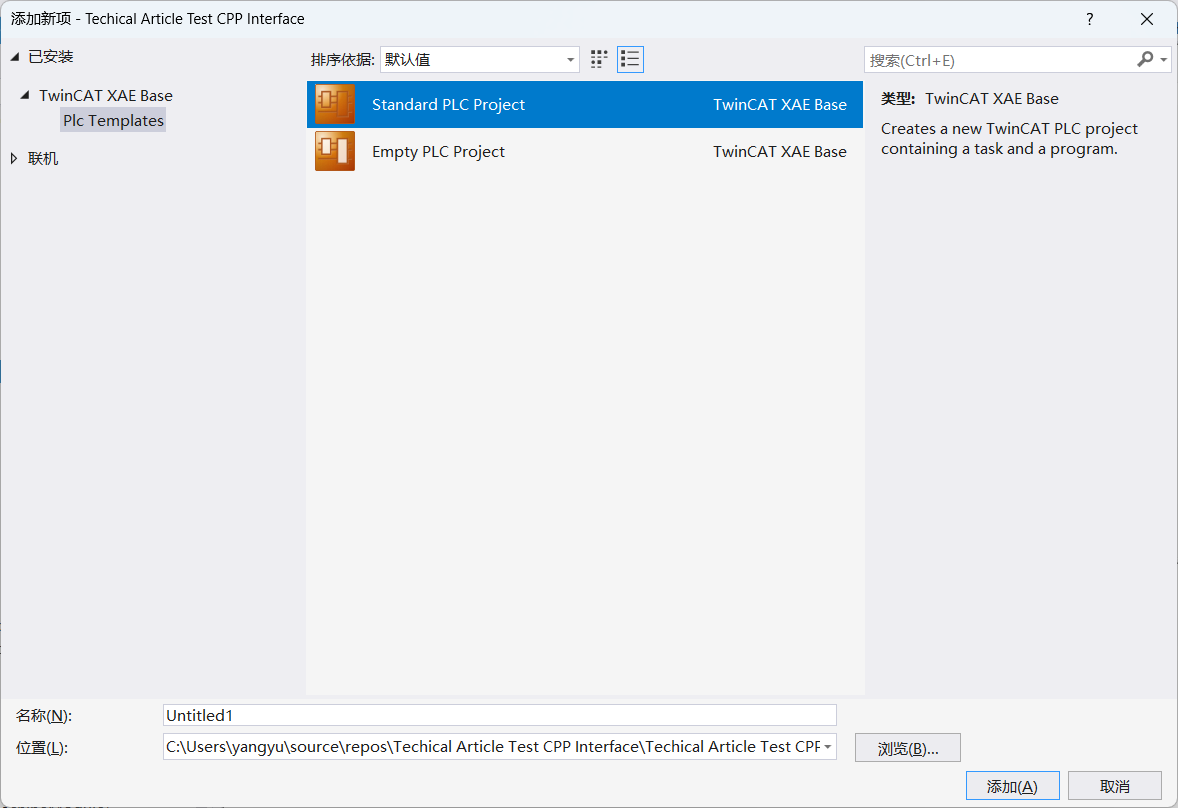


# PLC调用C++中的接口

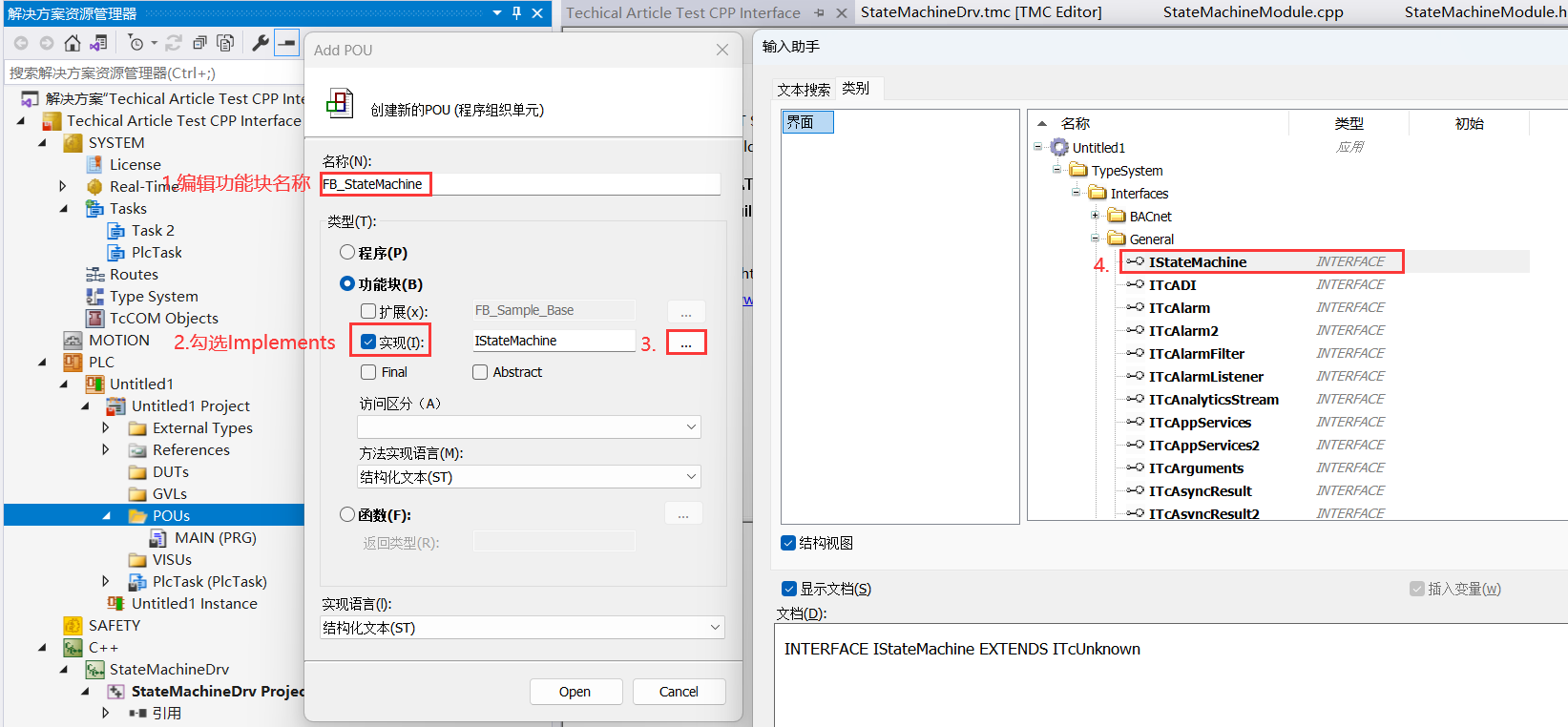
（1）在System下方的Type System处选择Interfaces选项卡查看系统接口中是否存在C++中创建的自定义接口IStateMachine：



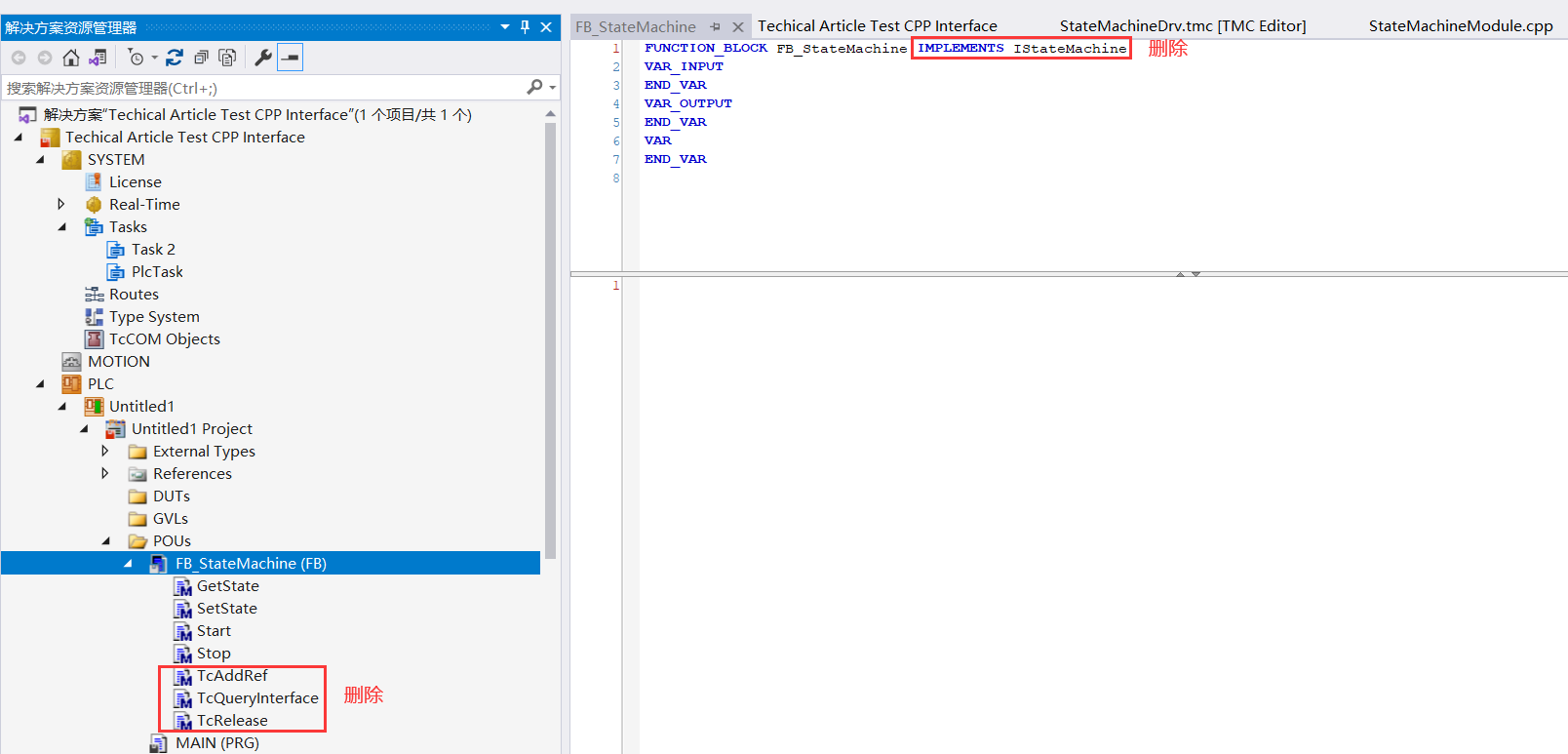
（2）新建一个PLC项目：



（3）在POUS文件夹中创建一个功能块FB\_StateMachine，在功能块编辑窗口中勾选Implements，右侧选择实现接口IStateMachine：



（4）创建完成之后的功能块会自动生成关键字Implement IStateMachine，由于功能块充当代理，本身不具体实现接口的具体方法（而是在C++中），因此，删除Implement IStateMachine。同时功能块会自动继承ITcUnknown父类中的方法TcAddRef（计数器的显式方法，与模块寿命有关）、TcQueryInterface（通过模块ID调用其中的方法） 和 TcRelease（递减计计数器并返回值，如果计数器为0，则对象自动删除），但此功能块本身却不需要这些方法，因此，全部删去：



（5）分别在功能块中创建两个方法FB\_init和FB\_exit：

图形用户界面, 应用程序

描述已自动生成

（6）修改FB\_StateMachine、FB\_init和FB\_exit中的代码：

首先，修改功能块的变量声明区，在中间变量区域先创建属性{attribute 'TcInitSymbol'}，目的是为了能够在实例中直接将代理功能块与对应C++模块中的接口ID关联，如果不编写此属性，则无法进入关联的界面。ipStateMachine : IStateMachine;则用于创建代理功能块的ID。最后需要创建一个HERSULT类型的函数返回值，用来监控函数处理状态：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

接着，编辑方法FB\_init，该方法创建完会默认新增两个输入变量，分别是bInitRetains以及bInCopyCode，代表TwinCAT软冷热启动以及online change状态（激活硬件配置启动程序之后默认情况下对应的显式调用格式为FB\_init(bInitRetains := TRUE, bInCopyCode:=FALSE );）。在FB\_init方法中，需要编写相关程序，当程序不处于在线更改模式下（即刚上电），判断接口值是否为0，如果是，则调用FW\_ObjMgr\_GetObjectInstance，将代理功能块实例与对应C++模块中的接口实例配对：

图形用户界面, 文本, 应用程序

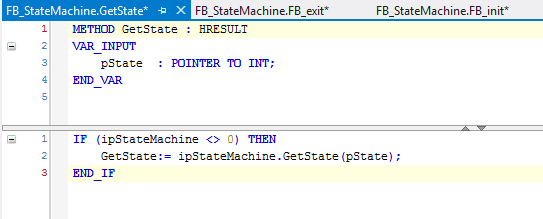
描述已自动生成

然后，编辑FB\_exit方法，当退出online change模式之后，使用函数FW\_SafeRelease引用递减计数器，释放内存资源，并将接口指针置为0：

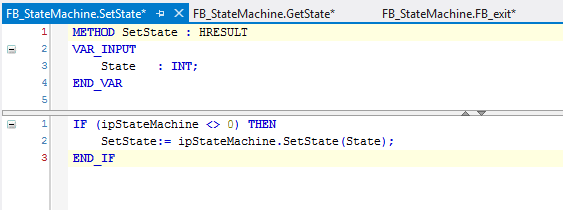
图形用户界面, 文本, 应用程序, 电子邮件

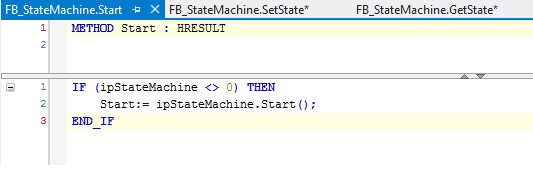
描述已自动生成

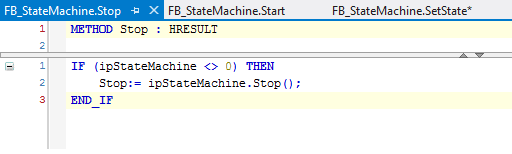
除此之外，编辑功能块继承接口中自定义的方法，由于C++中GetState函数返回的值会回到调用者，作为一个参数来传递，所以在C++中当时就设置的是一个指针pState，同理，在GetState方法中，也要设置为INT类型的指针，并且在程序区域判断接口是否等于0。如果不是，则能够调用接口中的方法，因此进行调用，而其中的参数即为pState：



同理，编写SetState、Start、Stop方法。（这里SetState的State只参与运算，不用作为参数传递，因此，可以设置为普通的INT类型变量）

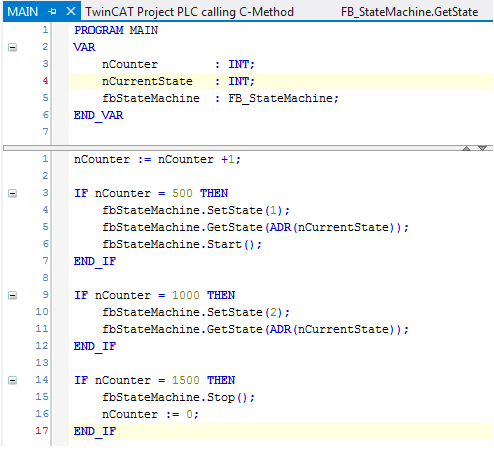




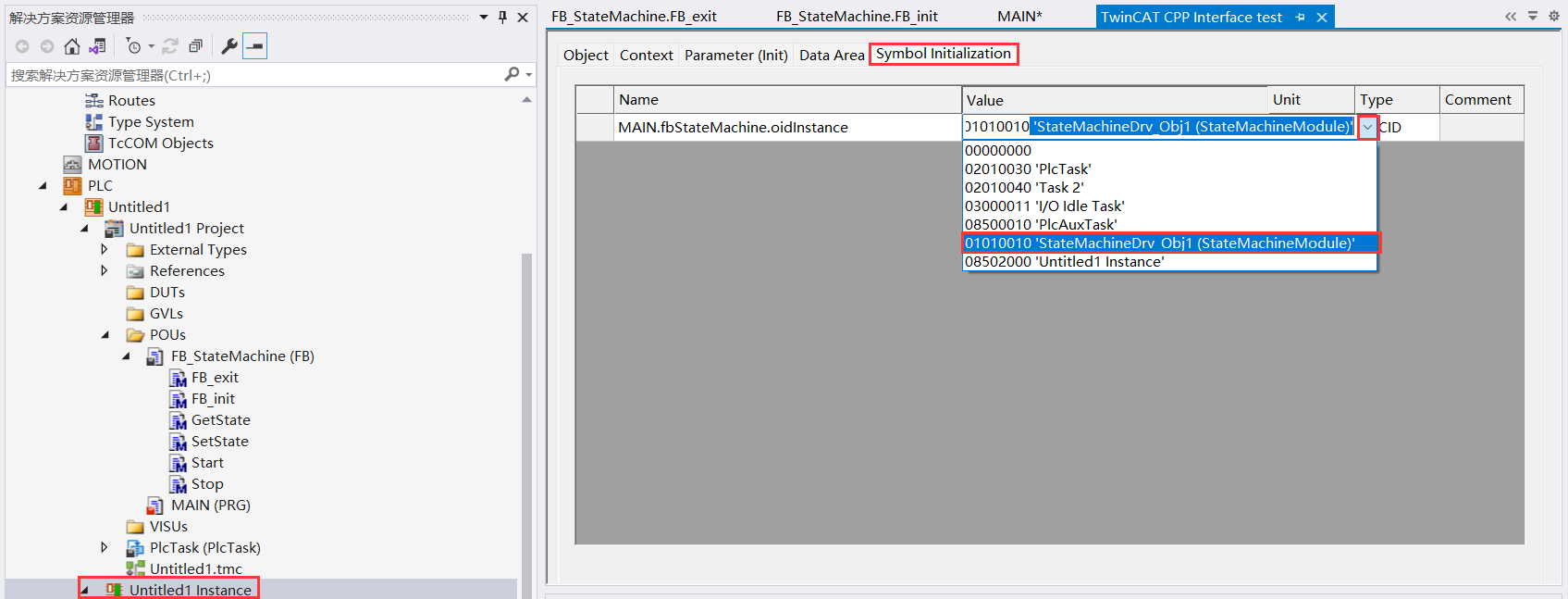


最后，在主程序中调用功能块FB\_SateMachine，声明三个变量nCounter、nCurrentState和功能块实例fbStateMachine：

* PLC 计数器 nCounter计数器的循环递增
* 如果 nCounter = 500，则C++状态机以状态“1”启动，以递增其内部C++计数器。然后使用 GetState（ADR（nCurrentState）） 读取C++的状态。
* 如果 nCounter = 1000，则C++状态机设置为状态“2”，以递减其内部C++计数器。然后使用 GetState（ADR（nCurrentState）） 读取C++的状态。
* 如果 nCounter = 1500，则停止C++状态机。PLC nCounter 也设置为“0”，以便一切从头开始。



（7）编译之后在PLC实例的Symbol Initialization中关联接口：



（8）激活硬件配置下载程序：可以观察到PLC中随着nCounter不断增加，到500之后，调用接口中SetState方法，nCurrentState被置为1，并且StateMachine被启动。nCounter到1000之后，再次调用SetState方法，nCurrentState被置为2.最后，当nCounter超过2000时，StateMachine被停止，nCounter再次归0，从而开始一轮新的循环。在C++的TwinCAT Live Watch中，也可以观察到当前的状态和输出计时器的值：

图形用户界面, 文本, 应用程序

描述已自动生成

**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 |
|  |