BECKHOFF

TwinCAT 3 C++/Simulink教程

Version 3.11

毕孚自动化设备贸易（上海）有限公司

2017 年 5 月

前言

TwinCAT3 是基于 PC 的控制软件并且它开启了一个新的时代，是倍福公司历史上又一个里程碑。

特别是在高效的工程领域中 TwinCAT3 将模块化思想以及其灵活的软件架构，融入到整个平台。

几乎每一种控制应用程序都能在 TwinCAT3 中实现。从印刷设备、木工设备、塑料机械或门窗设备、风力发电机和实验台，亦或是楼宇，诸如剧院，以及运动场，一切都可以通过 TwinCAT3 实现自动化。

用户可以选择不同的编程语言来实现这些应用。除了经典的 PLC 编程语言的 IEC 61131-3，用户现在也可以用高级语言 C 或 C++，以及 MATLAB®/ Simulink®。

整合了运动功能从而简化了工程项目，以及全新的安全应用编辑更加人性化。

这些以及更多的特性都证明了为什么 TwinCAT3 也名为扩展的自动化。

本书针对任何想要学习倍福 TwinCAT3 软件如何实现基于 PC 控制编程的读者，阅读本书需要预先具备 IEC61131-3，C/C++或 MATLAB®/ Simulink®中至少一种编程语言的知识。

本书内容的架构安排如下：

第一章围绕 TwinCAT3 中支持 C++编程进行展开，从软件安装的注意点，简单的 C++项目创建、编写、调试，以及利用 C++语言封装成模块进行调用与被调用的方式等一一展开进行详细介绍。

第二章介绍 TwinCAT3 与 MATLAB®/ Simulink®如何实现交互，首先介绍了安装注意事项，其次就围绕 TE1400 和 TE1410 这 2 个交互工具开展学习，介绍了如何利用 MATLAB®/ Simulink®这一款强大的数学建模工具，直接服务于 TwinCAT3 平台，使得 TwinCAT3 应用领域又得到了很大的提升。

本书所有的内容都会不间断更新，如果想获取更新的教材可以通过访问 FTP 获取到，当然本书所有配套的案例程序也会在此 FTP 中供所有读者免费获取。

FTP 地址：ftp://ftp.beckhoff.com.cn/TwinCAT3/TC3_training/

欢迎对本书的结构、内容提出意见和建议，请发邮件至：

y.yang@beckhoff.com.cn

杨煜敏

2015 年 12 月 1 日

目录

一、 TwinCAT3 的 C++使用篇

1. 软件安装

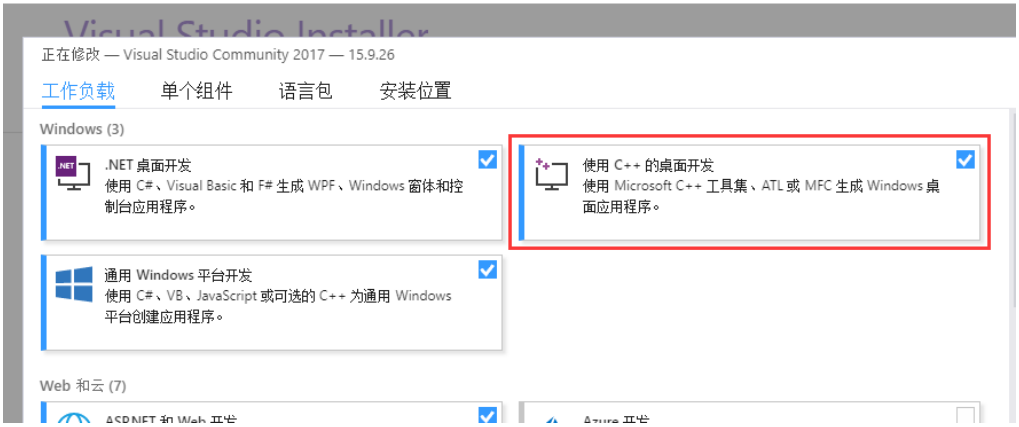
C++ 编 程 可 以 支 持 的 VS 软 件 版 本 ： Microsoft Visual Studio 2010/2012/2013/2015/2017，Level Professional，Premium，Ultimate，Community 及以上英文版本。本例程是以 VS2015 Community 做演示。

1.1 软件的安装步骤。

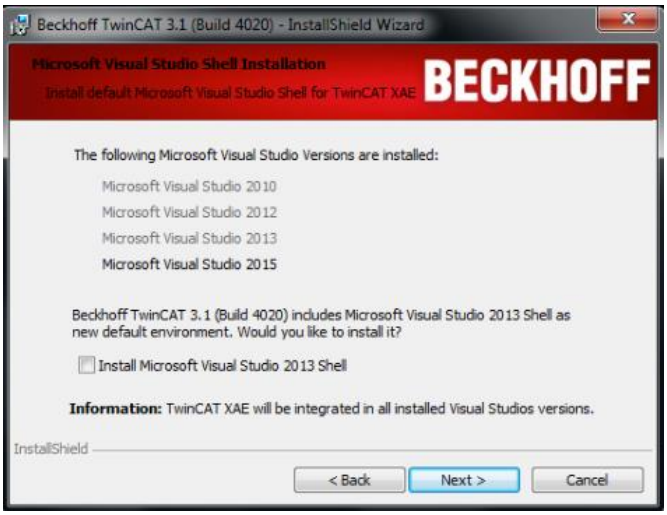
（1） 提前安装好 VS2015 Community，具体安装步骤省略。

安装过程中务必勾选 C++编译器，否则将无法使用 twincat3C++功能

如果您使用的是 VS2017：



（2）随后开始安装 Twincat3，和之前安装 full 版本的唯一区别是在安装过程中多了一个选项，TC3 软件会自动识别你当前电脑所安装的 VS 版本，之后勾选 Twincat3 的 runtime 嵌入到哪一个 VS 中，其他步骤可以参考之前的安装文档。如果在安装 VS 完整版之前已经装好的 TwinCAT3，则请先卸载 TwinCAT3，待 VS 完整版安装完成之后再进行 TwinCAT3 的安装。

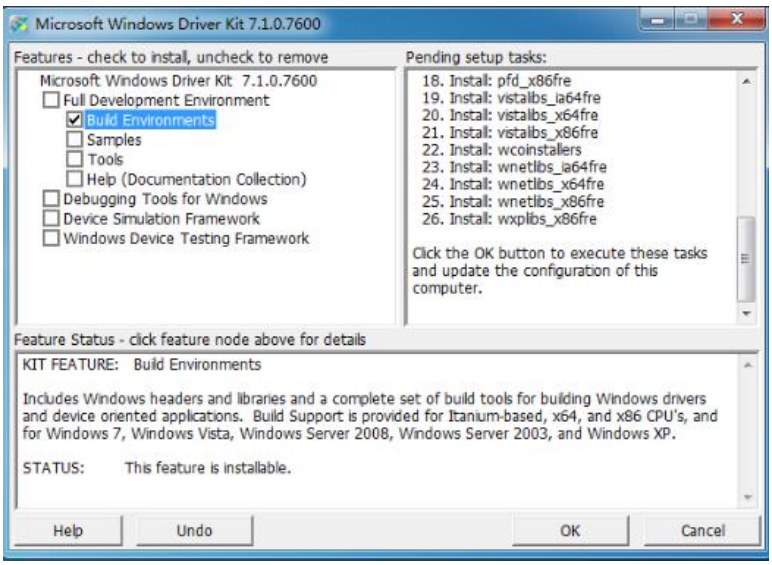


识别到电脑中有 VS 2015，因此不需要勾选安装 2013 Shell

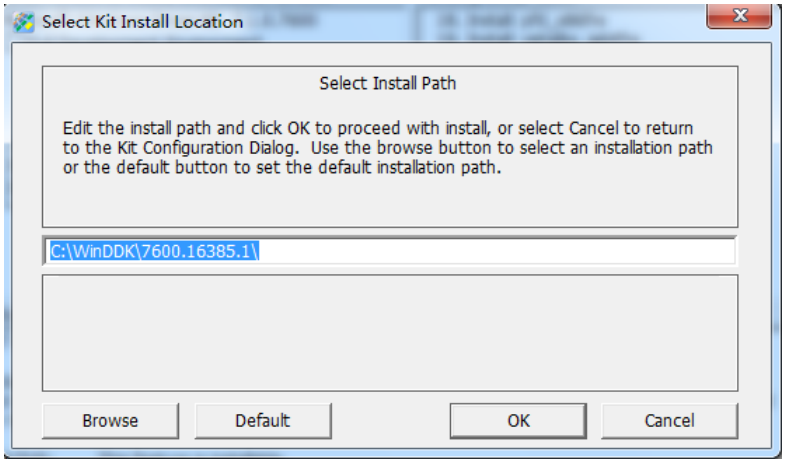
下一步勾选所需要嵌入的 VS 版本，因此勾选事先安装好的 VS2015

（5）接下来需要安装一个很重要的插件："Windows Driver Kit" (WDK) 下载链接：https://www.microsoft.com/en-us/download/details.aspx?id=11800

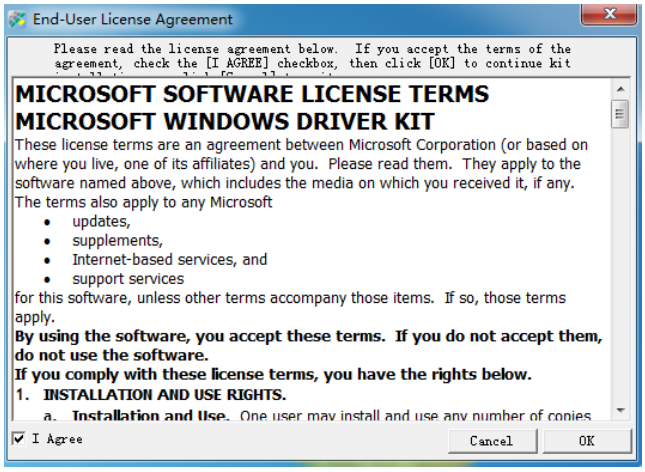
安装这个插件是为了能在 TwinACT 3 工程环境创建和编辑 C++模块。下载后用镜像打开选择“Build Environment”后点击 OK



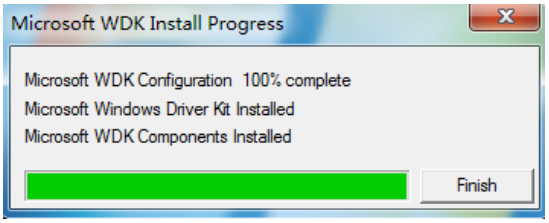
（6）弹出窗口点击 OK



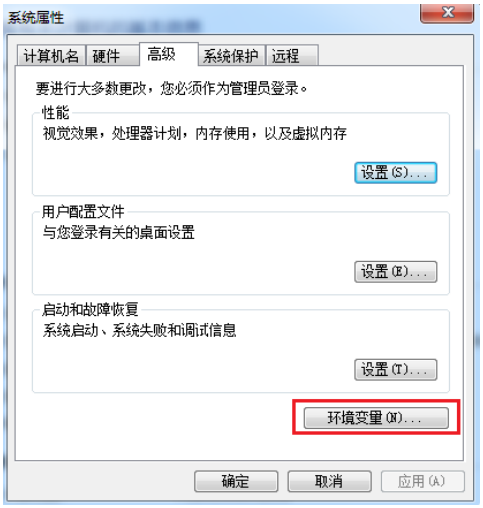
（7）选择“I agree”后点击 OK



（8）安装完成后点击 Finish 结束



（9）随后打开【系统属性】修改环境变量，点击【环境变量】

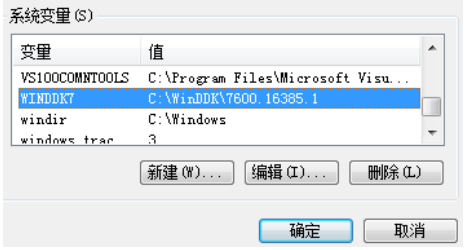


（10）点击系统变量下的“新建”，分别填入变量名和变量值，

变量名：WINDDK7

变量值：C:\WinDDK\7600.16385.1

（11）设置好后会出现在系统变量中

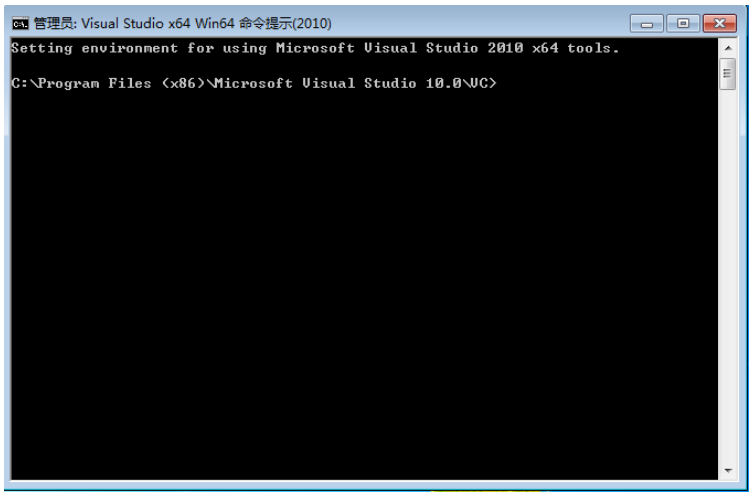


2. TwinCAT3 在 64 位操作系统中数字证书的安装

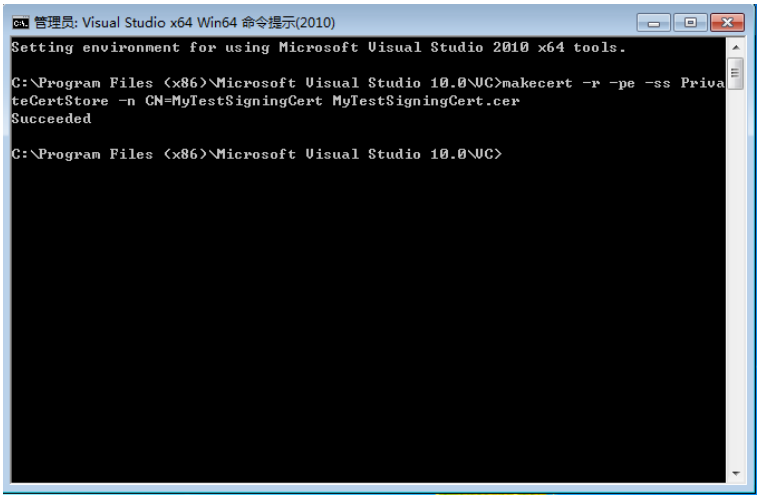
TwinCAT3 安装在 64 位操作系统上的时候，开发并运行 PLC 没什么问题，但如果要开发并运行 matlab-simulink 或者 C++则需要安装测试证书，以下步骤就是教大家如何正确安装数字证书。

2.1 在开发电脑中创建并安装数字证书

（1）以管理员权限打开 Visual Studio 2010/2012/2013/2015 命令提示窗口（所有应用程序→Microsoft Visual Studio 2012/2013→Visual Studio Tool→Visual Studio 2012/2013 命令提示窗口）。

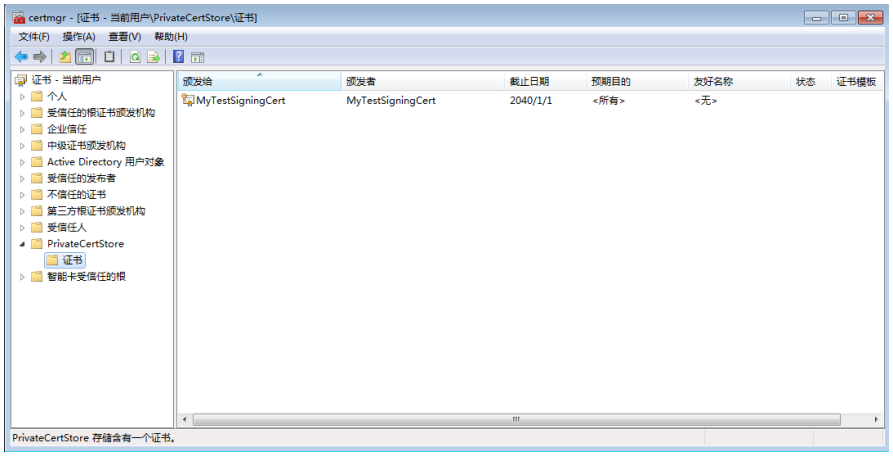


（2）创建证书并保存，输入以下指令：makecert -r -pe -ss PrivateCertStore -n CN=MyTestSigningCert MyTestSigningCert.cer。敲回车后提示 Succeeded

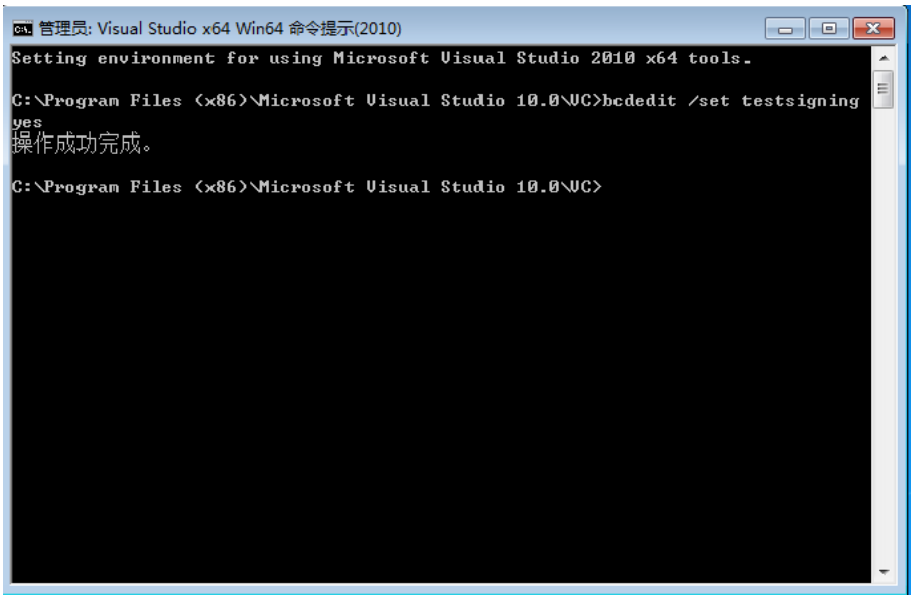


（3）命令栏输入 certmgr.msc 打开证书管理查看到刚才的证书创建成功

注意：千万不要多次添加，如果看到有重复的必须手动删除。



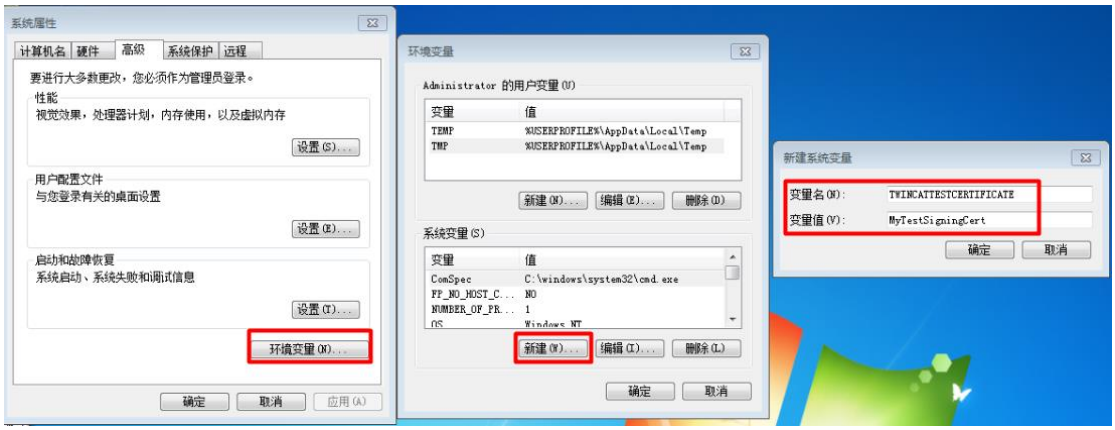
（4）再次以管理员权限打开 Visual Studio 2012/2013 命令提示窗口，输入命令激活测试证书：bcdedit /set testsigning yes。



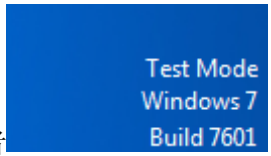
（5）接下来打开系统环境变量添加新的系统变量

变量名：TWINCATTESTCERTIFICATE

变量值：MyTestSigningCert

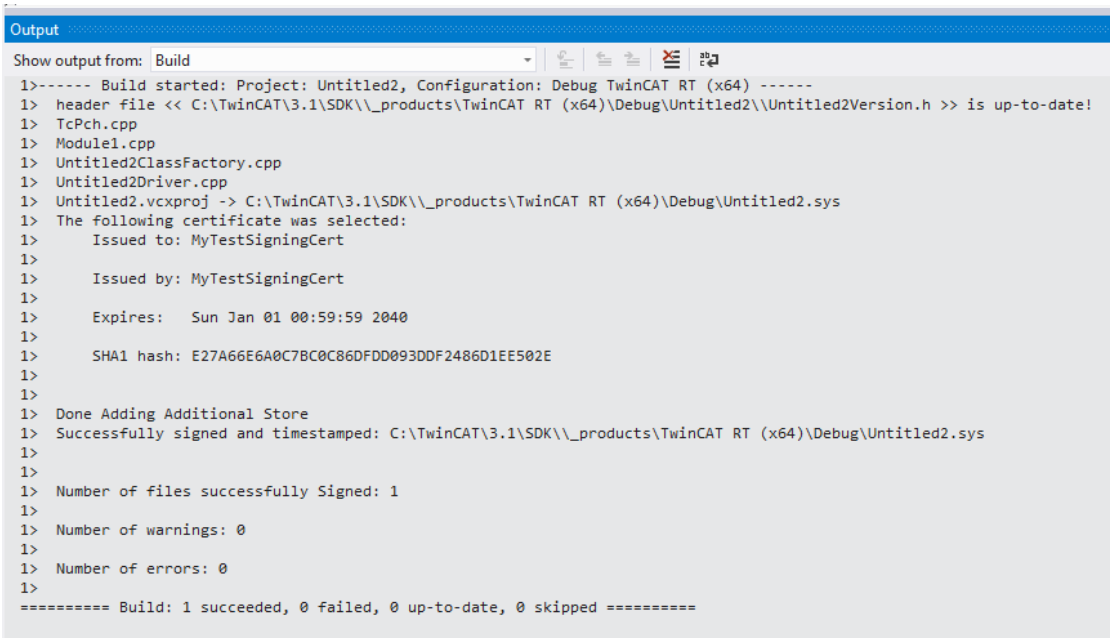


注意：做好以上步骤后必须重启电脑，你会发现电脑右下角显示以下内容。

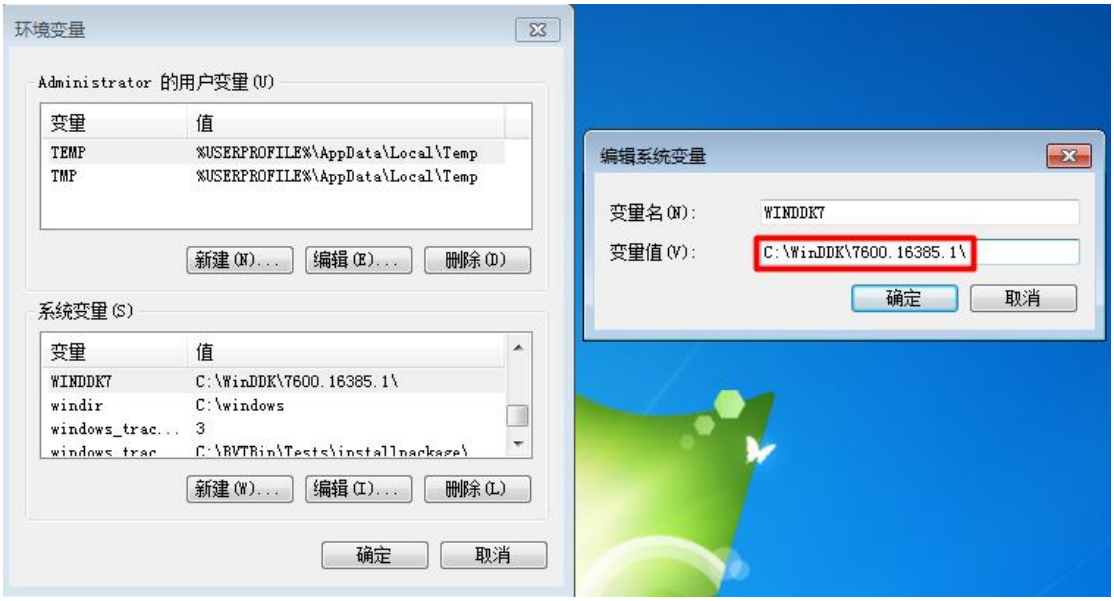
或者


如果是 Win10 则会使这样

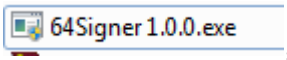

（6）随后打开 TwinCAT3，新建 C++项目，你就会发现调用测试证书编译成功：



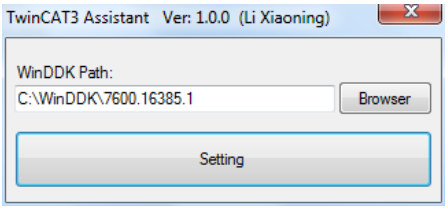
（7）最后如果你的电脑进行以上操作后运行程序还是会有报错，则可以把环境变量中 WDK 的变量值最后加”\”



为了方便大家设置环境变量以及 64 位系统上的证书创建，本教材提供了一键设置工具



，只需要双击打开，



点击 setting 就会自动完成本章节之前所描述的步骤，如下：

1.软件安装→（9）~（11）

2.1. 在开发电脑中创建并安装数字证书

} 一键完成

无论是 32 位还是 64 位都会自动完成各自所属步骤，但别多次点击，否则会创建多个数字签名证书，此时就需要打开证书列表进行删除，保证只有一个证书存在。

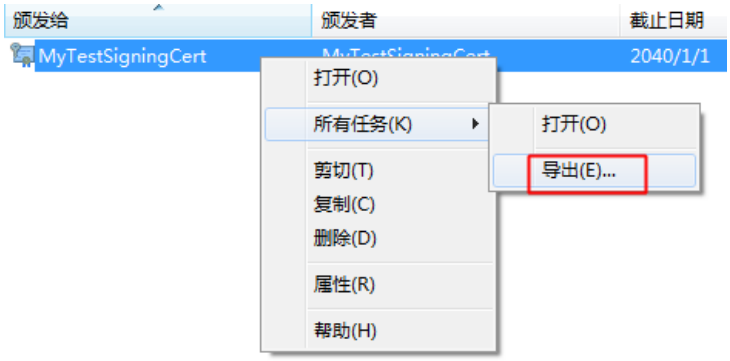
设置完成后重启电脑即可。

2.2 在 64 位目标控制器中安装数字证书

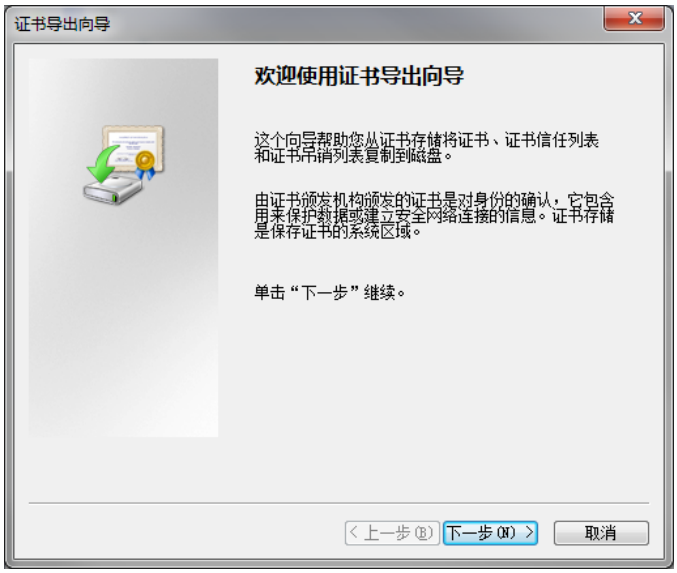
如果所开发的 C++或者 Matlab/Simulink 模块运行在 EPC 或者 IPC 中，那这台目标控制器也需要安装并配置 64 位数字证书。

（1）首先在本地创建此证书（步骤在上文中已经有详细介绍，参考 2.1（1）-（3））

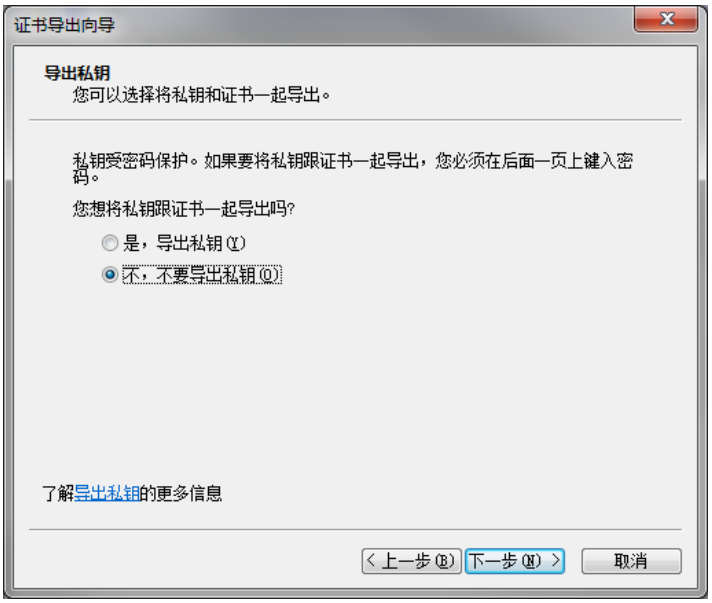
（2）随后手动导出此授权，在 certmgr 中找到此证书，右键导出



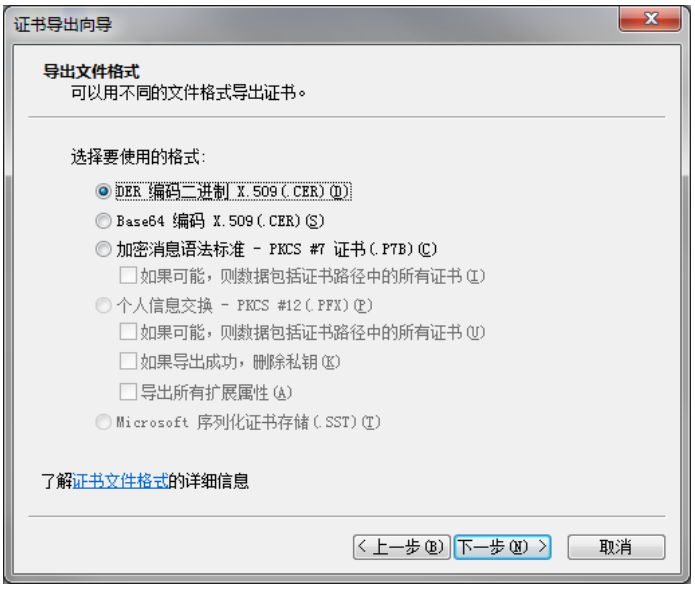
下一步



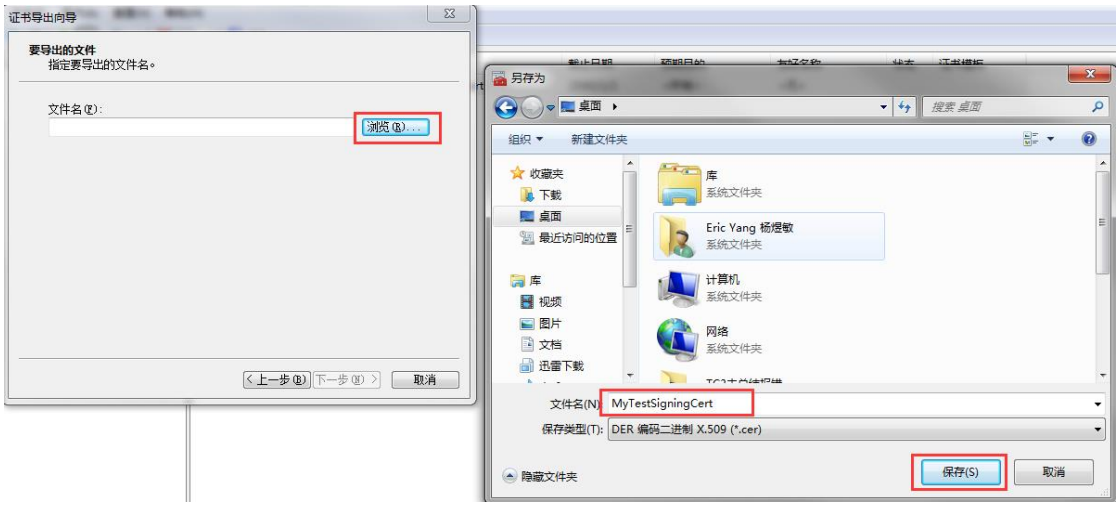
继续下一步



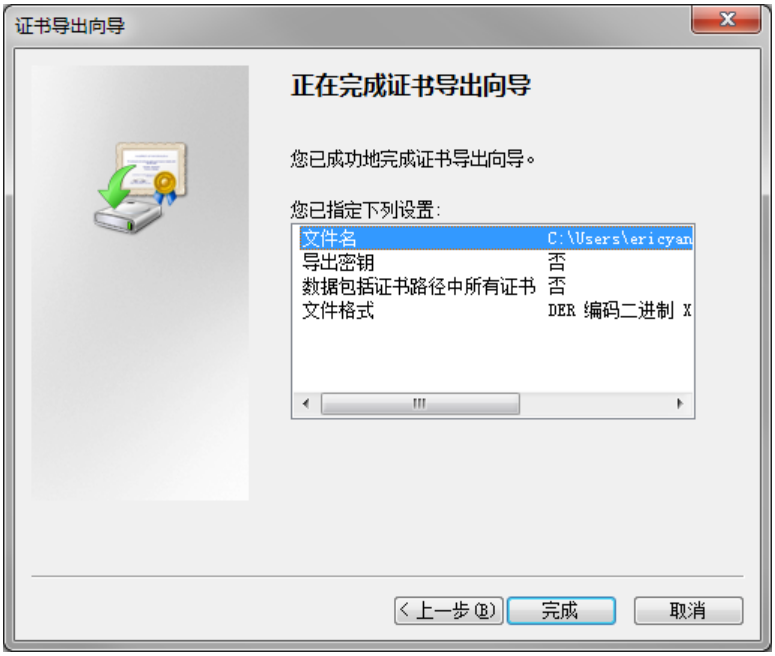
现在默认设置继续下一步



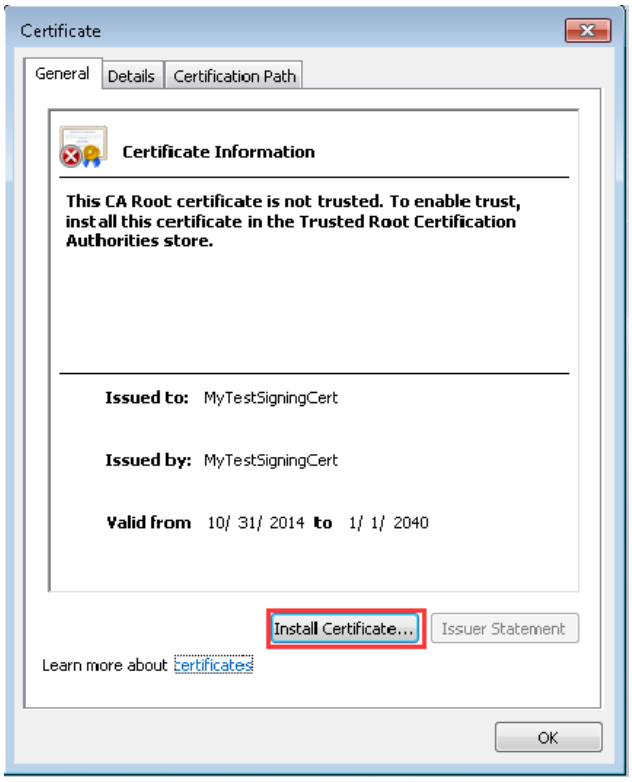
随后选择一个路径保存此证书



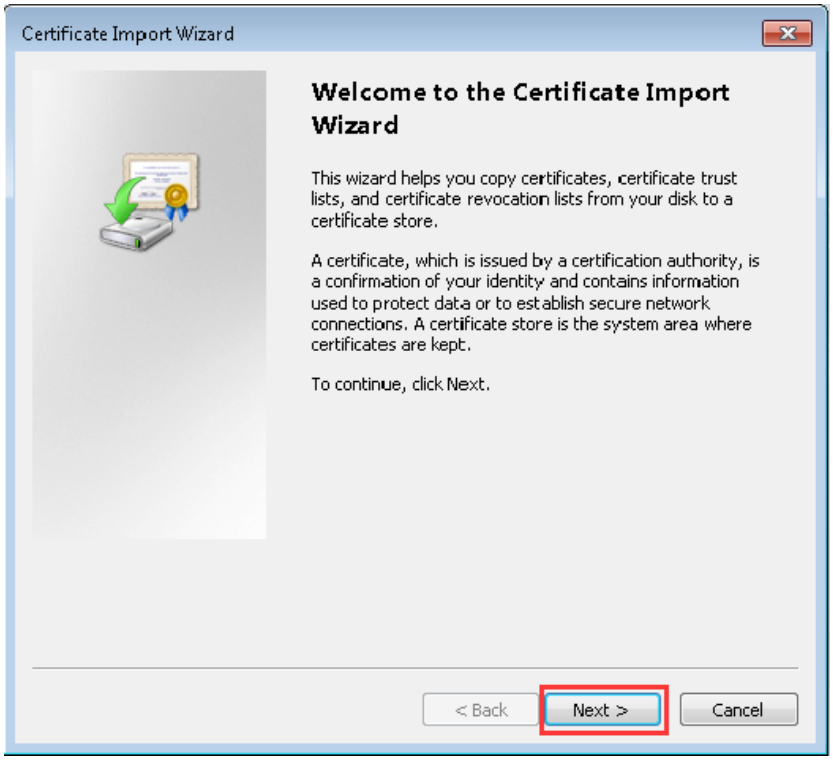
最后点击完成即可导出证书



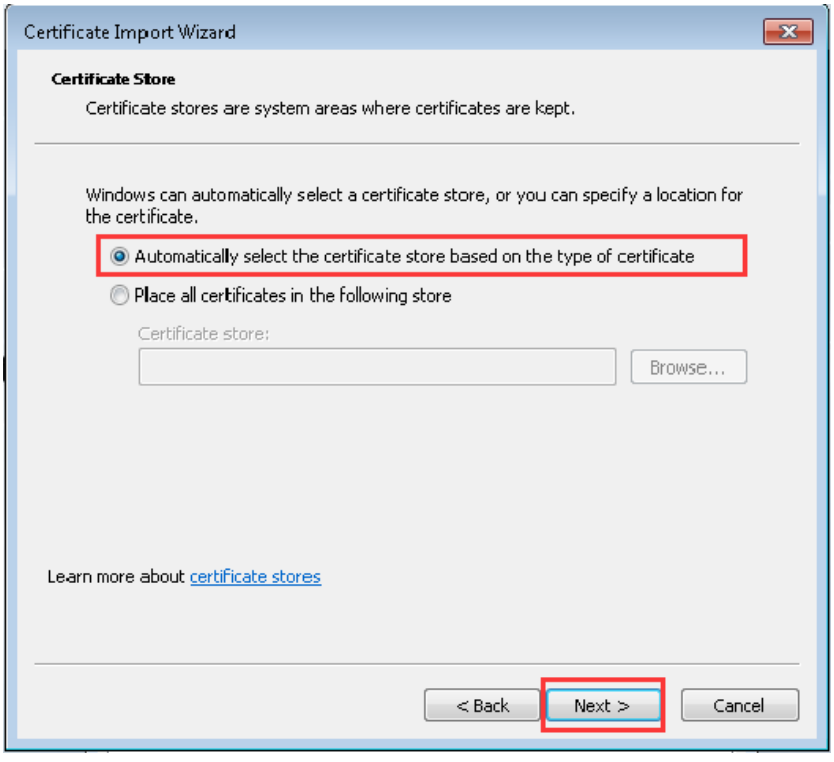
（3）拷贝证书到控制中双击进行安装，点击 Install Certificate



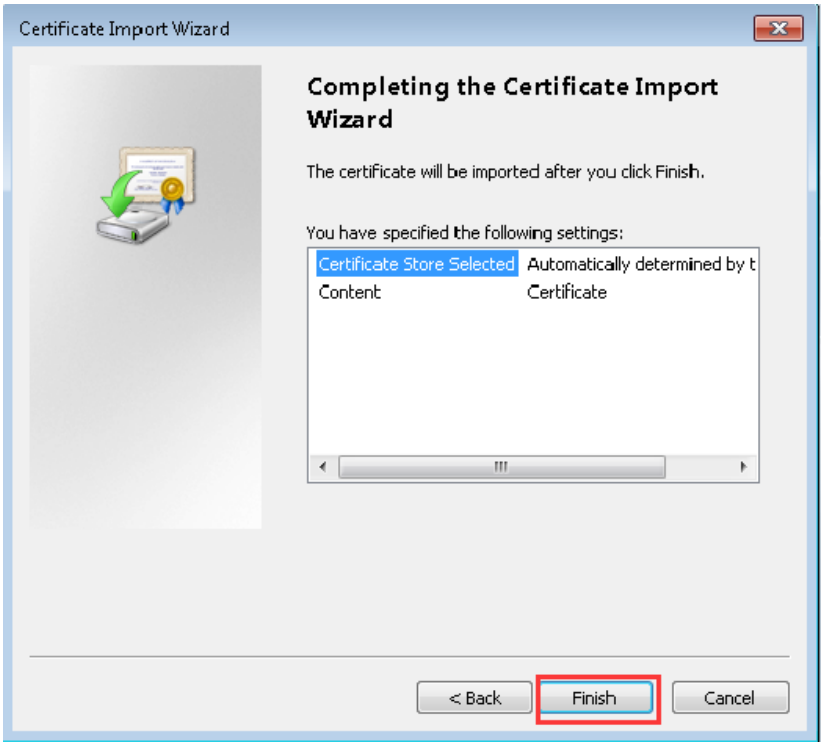
点击 Next



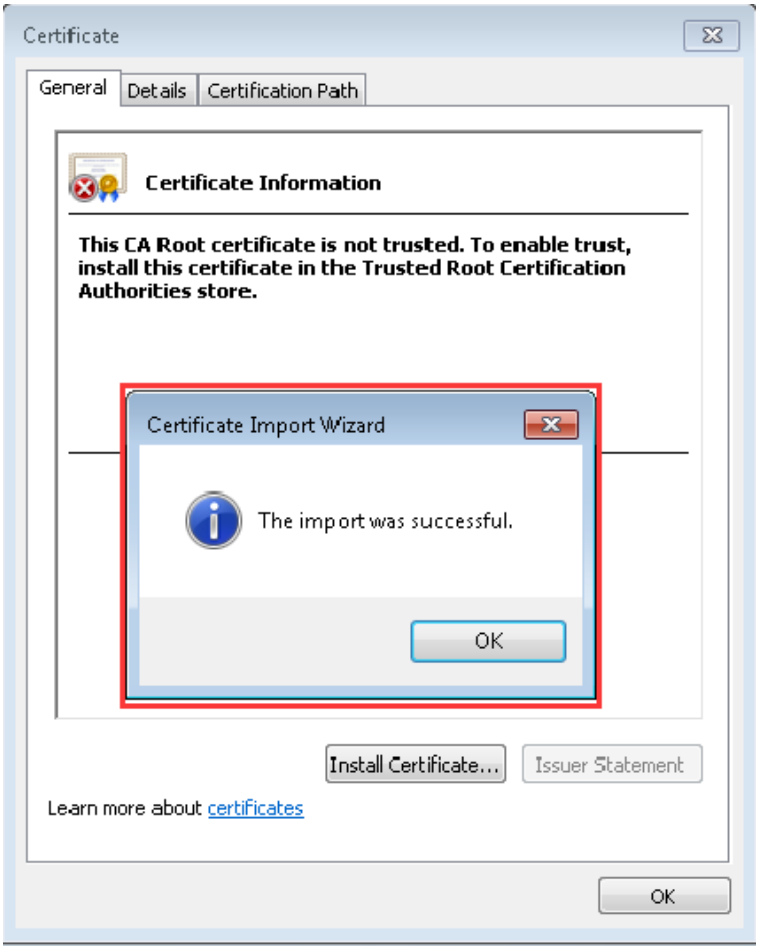
选择自动选择存储位置，随后 Next



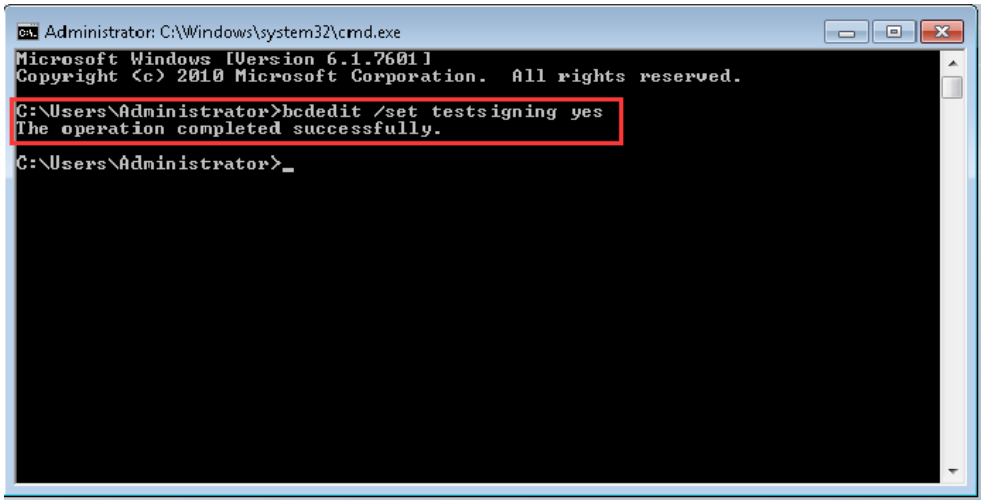
点击 Finish



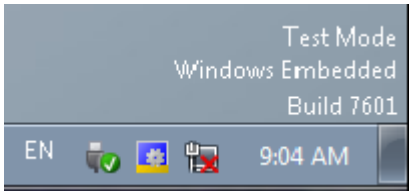
最后弹出窗口安装成功



（4）随后打开 CMD，输入命令激活测试证书：bcdedit /set testsigning yes



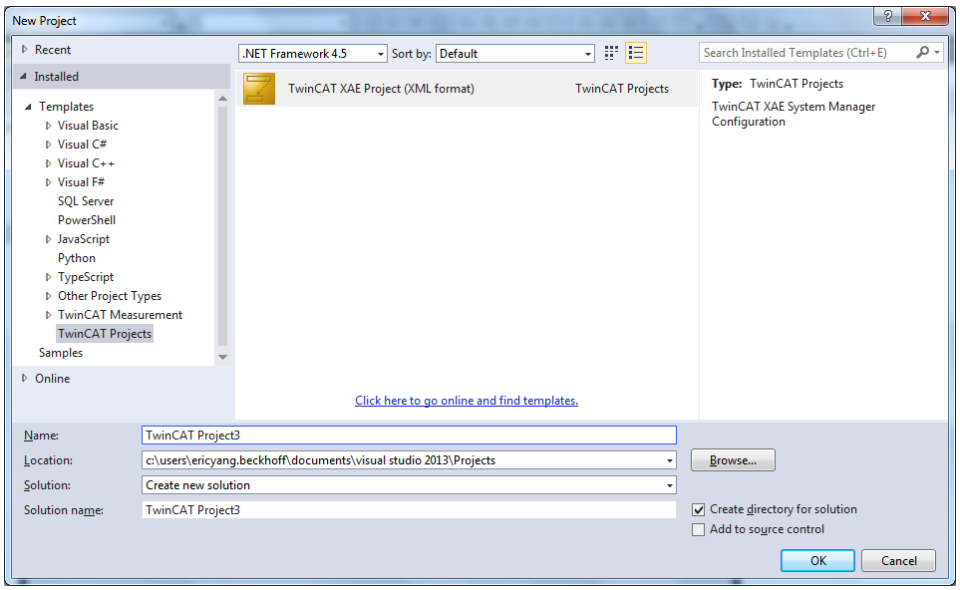
重启电脑后在右下角出现 Test Mode 就说明签名安装成功



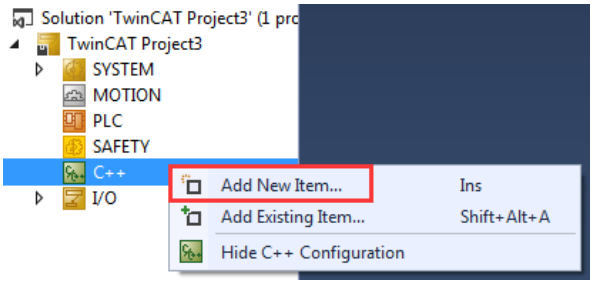
3. C++程序和硬件做链接

3.1 C++程序和硬件连接的步骤。

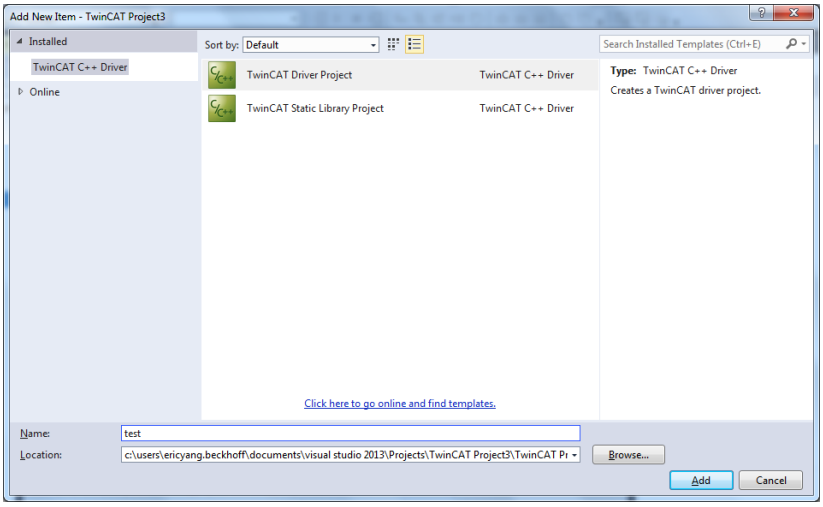
（1）新建项目



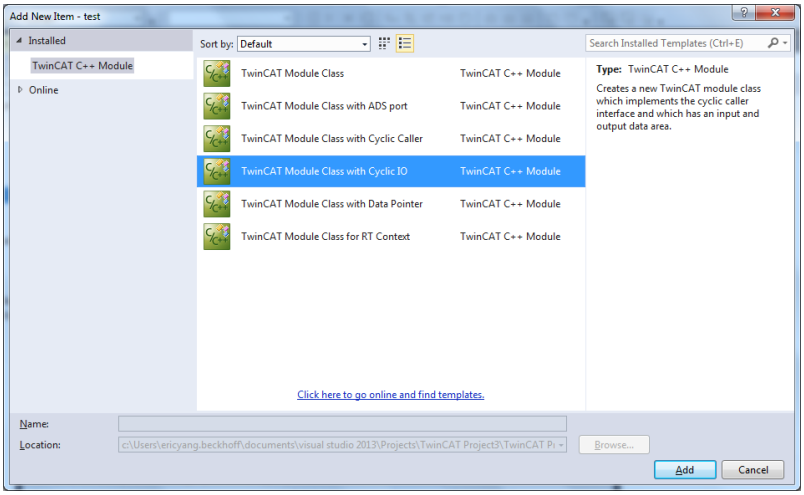
（2）新建 C++项目，名称是英文（不能是中文和符号）。



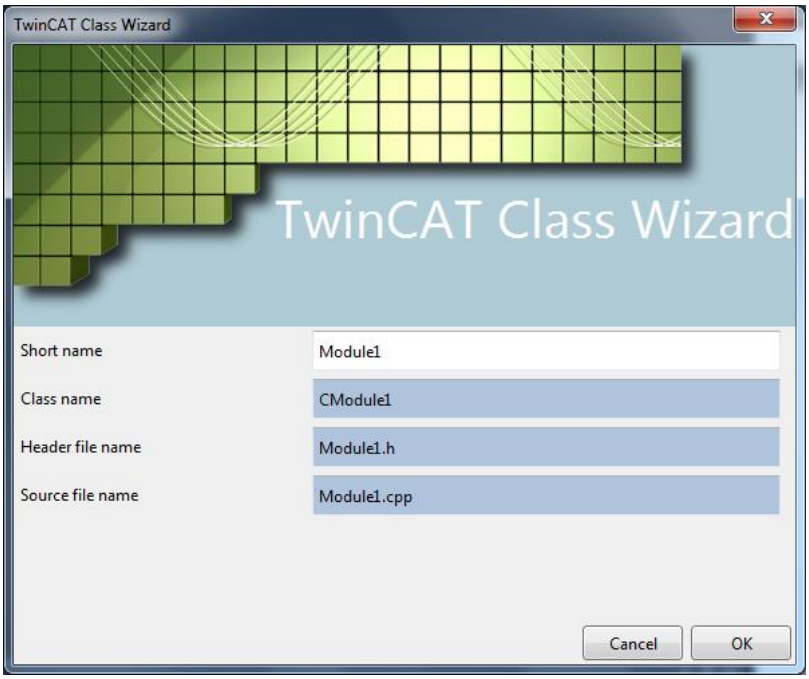
（3）选择 TwinCAT Driver Project 后点击添加。



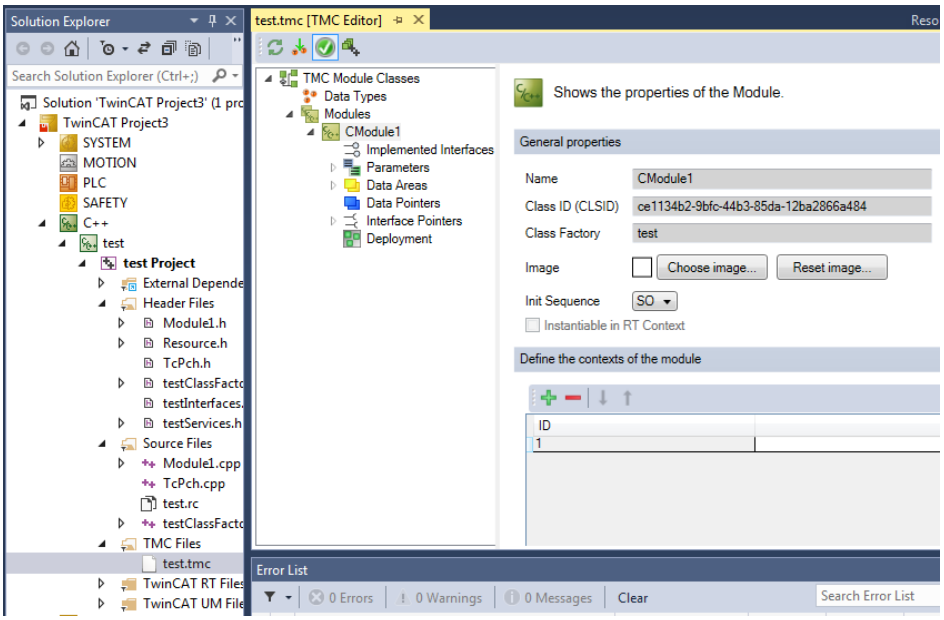
（4）选择 TwinCAT Module Class with Cyclic IO 点击添加。



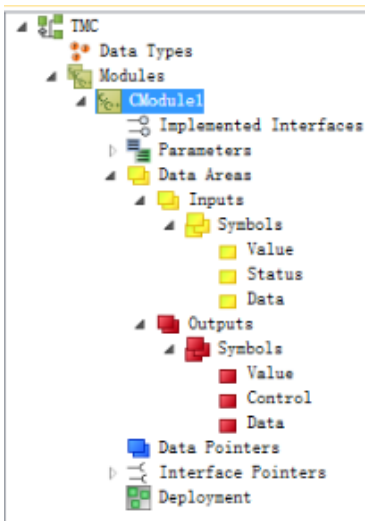
（5）定义一个 short name 或者用默认的 Module1 也可以，随后点击 OK。



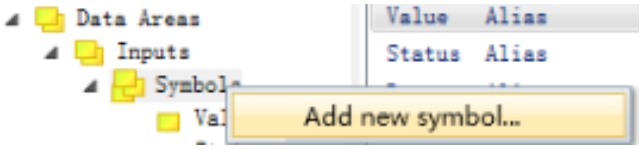
（6）双击 TMC Files 下的 test.tmc 开始创建变量，



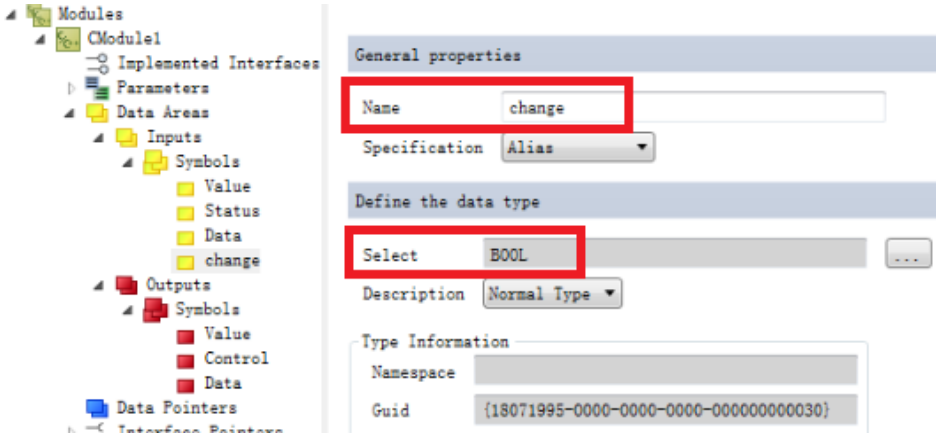
（7）点开 Data Areas 下的 Inputs 和 Outputs 可以发现系统已经创建了 3 个变量，可以用系统给的变量，当然也可以手动新建。



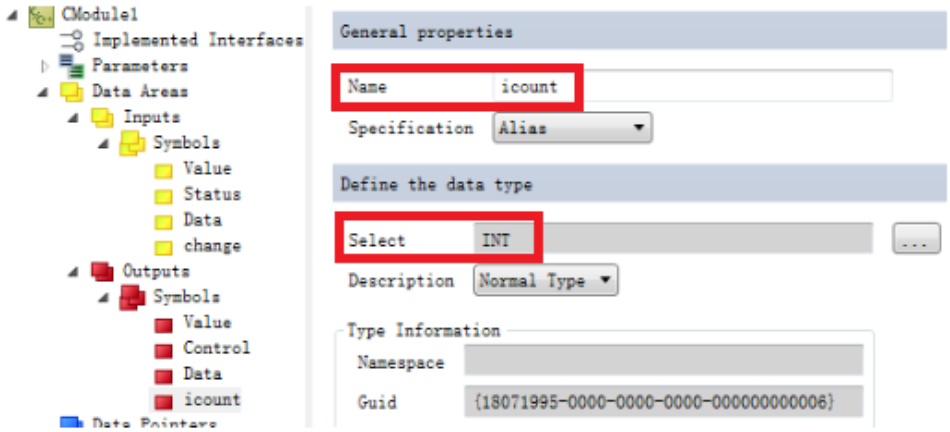
（8）先手动创建一个 Inputs 变量，右键 Inputs 下的 Symbols，选择 Add new symbol，



（9）把新建的变量名字和类型分别改成如图所示：Name：change；Select：bool。

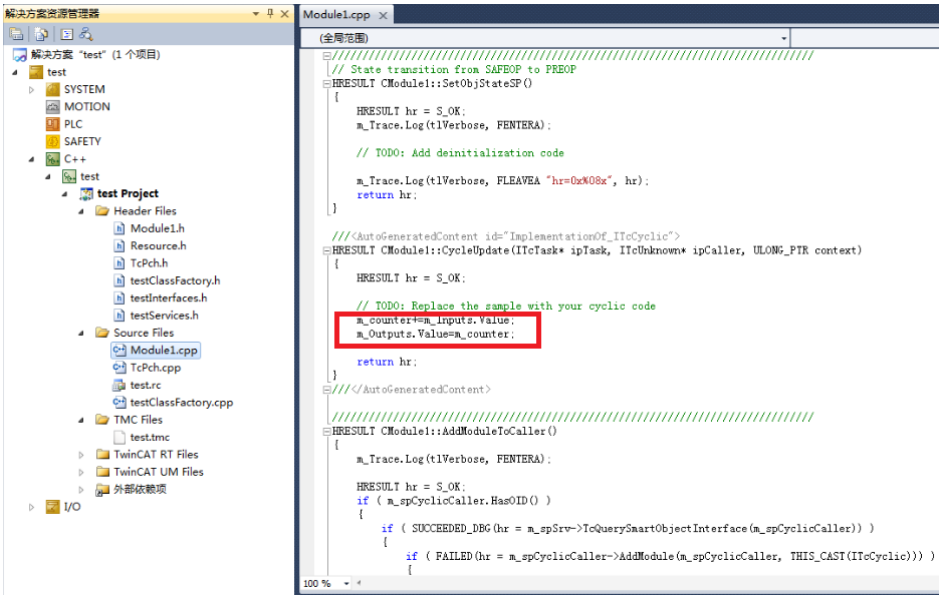


（10）同样也创建一个输出变量 icount，类型为 int，



（11）变量写好后右键 test Project，选择 TwinCAT TMC Code Generator 重新生成 TMC 代码。

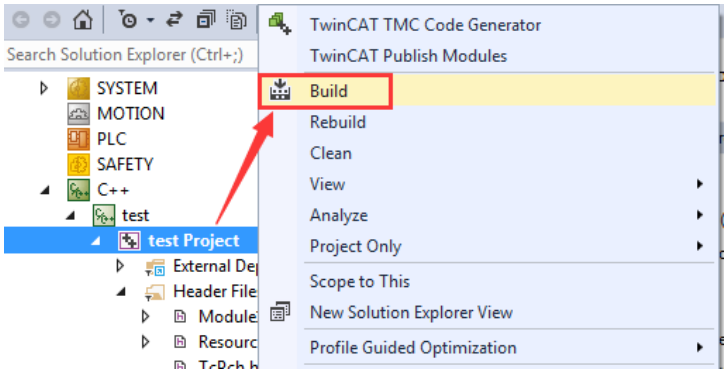
（12）双击 Source Files 文件夹下的 Module1.cpp 开始写 C++程序，程序编写在 //TODO: Replace the sample with your cyclic code 下面，也就是下图中红色部分，可以发现新创建的 C++项目中已经自带一条简单的程序，我们可以删除，替换成我们所需要的程序。



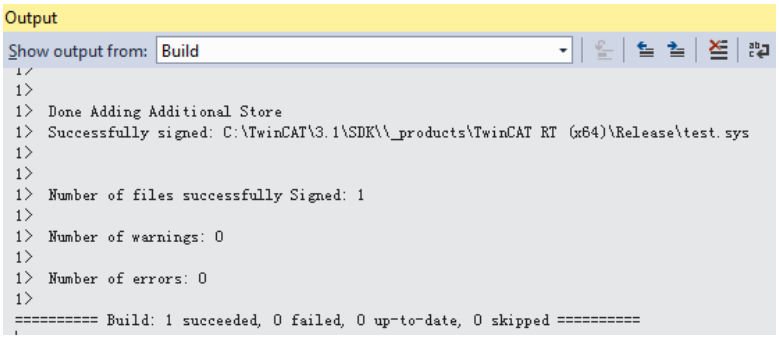
（13）把程序删除后用刚才创建的 2 个变量编辑一条简单的程序，change 导通触发 icount 计数器累加计数的功能。

```
    // TODO: Replace the sample with your cyclic code
if(m_Inputs.change)
{m_Outputs.icount++;
}
```

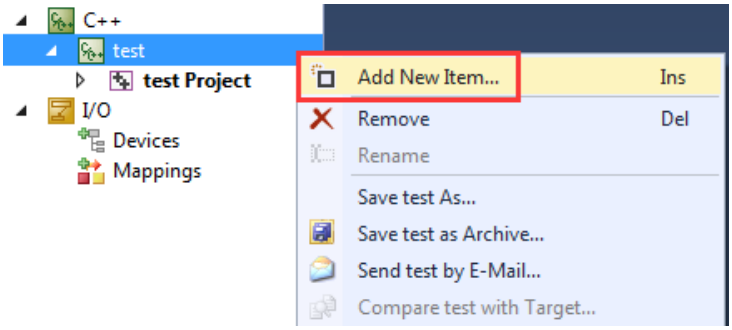
（14）程序写好后右键 test Project 选择生成开始编译 C++项目。



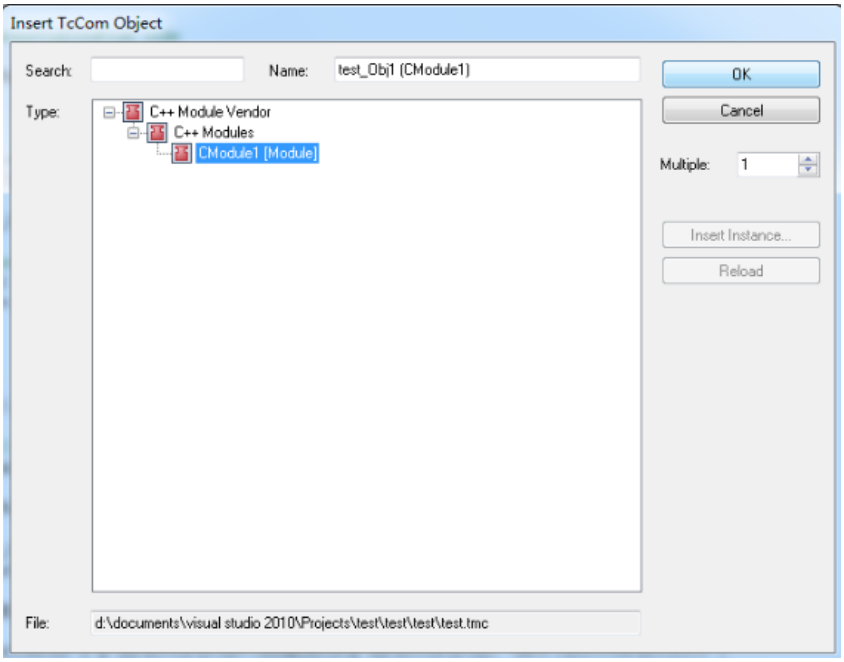
编译完成后在消息窗口可以看到如下提示：



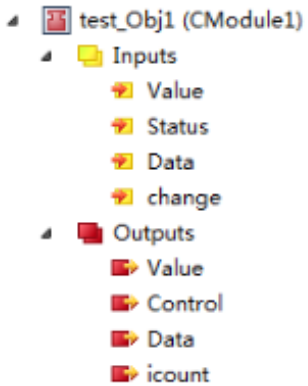
（15）右键 test 选择添加新项开始添加 C++接口模块，



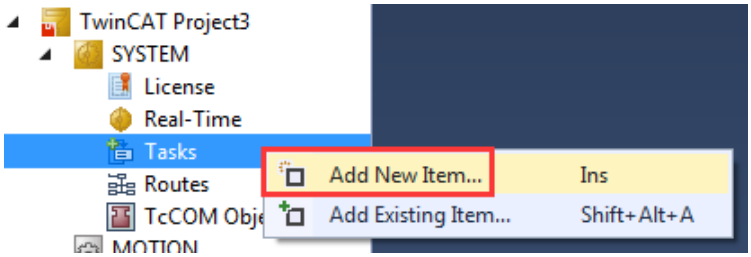
（16）弹出窗口选择 CModule1[module]后点击 OK，



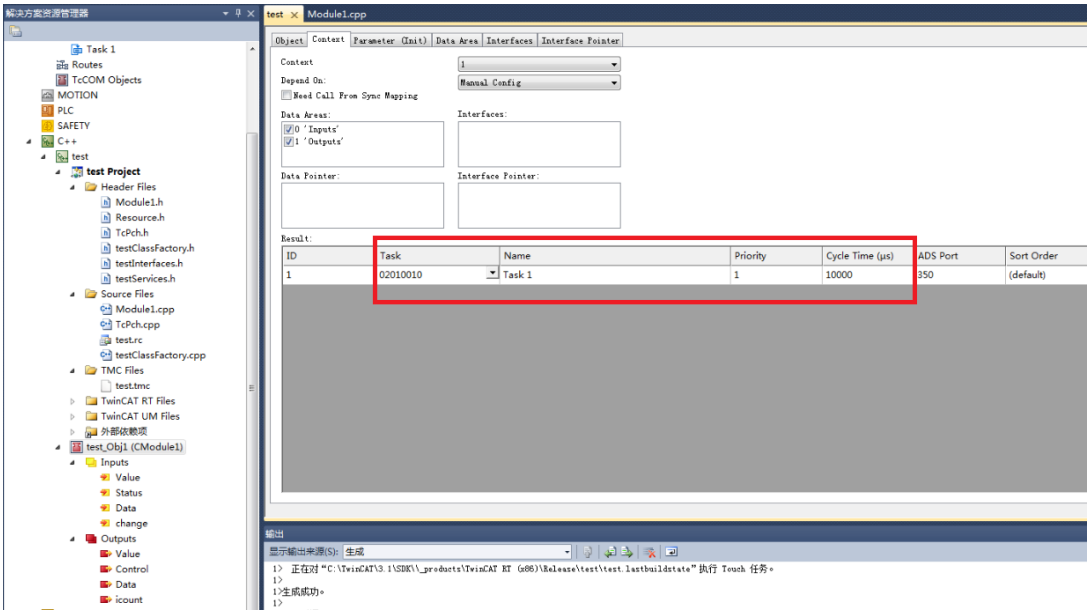
（17）添加完成后可以在 test 这个 C++项目中看到此项目的一些 IO 变量，



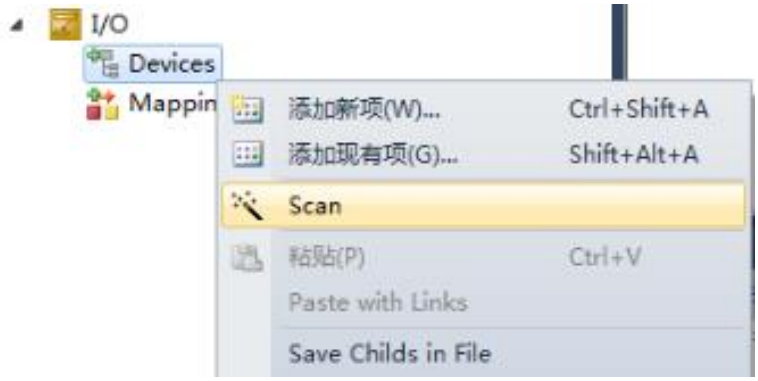
（18）右键 SYSTEM 下的 Tasks，选择添加新项开始创建 C++项目所需的 Task。



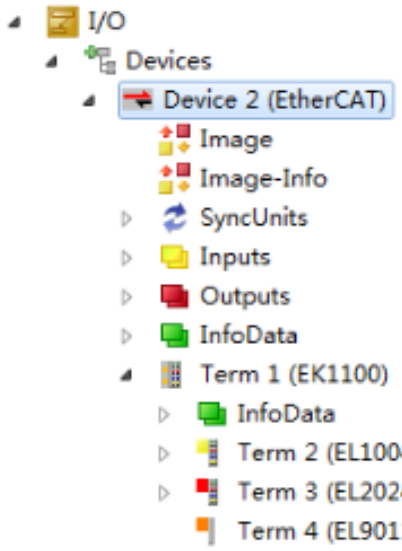
（19）双击 test_Obj1（CModule1），把新建的 Task 分配给 C++项目，



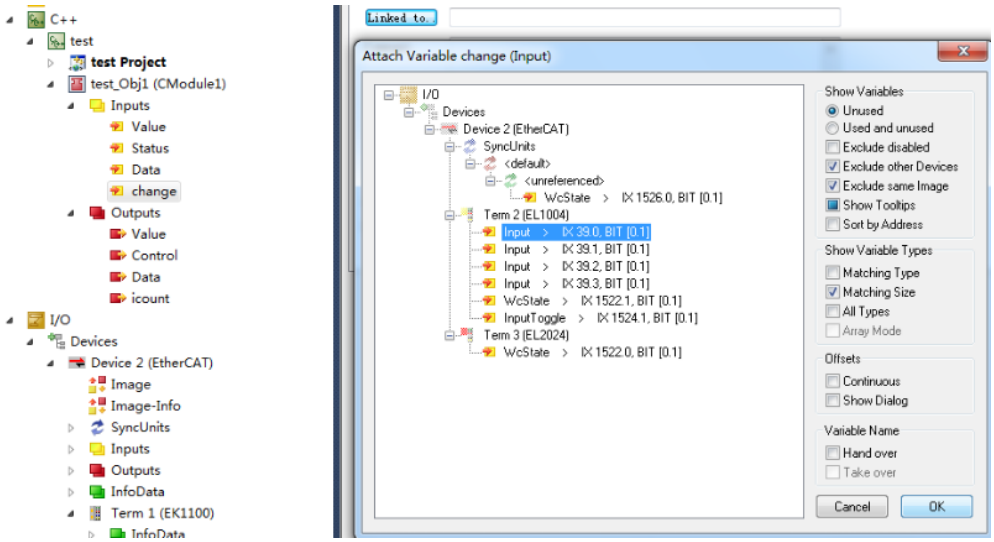
（20）右键 Devices，选择 Scan 开始扫描设备。



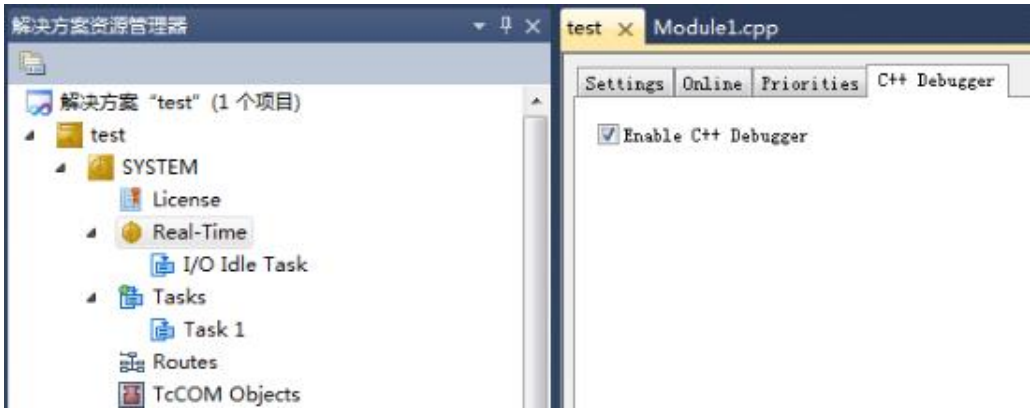
（21）本次试验用了简单的几个模块。



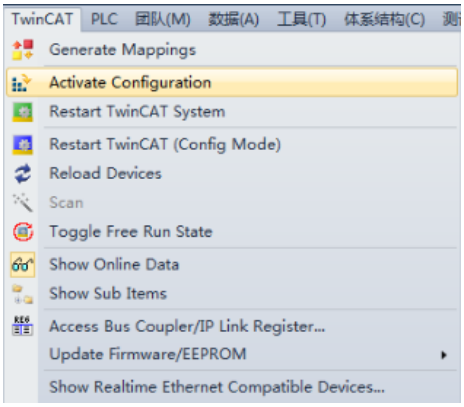
（22）把 C++项目中 change 变量连接到 EL1004 的通道 1 上，点击 OK。



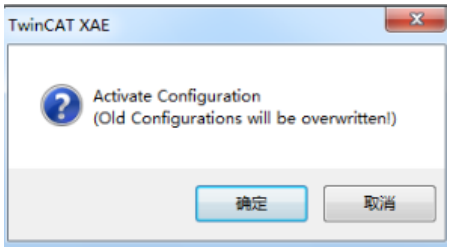
（23）最后双击 Real-Time，把 C++Debugger 中 Enable C++ Debugger 勾选，这样我们才可以在 C++中进行调试。



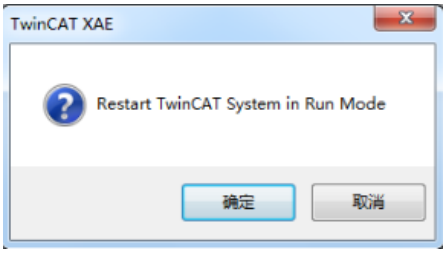
（24）打开 TwinCAT，选择 Activate Configuration 把配置下载到控制器中，



（25）弹出窗口点击确定。

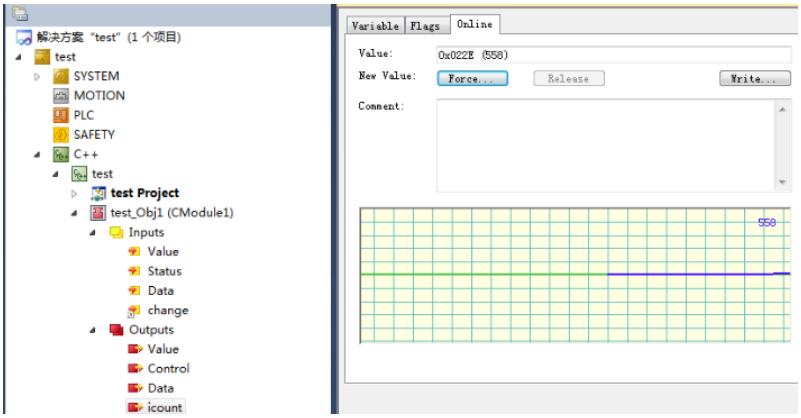


（26）提示切换到运行模式点击确定。



观察右下角图标是否变成绿色运行状态

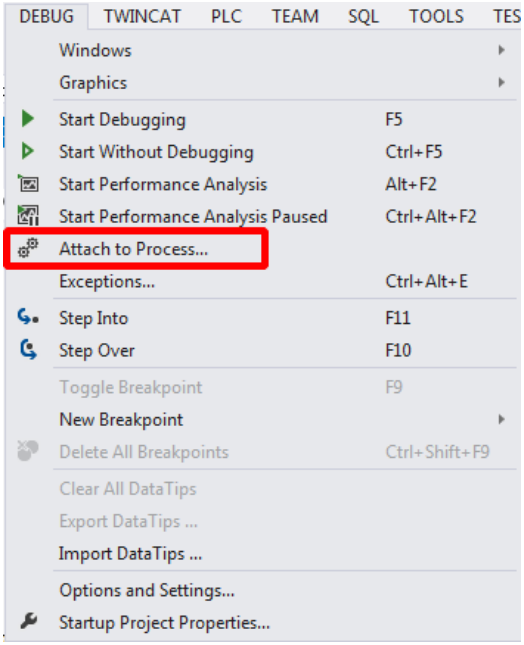
（27）点击 test_Obj1（CModule1）下的 icount，观察 Online，当外部按钮导通 EL1004 的通道 1 时，icount 数值变化。



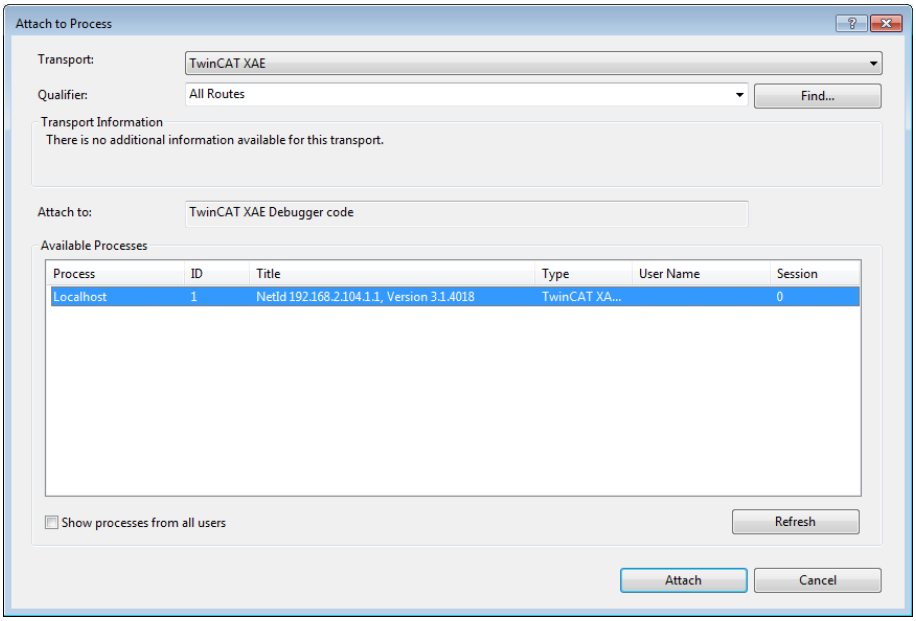
4. C++程序调试操作

4.1 C++程序调试步骤。

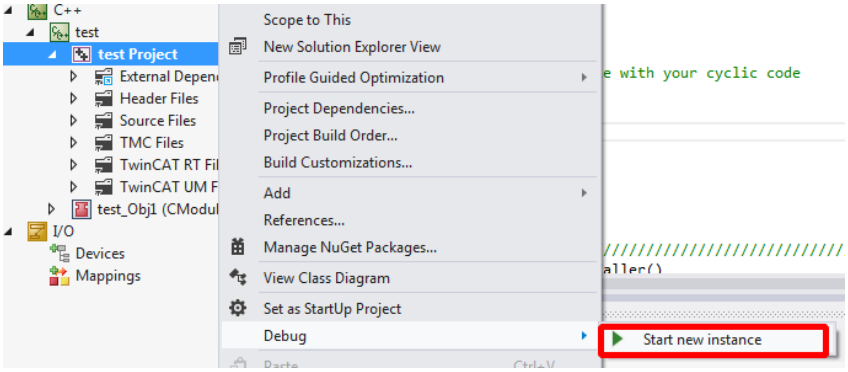
（1）点击调试菜单下的“attach to process”，



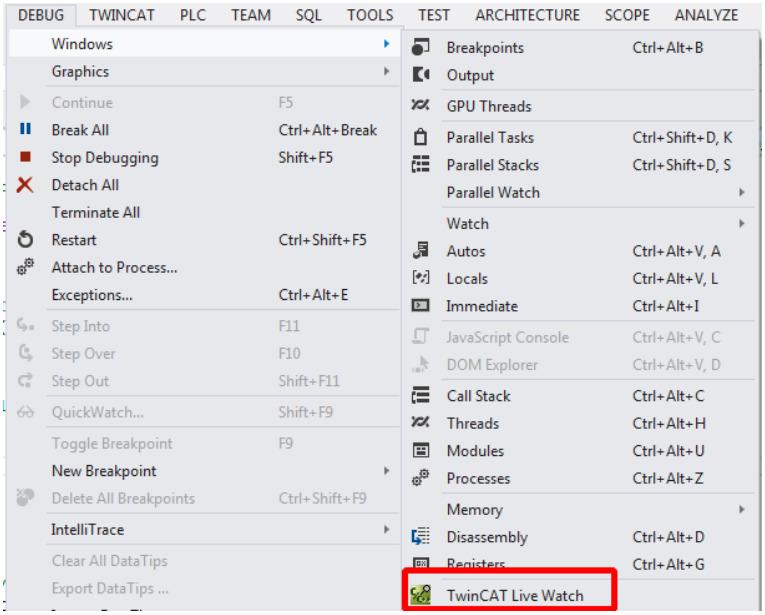
（2）弹出窗口，把传输改成 TwinCAT XAE，限定符改成 All Routes，随后双击可用进程中您想要调试的那一个进程。



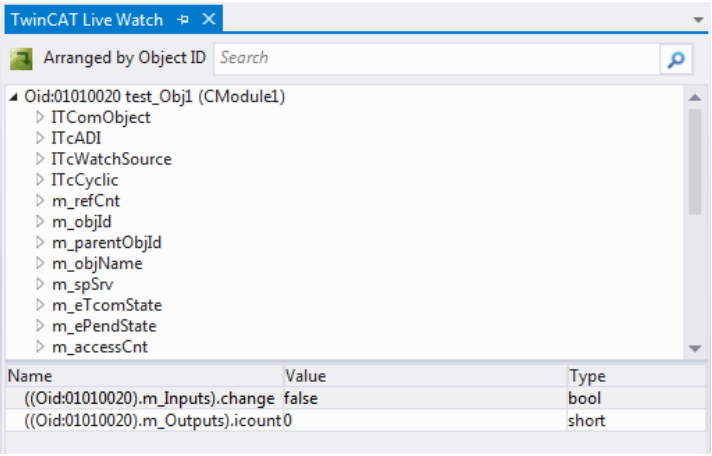
（3）随后右键 C++项目找到 debug→start new instance。



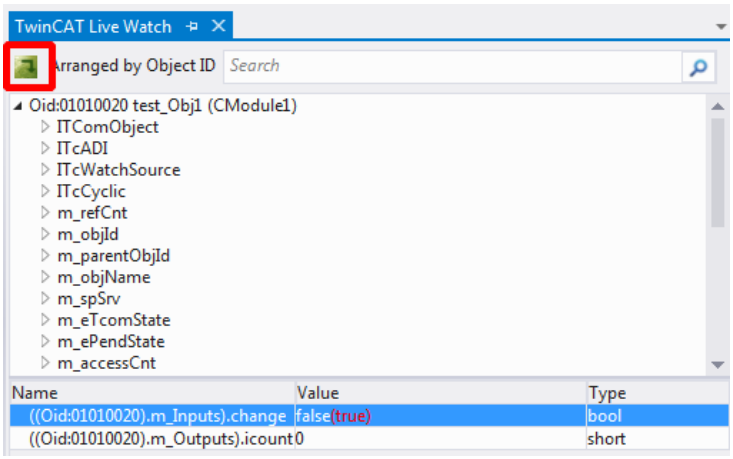
（4）随后会自动弹出 TwinCAT Live Watch，如果没有弹出，可以在 debug→windows 中找到并打开。



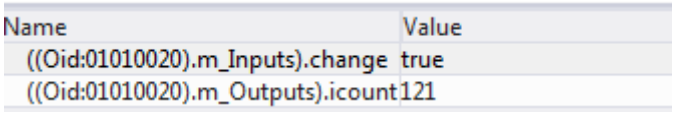
（5）随后就可以看到一个模快的实例可以在线监控，你可以任意选择需要监控的变量，比如我把之前创建的 2 个变量双击加入监控栏中。



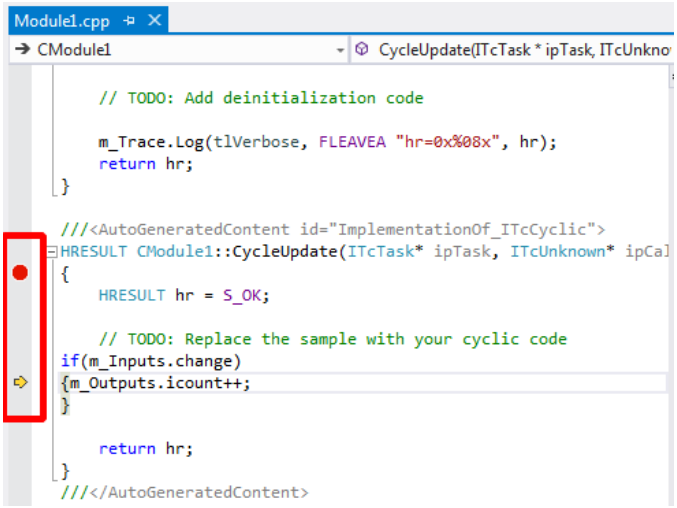
（6）双击需要修改的变量输入修改值，随后点击左上角的绿色 download 按钮就可以在线修改变量。



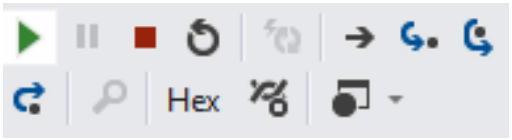
（7）可以发现 icount 开始循环累加。



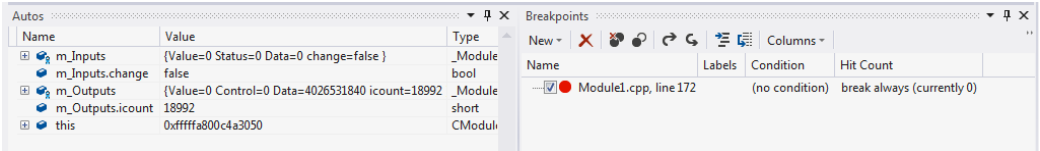
（8）当然你也可以在程序中直接添加断点进行调试，添加断点方式很简单只需要在右边灰色框中直接点击就可以添加新断点。



通过工具栏也可以进行 step into,step over, step out 等功能。



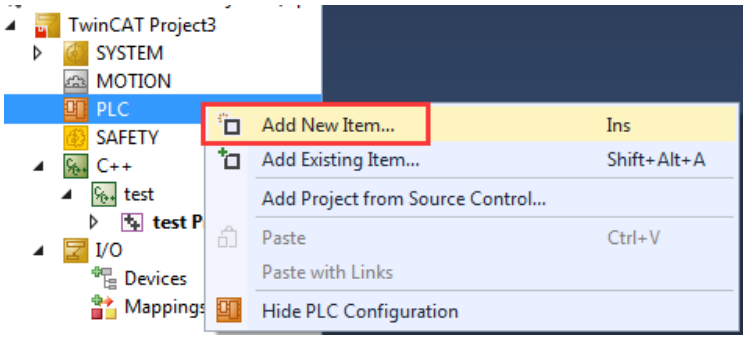
同时在消息窗口可以观察到断点位置，和当前调试的整个过程



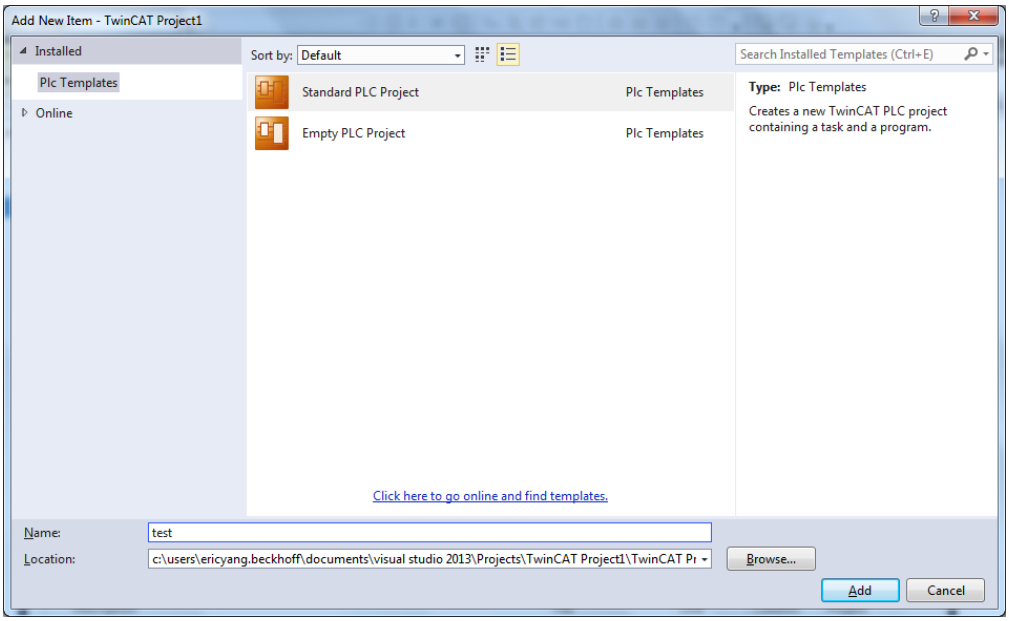
5. C++程序和 PLC 程序变量映射

5.1 C++程序与 PLC 程序链接步骤。

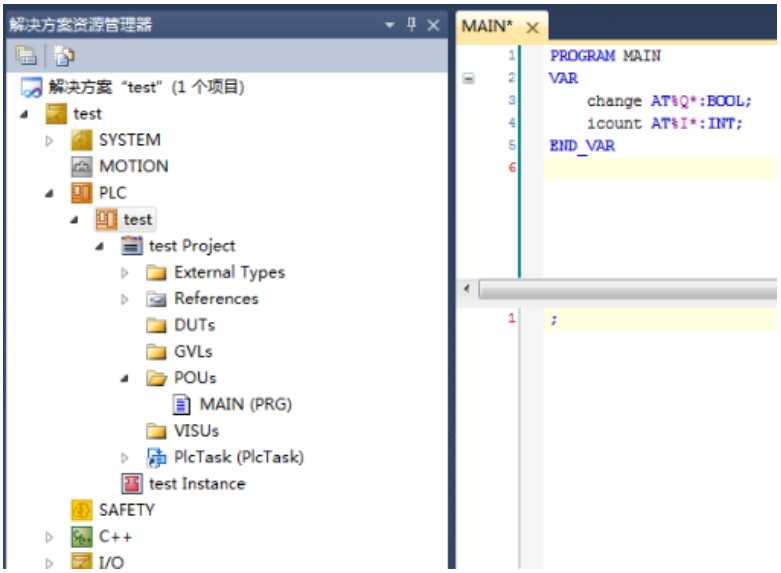
（1）用之前的 C++的案例程序，首先我们需要添加 PLC 程序，右键 PLC 选择“添加新项”



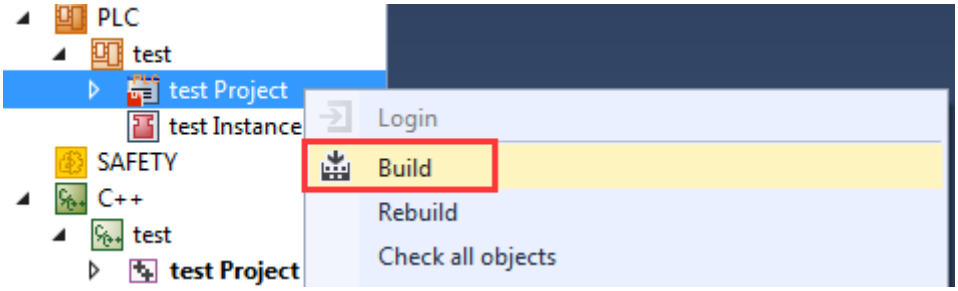
（2）弹出窗口后把名称改成英文（不能是中文和字符）。



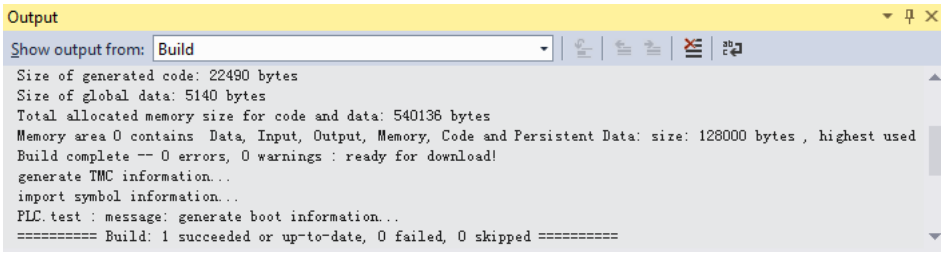
（3）打开 POUs 下的 MAIN 编辑 PLC 程序，创建两个变量即可，



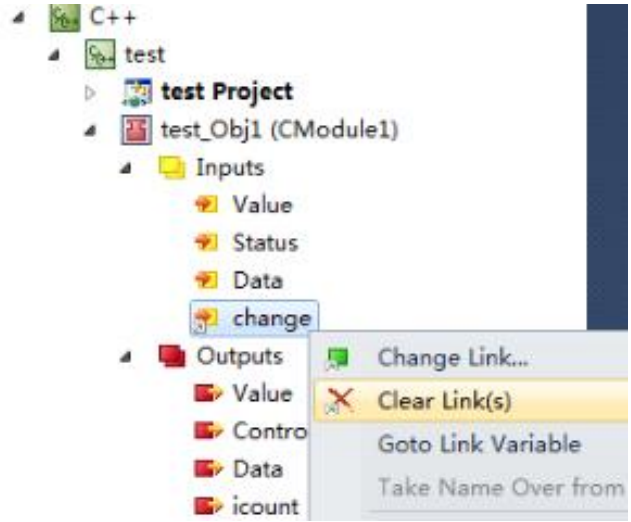
（4）程序写好后右键 test Project 选择“生成”开始编译程序。



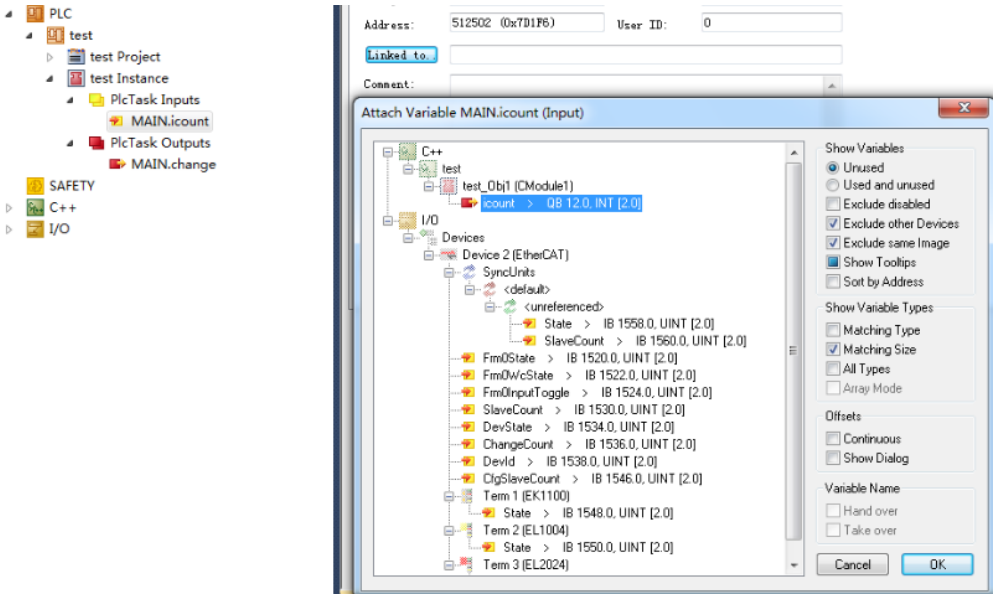
编译好后消息窗口显示如下：

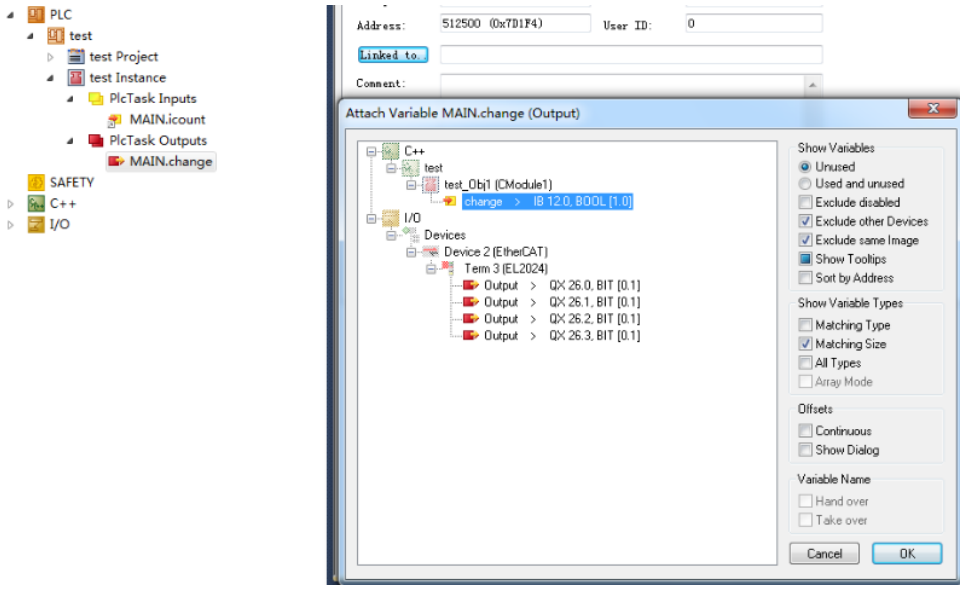


（5）清除之前 C++项目中变量链接

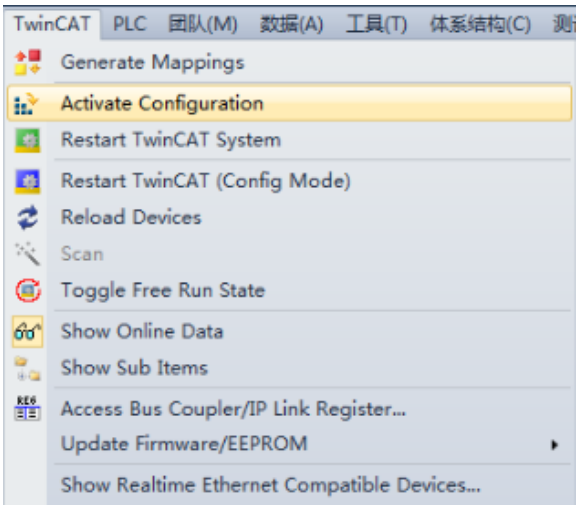


（6）分别把 PLC 程序中两个变量链接到 C++程序中两个变量上去。

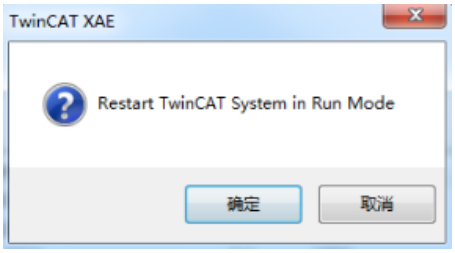




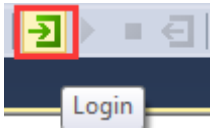
（7）打开菜单栏 TwinCAT，选择 Activate Configuration 把配置下载到控制器中，



（8）提示切换到运行模式点击确定。



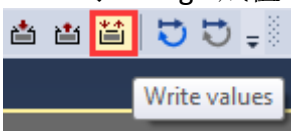
（9）在工具栏中点击“Login”把程序在线。



（10）随后点击“Start”把程序运行，

（11）在 change 类型旁边的准备值点成 TRUE，如图

（12）工具栏找到“Write values”对 change 赋值



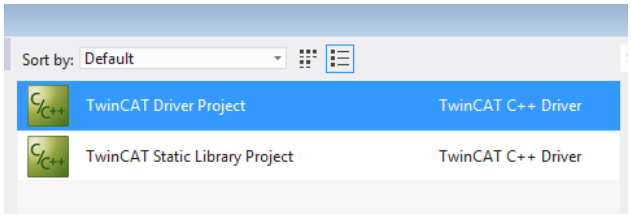
最终就可以观察到 icount 在不断做累加

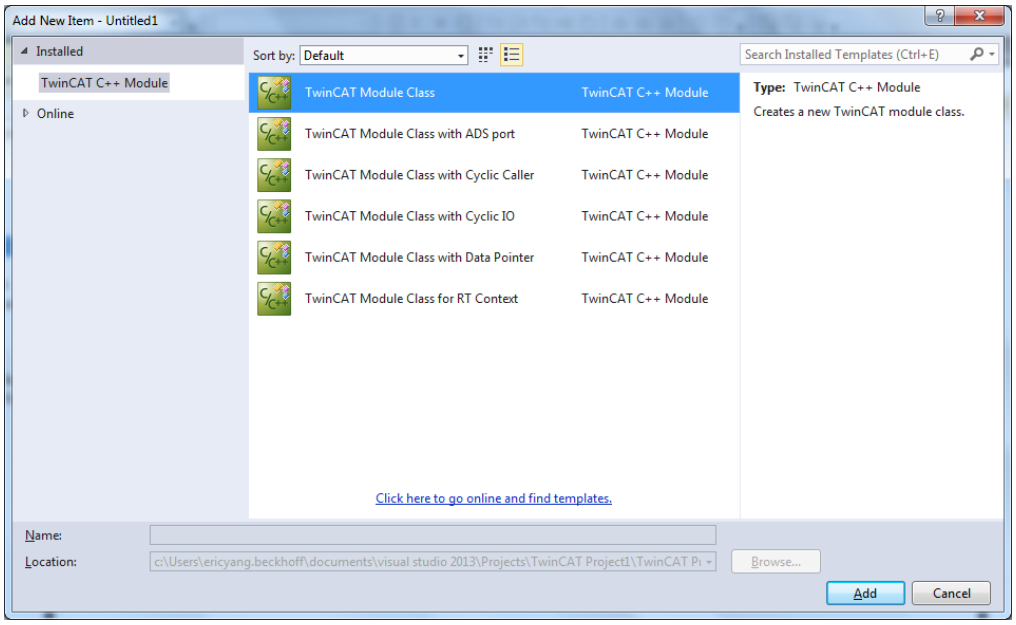
6. PLC 调用 C++模块

TC3 中的 C++开发除了可以直接指定 Task 循环执行（Call by task）之外，当然还可以发布成一个模块直接给另一个模块调用，例如 PLC，因此这篇会介绍如何用 PLC 调用 C++创建的模块。

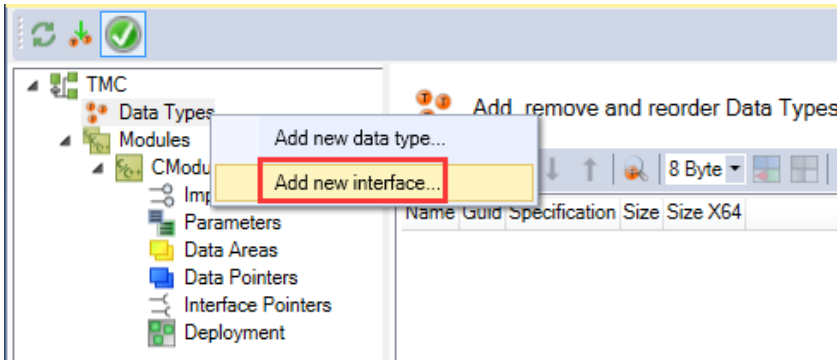
6.1 创建 C++模块步骤

（1）新建 C++项目 TwinCAT Driver Project，并且选择 TwinCAT Module Class 点击 Add





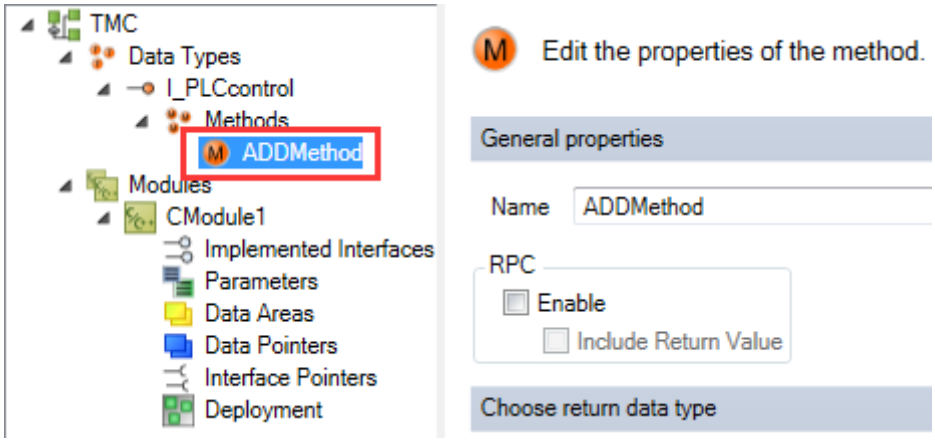
（2）打开 TMC 文件，首先建立一个接口，右键 Data Types 点击 Add new interface



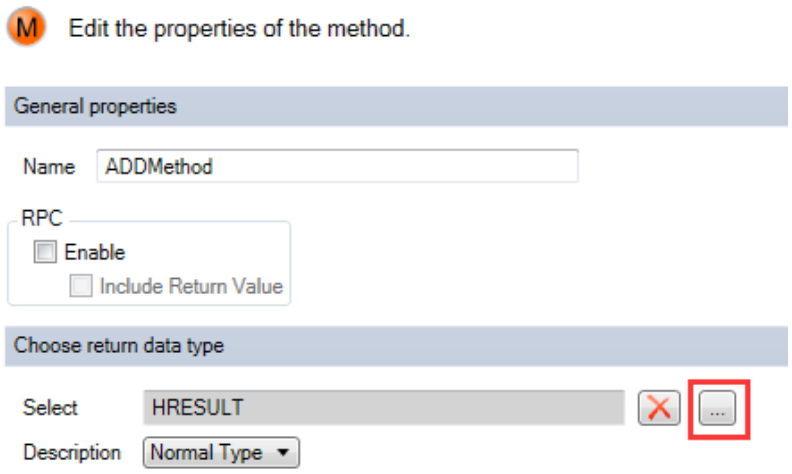
（3）随后在新建接口下新增一个方法，并且修改接口名字和 2 个方法的名字

例如：接口名：I_PLCcontrol

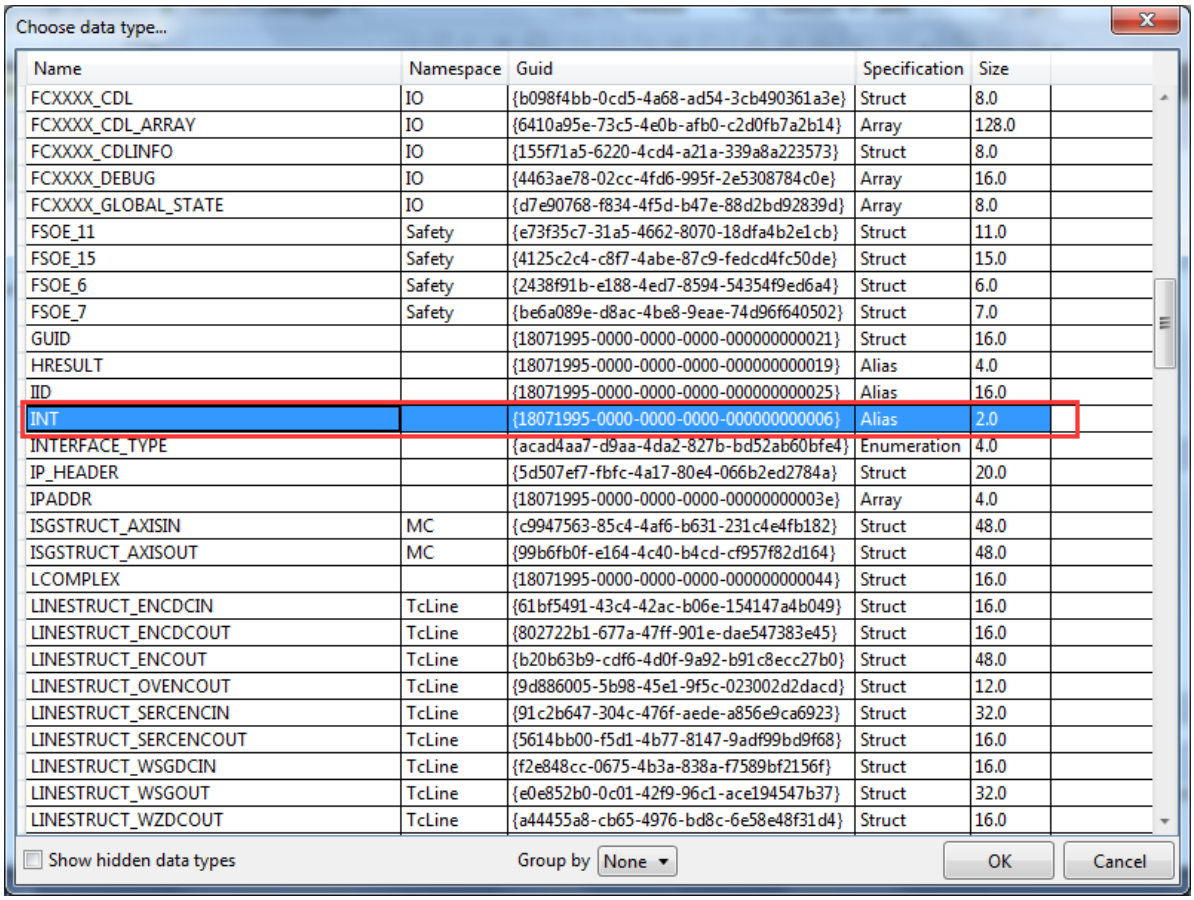
方法名：ADDMethod



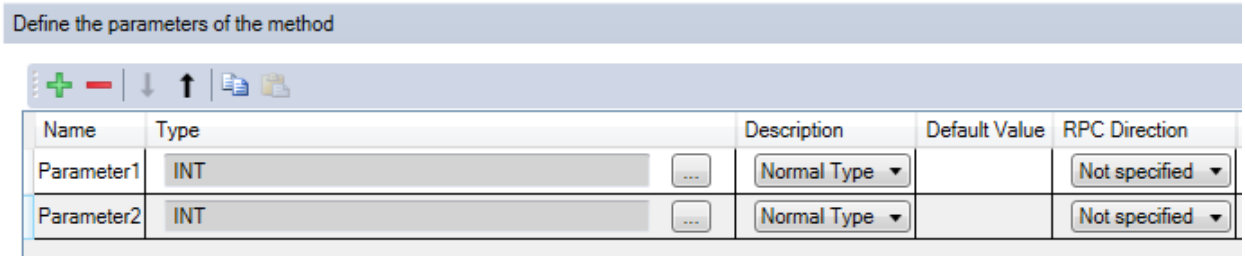
（4）分配此方法的返回类型是 INT 类型，点击下图中红色方块



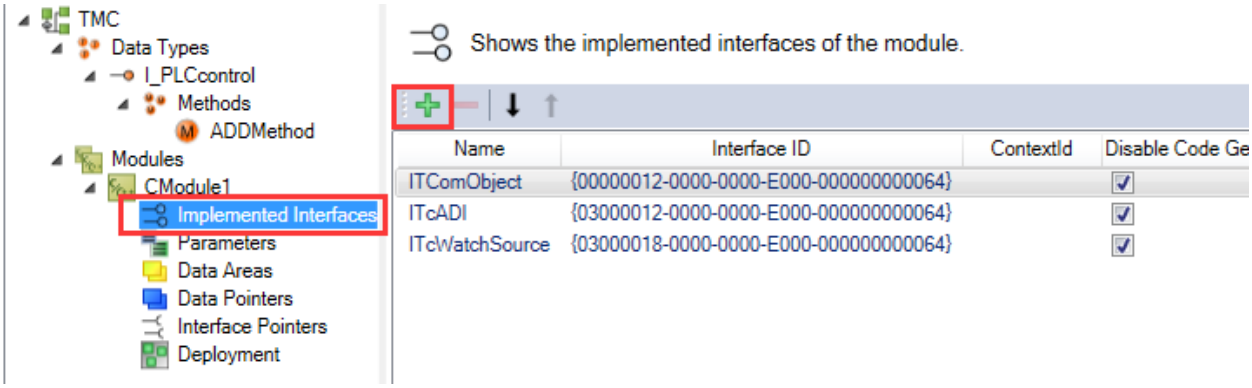
（5）找到 INT 类型点击确定



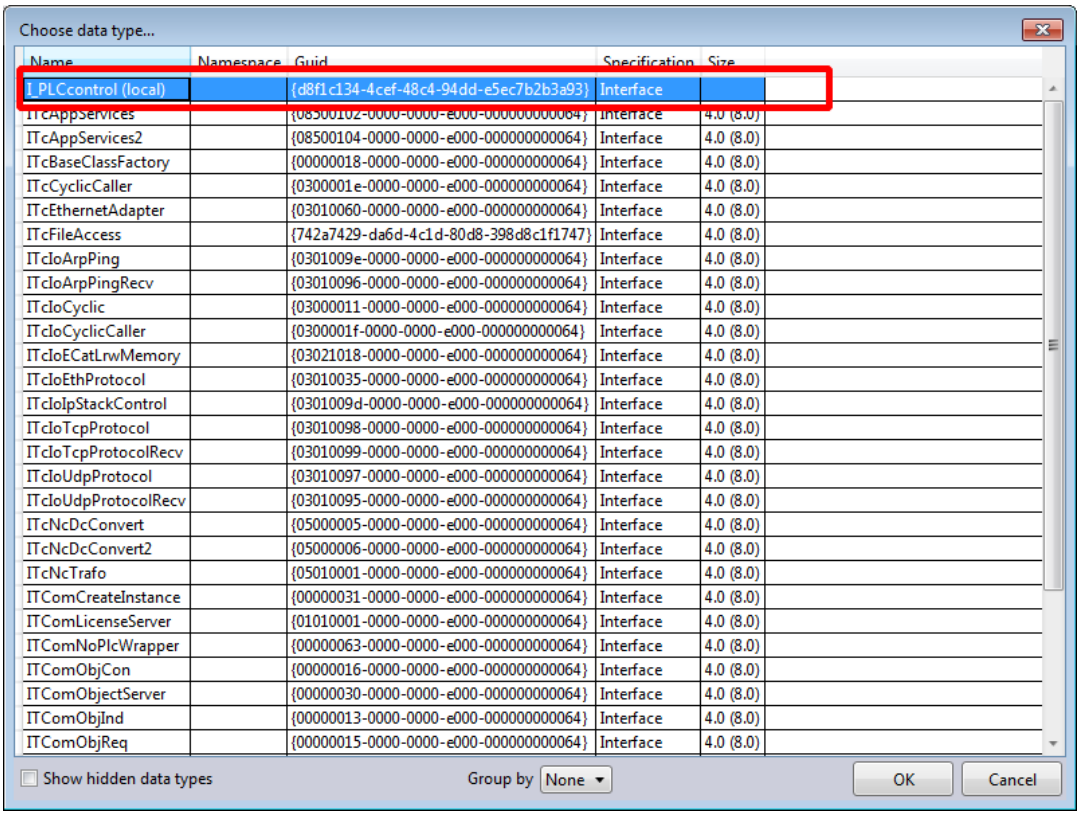
（6）在 ADDMethod 中添加 2 个行参，分别是 Parameter1 和 Parameter2，类型都是 INT 类型



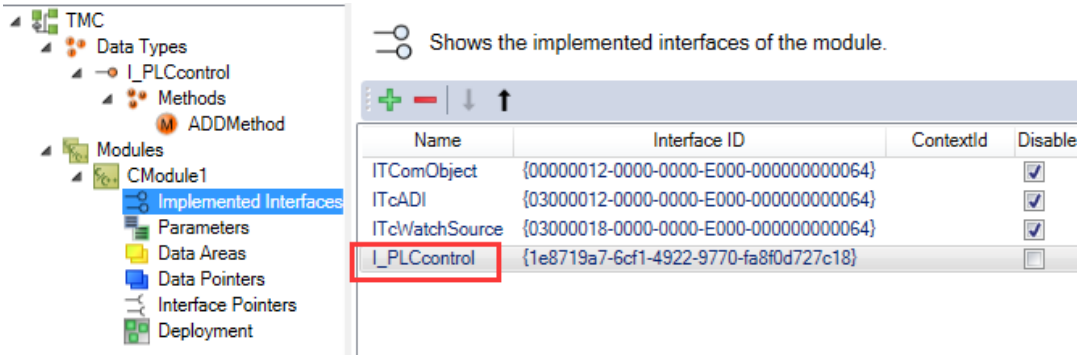
（7）接口和方法创建好后，对此接口进行注册实现，选择 implemented interfaces，点击加号



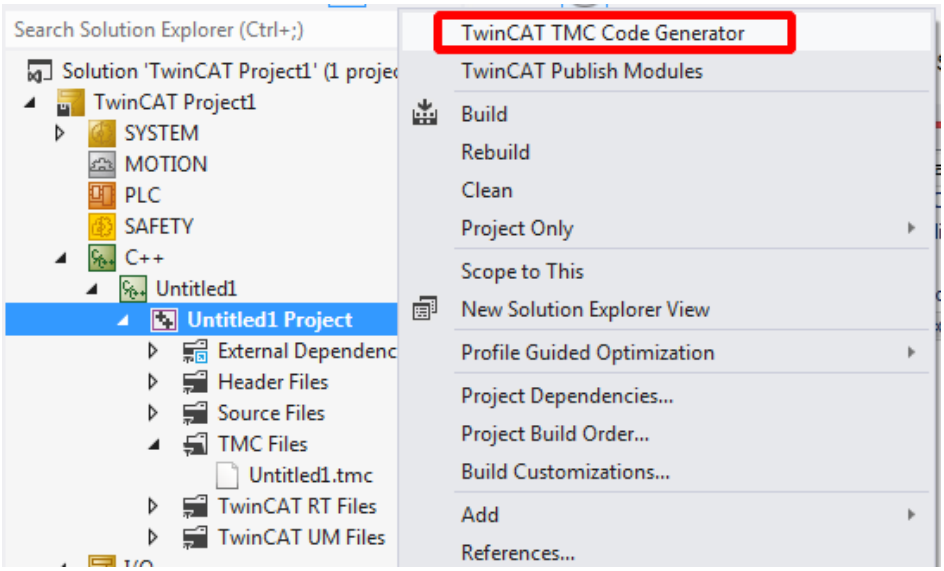
（8）找到刚才创建的 I_PLCcontrol 进行添加



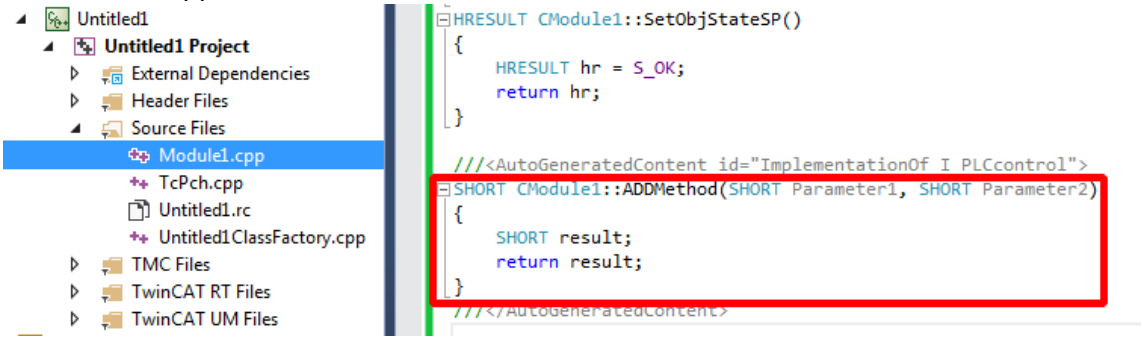
随后就出现在了 implemented interfaces 栏中



（9）TMC 编辑完毕后，右键 C++项目点击 TwinCAT TMC Code Generator



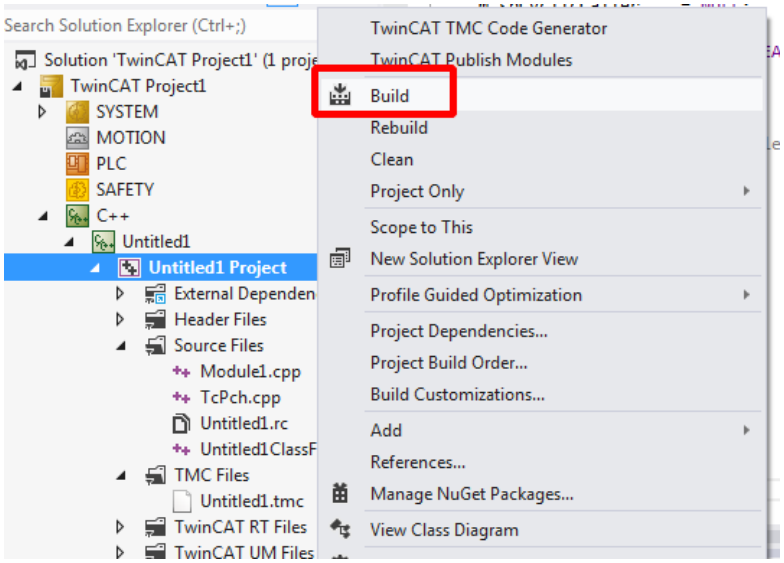
在 Module1.cpp 中就可以看到新增的接口实现可以写代码

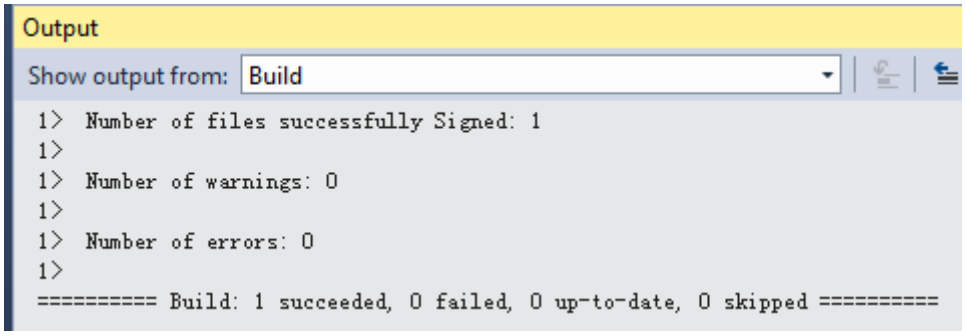


（10）写一个简单的算法，2 个行参 Parameter1+Parameter2，之后得到的结果给到返回类型

```
///<AutoGeneratedContent id="ImplementationOf_I_PLCcontrol">
SHORT CModule1::ADDMethod(SHORT Parameter1, SHORT Parameter2)
{
    SHORT result = Parameter1 + Parameter2;
    return result;
}
```

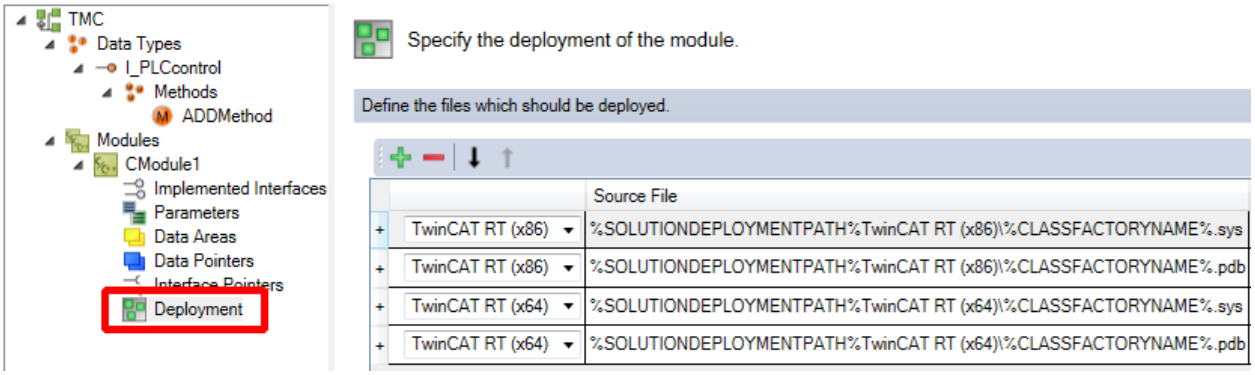
（11）写好实现代码后右键项目进行编译检查是否有错。



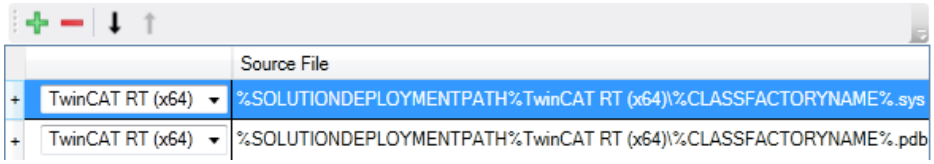


（12）编译没有错后我们可以支持把 C++写好的代码进行发布，编译成二进制驱动文件，也就是 TcCOM，这种类似 COM 组件的方式可以直接复制到没有完整版 VS 的电脑中使用，优点是反复利用 TcCOM 组件，保护源代码等等

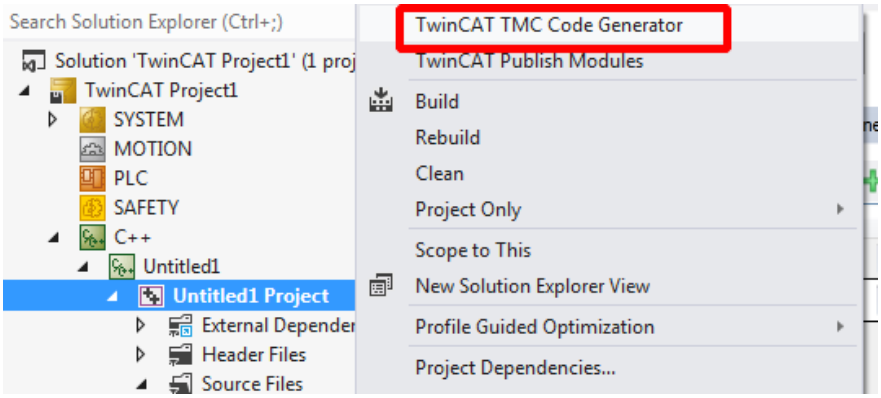
重新打开 TMC，选择 Deployment，选择需要发布的平台（32 位还是 64 位）



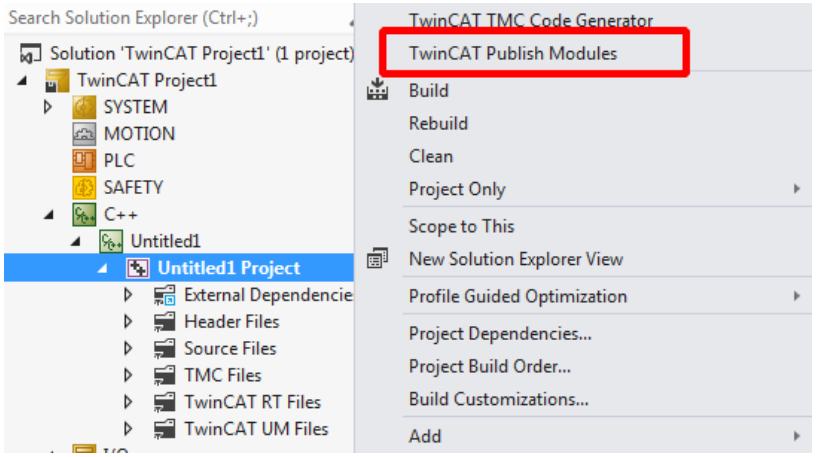
（13）例如我电脑是 64 位的，那我就可以只导出 64 位模块，把 32 位删除



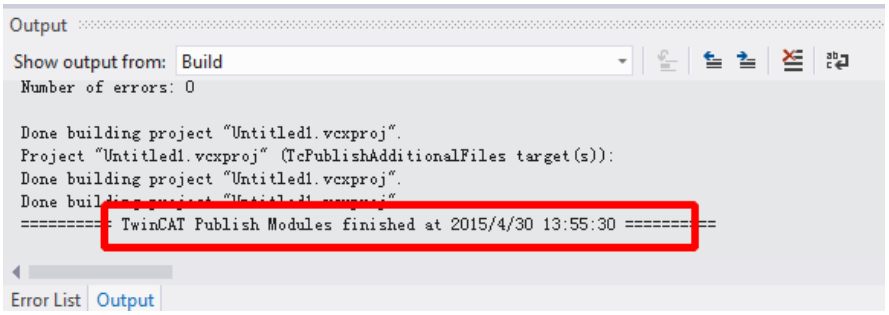
（14）重新更新下 TMC 文件



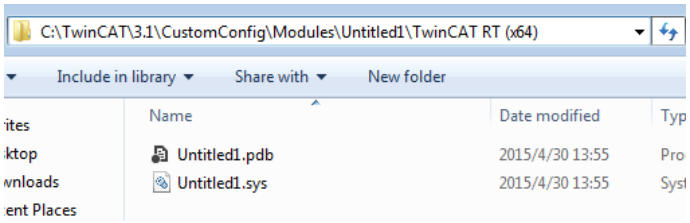
（15）随后右键 C++项目选择 TwinCAT Publish Modules 进行模块导出



消息窗口中提示模块导出成功



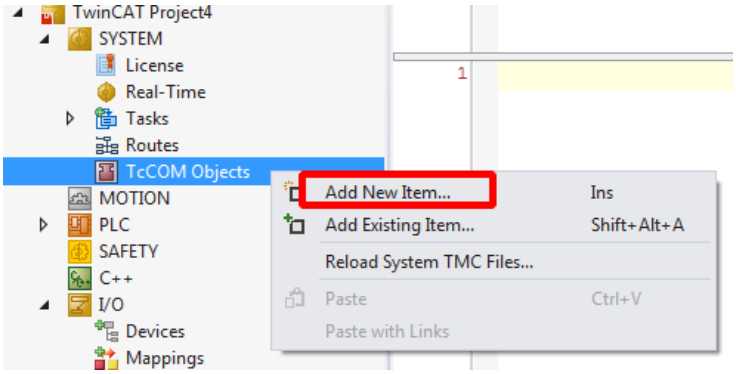
（16）我们也可以通过路径 C:\TwinCAT\3.1\CustomConfig\Modules 找到导出的模块



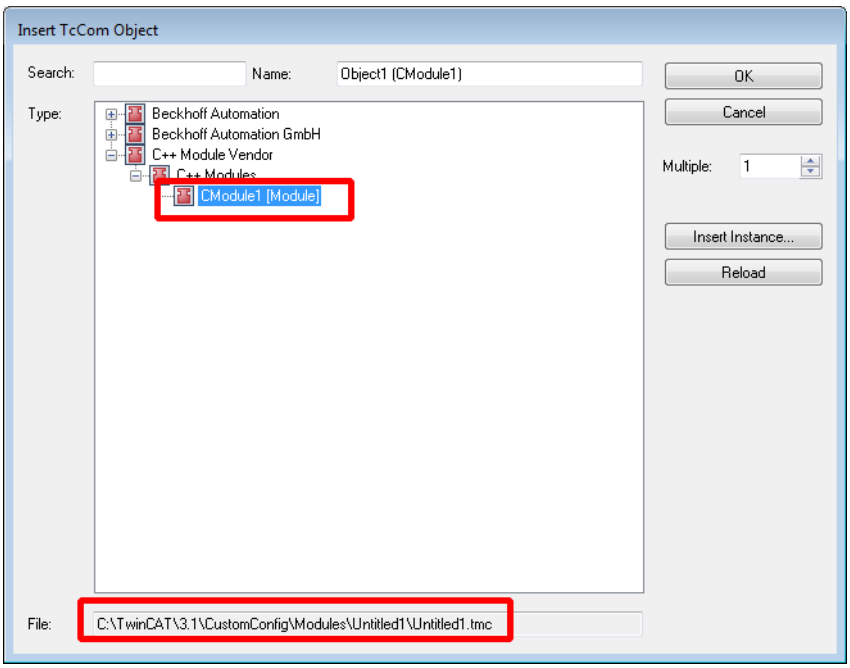
以上是 C++模块的创建

6.2 接下来就是创建 PLC 项目调用 C++模块

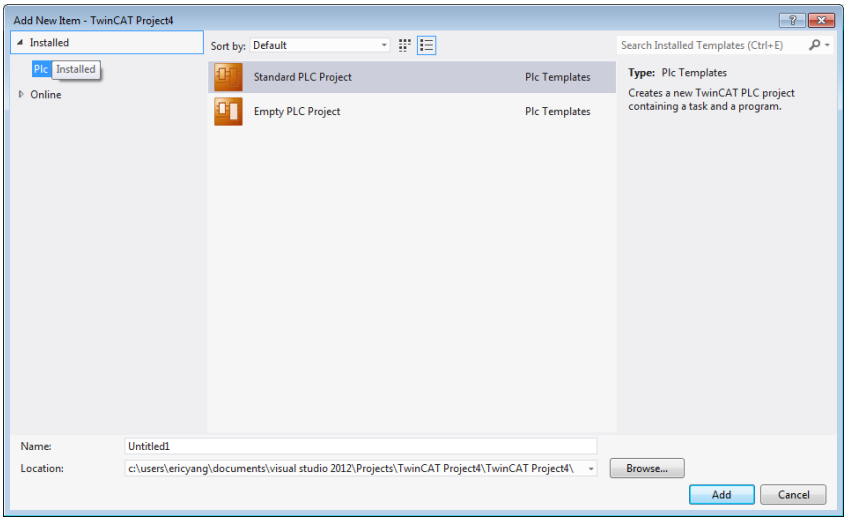
（1）右键 TcCOM Objects 添加新项目



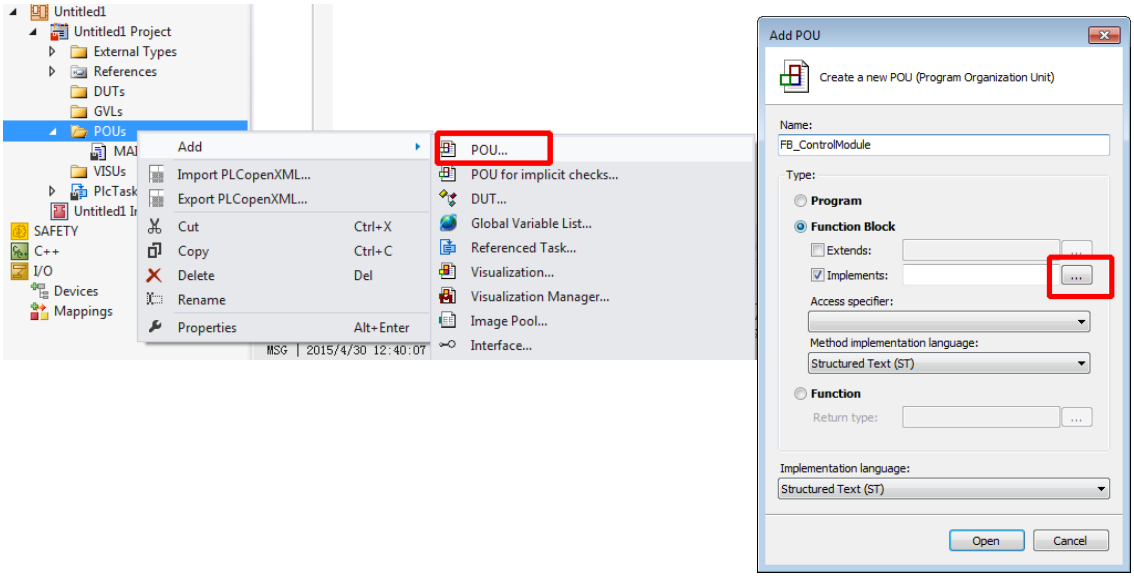
（2）选择 C++Modules 下添加之前发布的模块



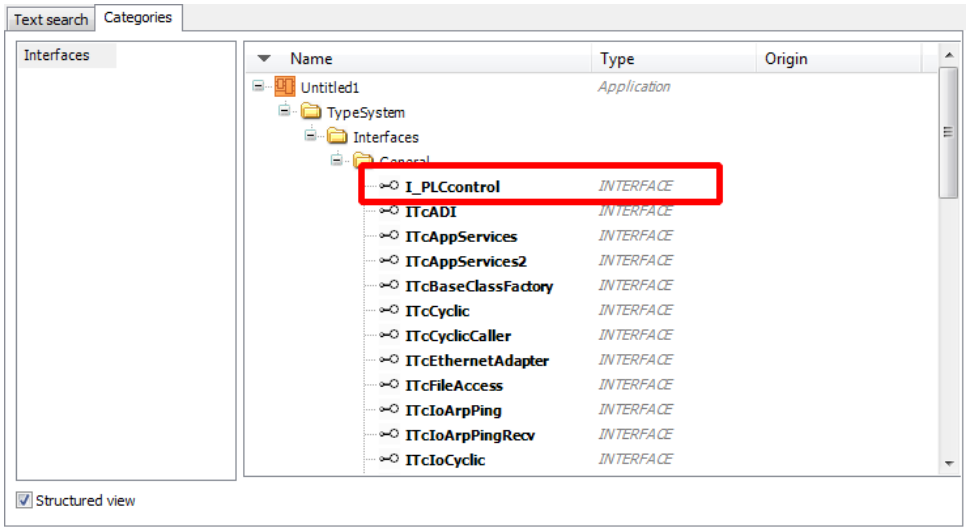
（3）新建 PLC 项目



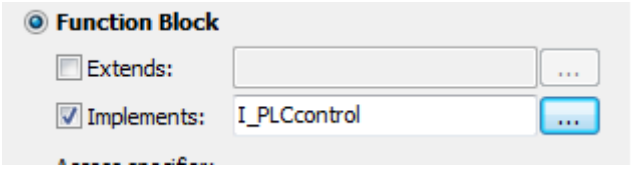
（4）添加功能块，修改功能块名称并把 implements 打勾，点击图片中红色选项。



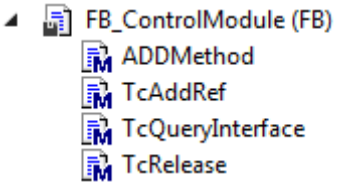
（5）从中找到之前在 C++项目创建好的接口 I_PLCcontrol



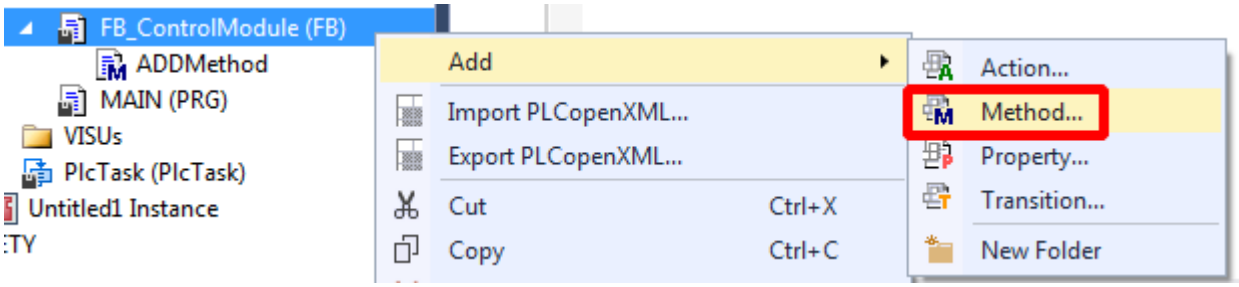
（6）选择好后点 OK



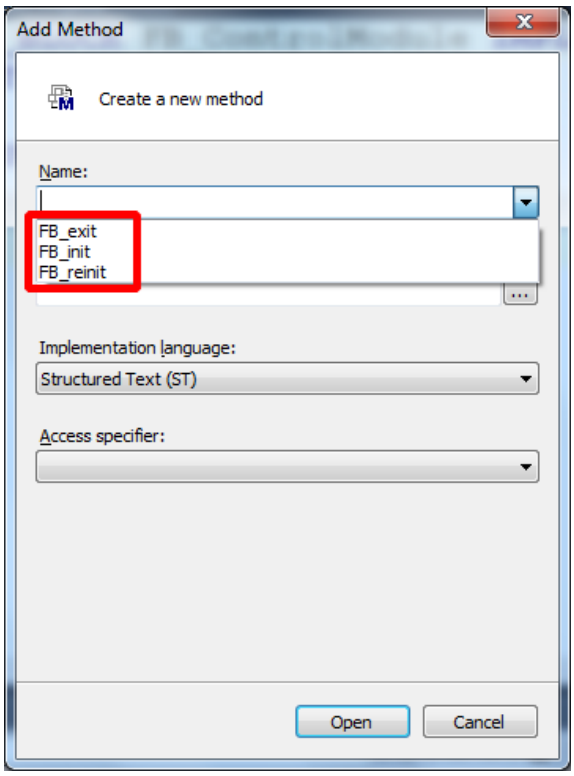
默认实现这个接口的功能块是空的，所以里面有很多需要自己写



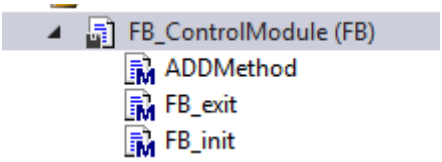
（7）首先把不必要的一些 Method 删除，只留下 ADDMethod，并且如果希望此功能块可以调用 C++中 I_PLCcontrol，且可执行此接口中的方法，就必须编写接口查找的代码，而且此代码必须是执行功能块前就完成，因此我们需要用到类似高级语言中构造函数的功能，在 TC3-PLC 中也有着类似的功能，右键功能块添加 Method



（8）在下拉框中可以发现，TC3 中已经有 3 个特殊方法供我们使用，其中 FB_init 就是构造函数，FB_exit 就是析构函数，我们依次添加这 2 个方法



（9）接下来开始对功能块中 3 个方法编写代码，以达到可以访问 C++接口，并且使用接口下的方法目的



（10）接下来写简单的代码，以下代码仅供参考，截图如下：

FB_ControlModule

```
FUNCTION_BLOCK FB_ControlModule
VAR
    {attribute 'TcInitSymbol' := ''}
    oidInstance: OTCID;
    ip_PlcControl: I_PLCcontrol;
END_VAR
```

FB_ControlModule.ADDMethod

```
{attribute 'checksuperglobal'}
METHOD ADDMethod : INT
VAR_INPUT
    Parameter1  : INT;
    Parameter2  : INT;
END_VAR
```

```
IF ip_PlcControl <> 0 THEN
    ADDMethod := ip_PlcControl.ADDMethod(Parameter1:=Parameter1 , Parameter2:=Parameter2 );
END_IF
```

FB_ControlModule.FB_exit*

```
METHOD FB_exit : BOOL
VAR_INPUT
    bInCopyCode : BOOL; // if
END_VAR
```

```
IF (ip_PlcControl <> 0) THEN
    ip_PlcControl.TcRelease();
    ip_PlcControl := 0;
    FB_exit := TRUE;
ELSE
    FB_exit := FALSE;
END_IF
```

FB_ControlModule.FB_init

```
METHOD FB_init : BOOL
VAR_INPUT
    bInitRetains : BOOL; // if TRUE, the retain variables are initialized (warm start / cold start)
    bInCopyCode : BOOL;  // if TRUE, the instance afterwards gets moved into the copy code (online change)
END_VAR
VAR
    iResult: DINT;
    ipObjServer: ITComObjectServer;
END_VAR
```

```
IF ip_PlcControl = 0 THEN
    iResult := FW_ObjMgr_GetObjectServer(_AppInfo.ObjId, ADR(ipObjServer));
    IF ipObjServer <> 0 AND iResult = 0 THEN
        IF (ipObjServer.TcQueryObjectInterface(oidInstance, IID_I_PLCcontrol, ADR(ip_PlcControl))) = 0 THEN
            FB_init := TRUE;
        ELSE
            FB_init := FALSE;
        END_IF
        ipObjServer.TcRelease();
    ELSE
        FB_init := FALSE;
    END_IF
END_IF
```

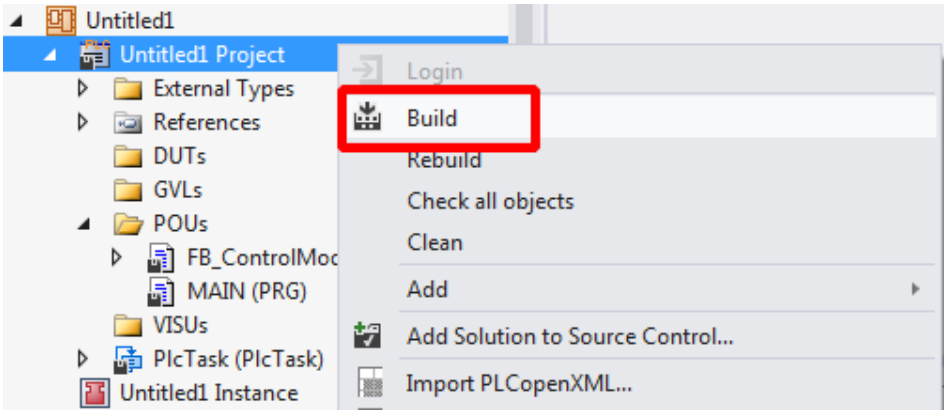
（11）在主程序中编写调用 C++中方法的代码

MAIN

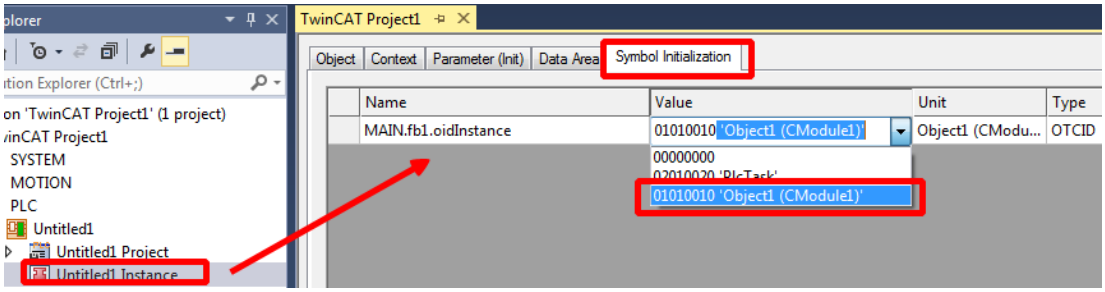
```
PROGRAM MAIN
VAR
    fb1 : FB_ControlModule;
    in1 , in2 : INT;
    out1: INT;
END_VAR
```

```
out1:=fb1.ADDMethod(Parameter1:=in1 , Parameter2:=in2 );
```

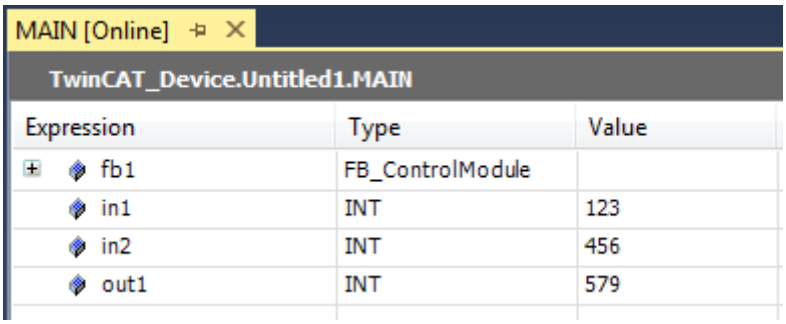
（12）右键 PLC 项目进行编译。



（13）为了把此功能块绑定 C++实例模块，双击 PLC 项目的 instance，选择 symbol initialization，把 value 设置为所调用的 C++模块



（14）active configuration 激活配置并且下载程序，可以发现 in1 赋值 123，in2 赋值 456，得到 out1 的结果自动就计算出是 in1 和 in2 的和为 579



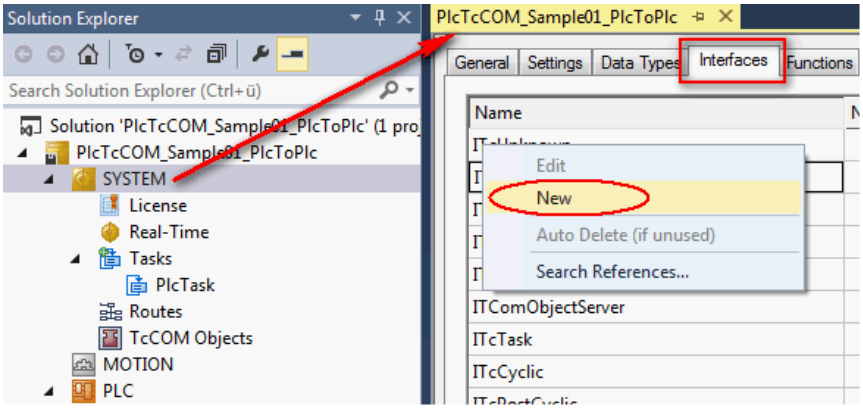
功能成功实现

7. C++调用 PLC 模块

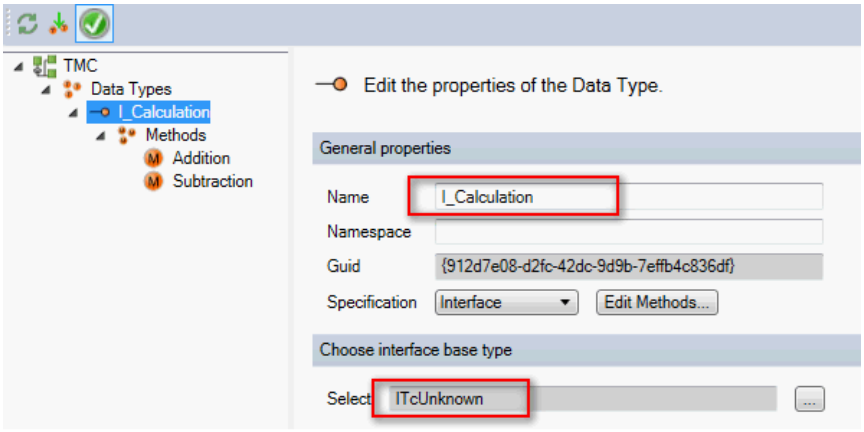
C++模块可以被 PLC 调用，当然也可以调用 PLC 模块，这篇会介绍如何用 C++调用 PLC 模块。

6.3 创建 PLC 模块步骤

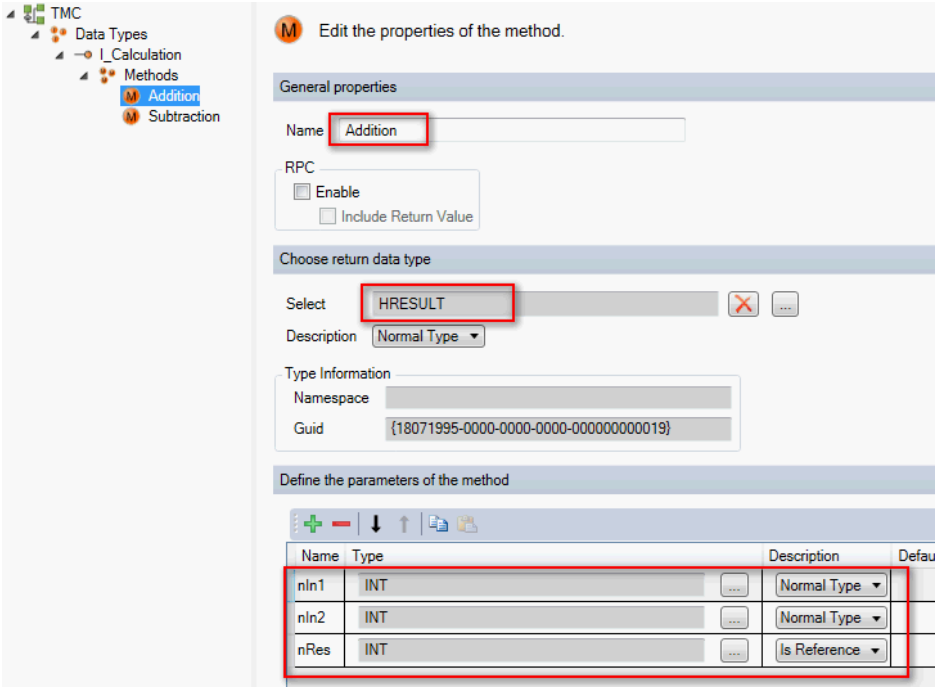
（1）新建项目，在 SYSTEM 中选择 Interfaces，对着空白处右键新建 New



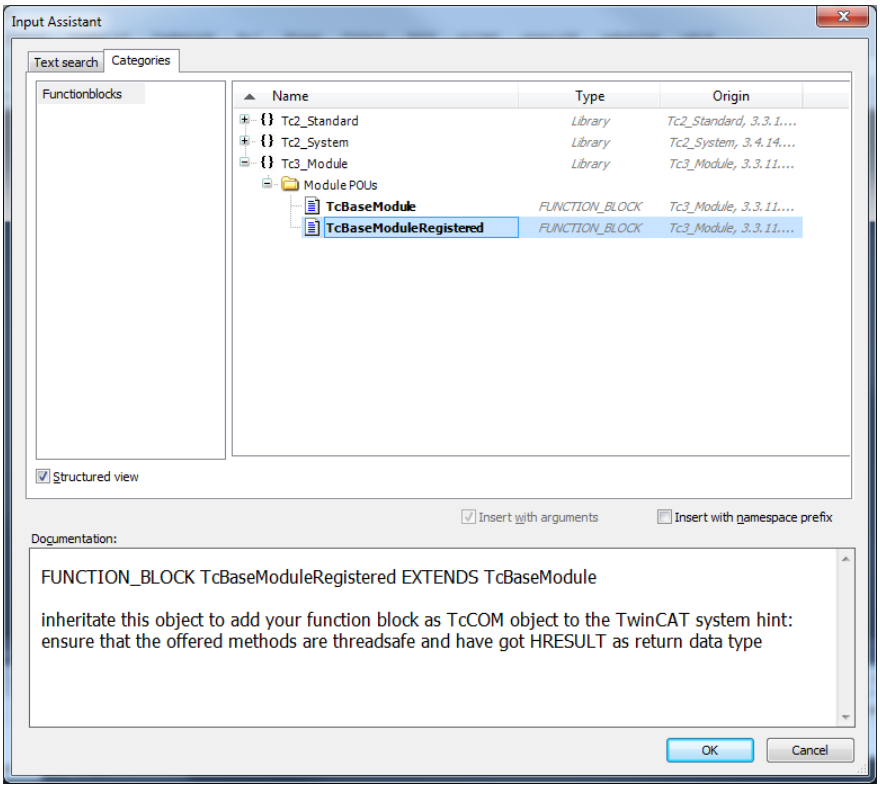
（2）新建一个接口，取名 I_Calculation，并选择此接口继承 ITcUnknown

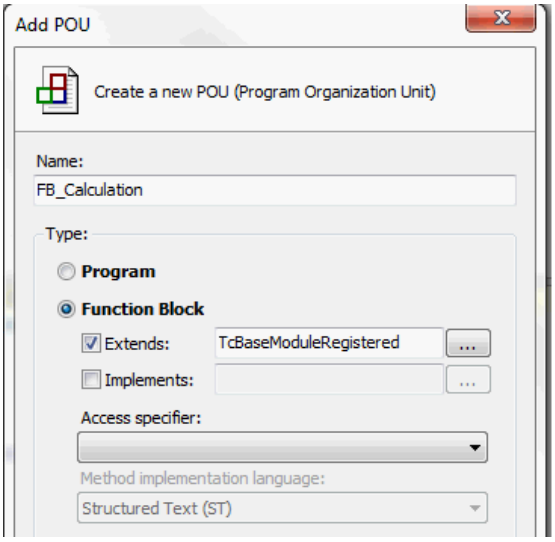


（3）为此接口创建两个方法 Addition 和 Subtraction，分别设置两个方法的返回类型都为 HRESULT，以及分别为两个方法定义 3 个参数变量 nIn1,nIn2,nRes，类型都为 INT，并把 nRes 改为 is Reference，方便传递出来，具体参考下图：



（4）创建一个功能块，扩展 Tc3_Module 中的功能块 TcBaseModuleRegistered

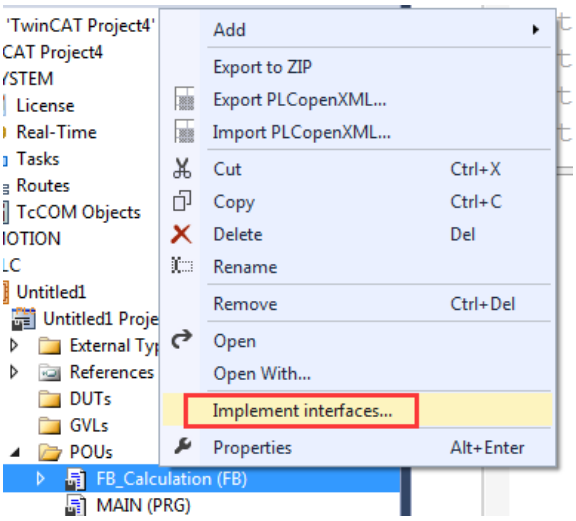




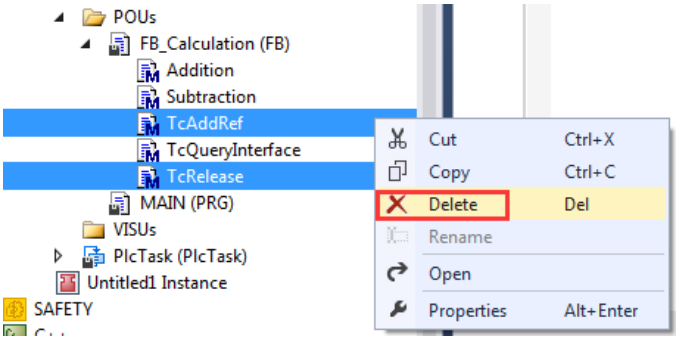
（5）创建好后，手动添加 attribute 和 IMPLEMENTS I_Calculation

```
4   {attribute 'c++_compatible'}
5   FUNCTION_BLOCK FB_Calculation EXTENDS TcBaseModuleRegistered IMPLEMENTS I_Calculation
6
7   VAR
8   END_VAR
9
```

（6）随后重新实现此接口

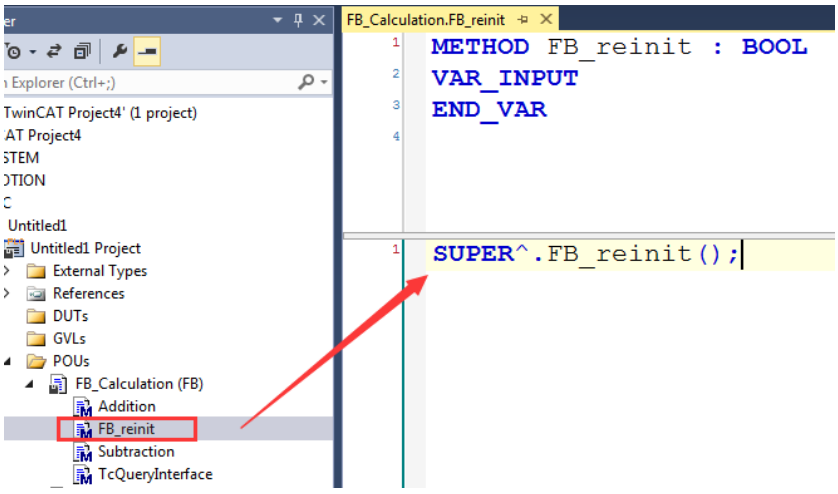


（7）你会发现会新增几个方法出来，我们保留其中三个，把 TcAddRef 和 TcRelease 手动删除

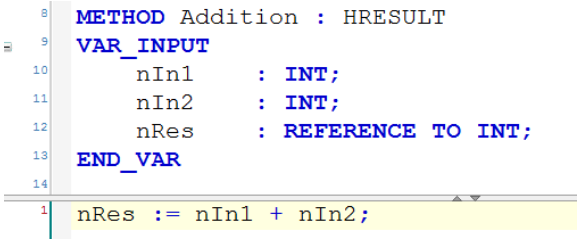


（8）接下来对各个方法写代码

FB_reinit 代码：



Addition 代码：（编写了加法的算法）



Subtraction 代码：（编写了减法的算法）

```
METHOD Subtraction : HRESULT
VAR_INPUT
    nIn1    : INT;
    nIn2    : INT;
    nRes    : REFERENCE TO INT;
END_VAR

nRes := nIn1 - nIn2;
```

TcQueryInterface 代码：（编写了此模块可被查询的规则）

FB_Calculation.TcQueryInterface

```
//{error 'add method implementation or delete method to use base imple
METHOD TcQueryInterface : HRESULT
VAR_INPUT
    iid : REFERENCE TO IID;
    pipItf  : POINTER TO PVOID;
END_VAR
VAR
    ipCalc : I_Calculation;
END_VAR
```

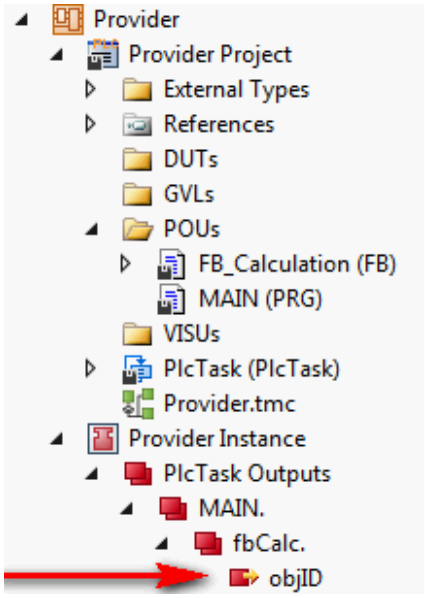
```
IF GuidsEqual(pGuidA:=ADR(iid) ,
              pGuidB:=ADR(TC_GLOBAL_IID_LIST.IID_I_Calculation) )THEN
    ipCalc := THIS^;// cast to interface pointer
    pipItf^ :=ITCUNKNOWN_TO_PVOID(ipCalc);
    TcAddRef();
    TcQueryInterface := S_OK;
ELSE
    TcQueryInterface := SUPER^.TcQueryInterface(iid, pipItf);
END_IF
```

Main 程序只需要对此功能块进行声明即可：

MAIN

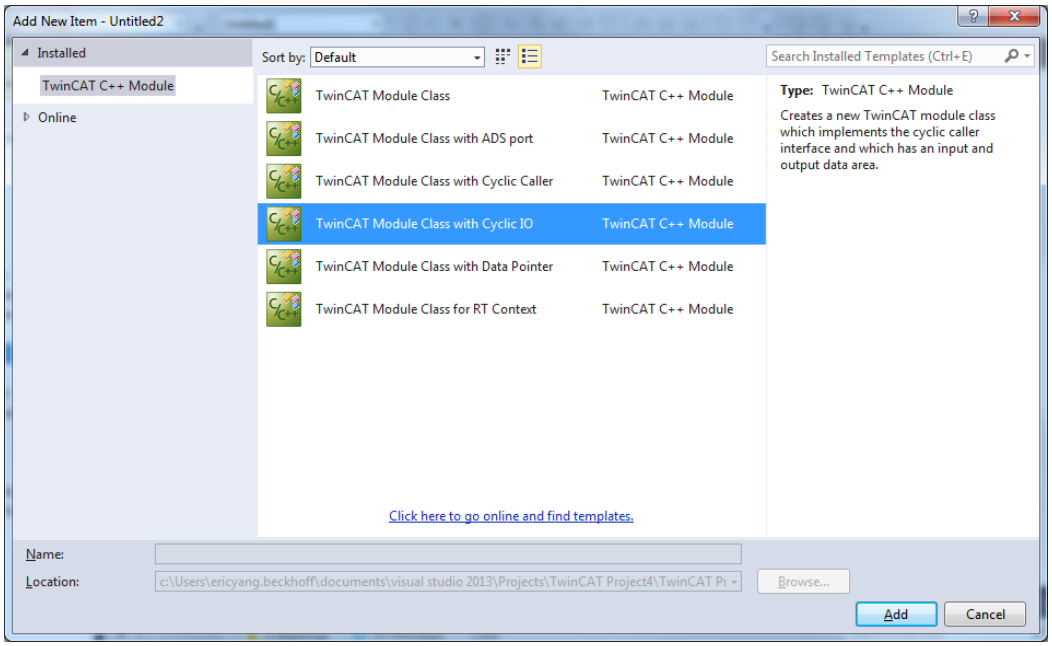
```
PROGRAM MAIN
VAR
    fbCalc : FB_Calculation(sObjectName:='MAIN.fbCalc' );
END_VAR
```

（9）代码全部写好，Bulid 成功后，在 Instance 中会又一个 objID 变量产生，此变量一会需要和 C++中的一个变量进行映射，这样 C++就可以知道所调用的 PLC 模块是哪一个

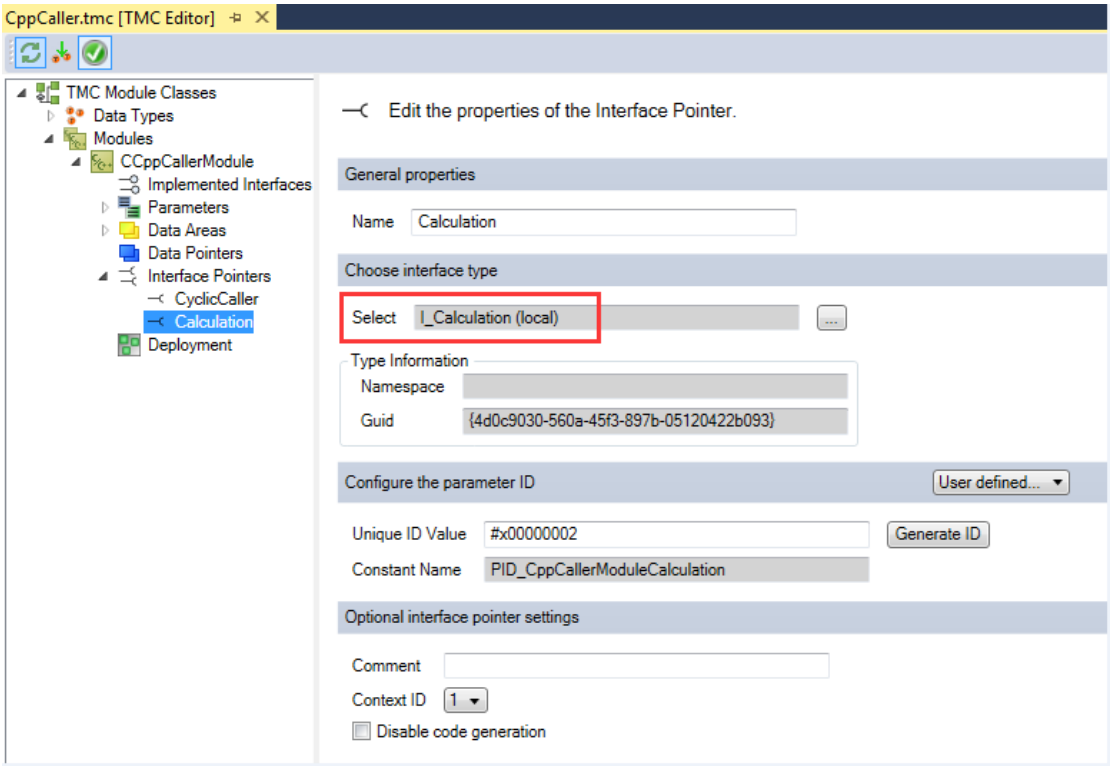


6.4 接下来就是创建 C++项目调用 PLC 模块

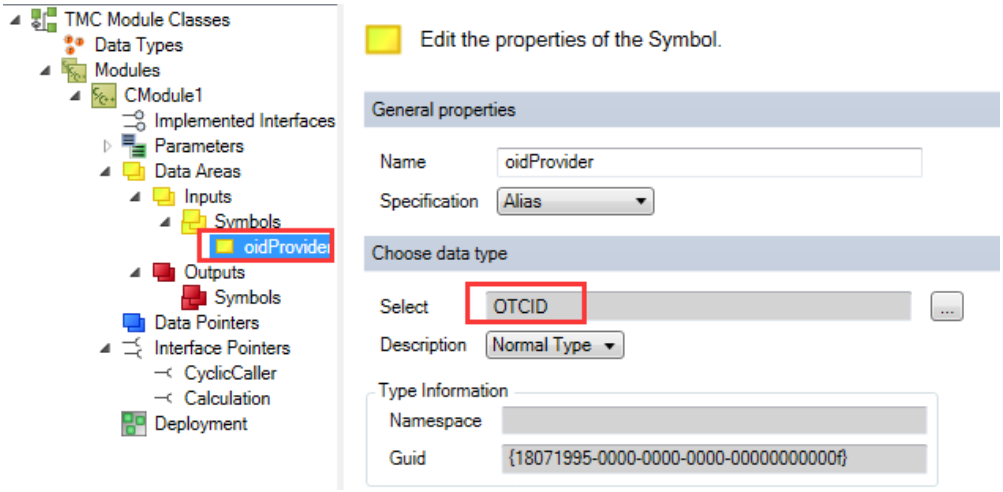
（1）首先创建 C++项目，可以选择 TwinCAT Module Class with Cyclic IO 模板



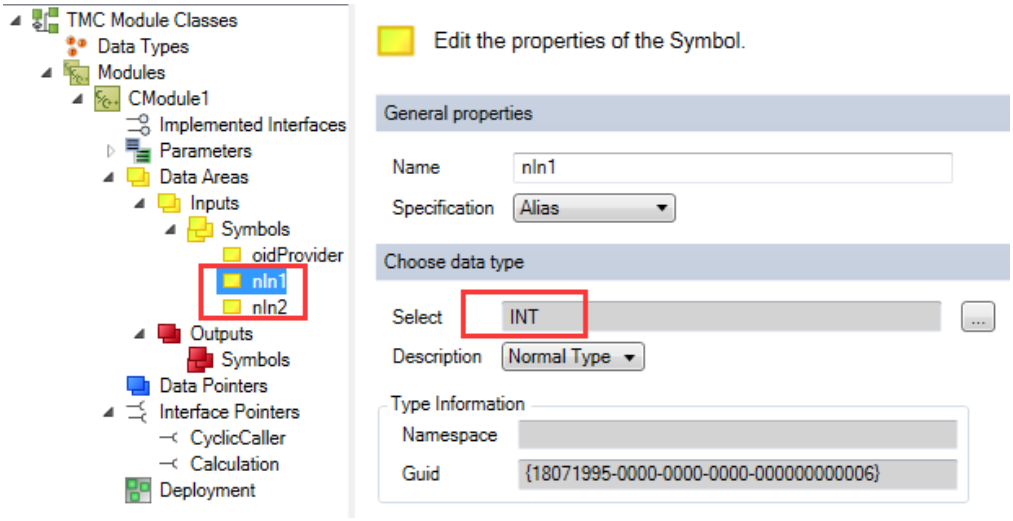
（2）打开 TMC 文件，在 Interface Pointers 中新建一个接口 Calculation，并且选择接口类型为 I_Calculation（之前在 SYSTEM 中已经创建）



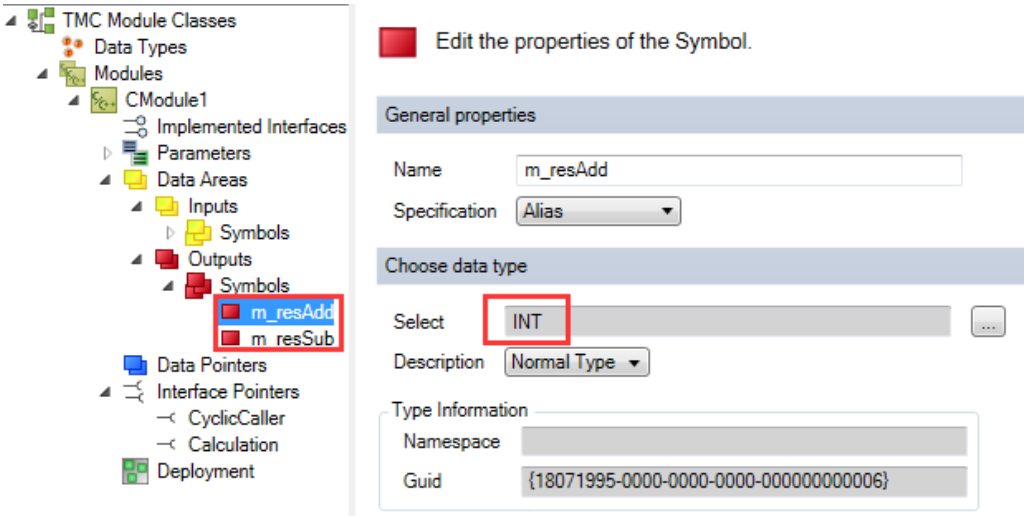
（3）紧接着在 Data Areas→Inputs 中创建一个变量 oidProvider，类型选择为 OTCID



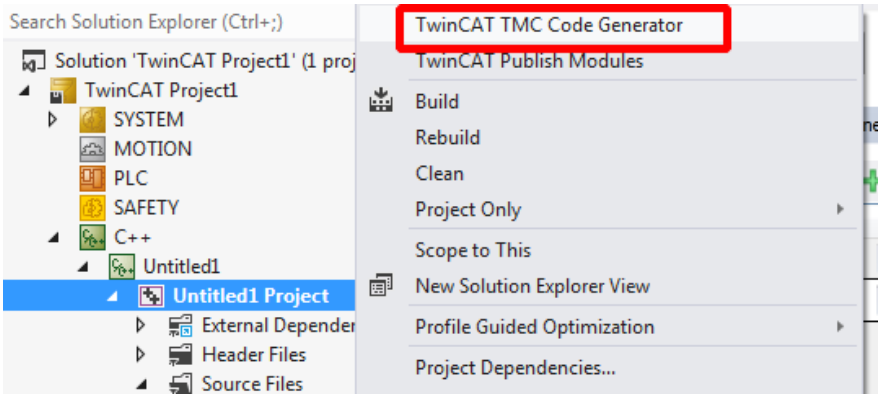
（4）再创建两个输入变量 nIn1 和 nIn2，类型都为 INT



（5）创建两个输出变量 m_resAdd 和 m_resSub，类型都为 INT



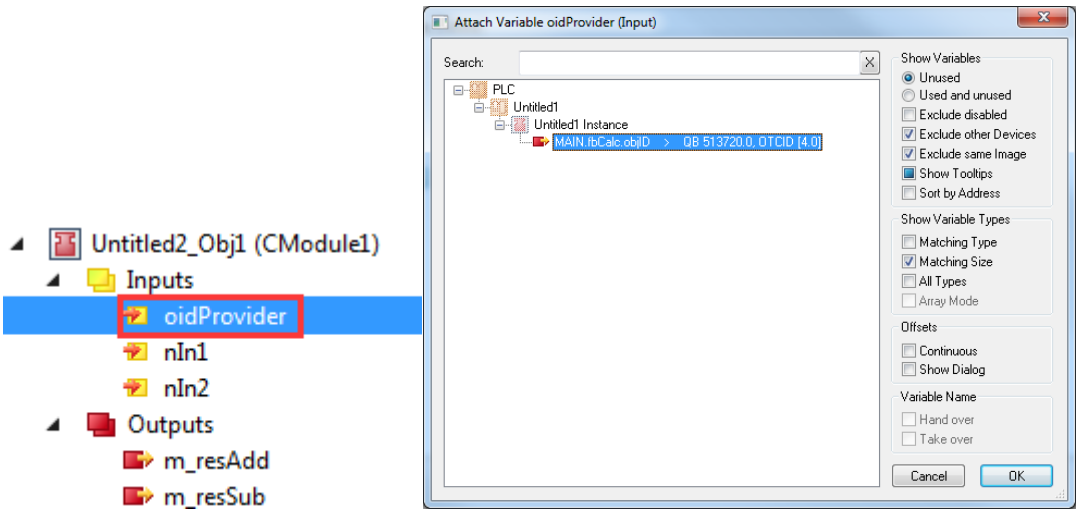
（6）生成下 TMC 文件



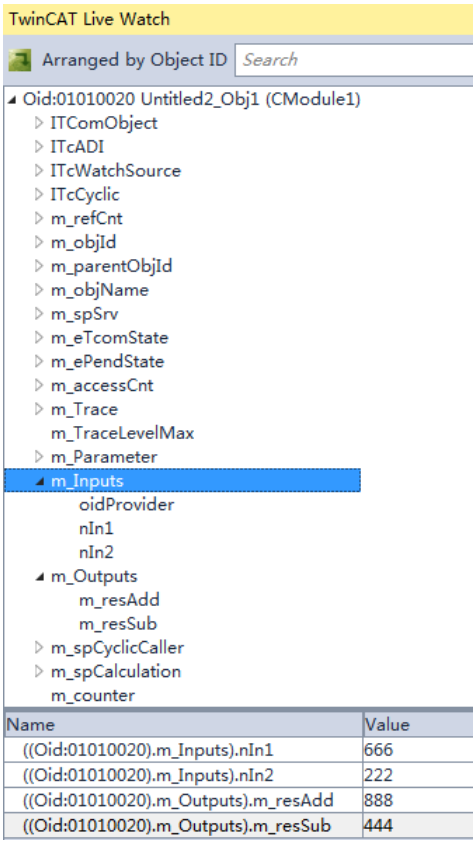
（7）在 CycleUpdate 中编写代码，查询 PLC 模块，以及调用 PLC 模块中 Addition 和 Subtraction 方法

```
HRESULT CModule1::CycleUpdate(ITcTask* ipTask, ITcUnknown* ipCaller, ULONG_PTR context)
{
    HRESULT hr = S_OK;

    // TODO: Replace the sample with your cyclic code
    if ((m_spCalculation == NULL) && m_Inputs.oidProvider != 0)
    {
        m_spCalculation.SetOID(m_Inputs.oidProvider);
        m_spSrv->TcQuerySmartObjectInterface(m_spCalculation);
    }
    if (m_spCalculation != NULL)
    {
        m_spCalculation->Addition(m_Inputs.nIn1, m_Inputs.nIn2, m_Outputs.m_resAdd);
        m_spCalculation->Subtraction(m_Inputs.nIn1, m_Inputs.nIn2, m_Outputs.m_resSub);
    }
    return hr;
}
```

（8）编写完成，对 C++项目单独编译下，随后添加 instance，把其中 oidProvider 变量映射到 PLC 的 objID 变量中



（9）这样就完成了 C++调用 PLC 模块，利用 TwinCAT Live Watch 可以在线给两个输入变量赋值，可以发现输出变量调用 PLC 模块进行了计算并且给出了结果


TwinCAT Live Watch
Arranged by Object ID
Search
Oid:01010020 Untitled2_Obj1 (CModule1)
ITComObject
ITcADI
ITcWatchSource
ITcCyclic
m_refCnt
m_objId
m_parentObjId
m_objName
m_spSrv
m_eTcomState
m_ePendState
m_accessCnt
m_Trace
m_TraceLevelMax
m_Parameter
m_Inputs
oidProvider
nIn1
nIn2
m_Outputs
m_resAdd
m_resSub
m_spCyclicCaller
m_spCalculation
m_counter
Name
Value
((Oid:01010020).m_Inputs).nIn1
666
((Oid:01010020).m_Inputs).nIn2
222
((Oid:01010020).m_Outputs).m_resAdd
888
((Oid:01010020).m_Outputs).m_resSub
444

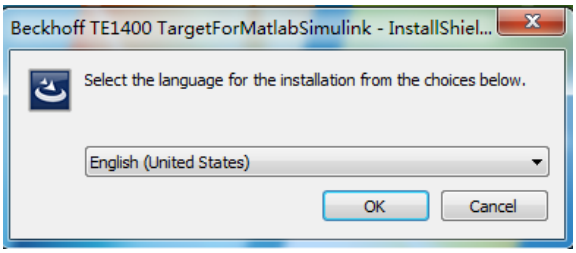
二、 TwinCAT3 中 Matlab-simulink 使用

1. Matlab-simulink 安装篇

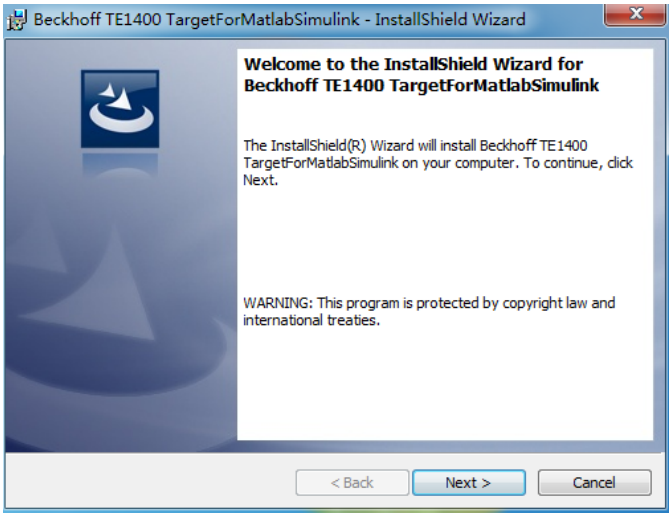
1.1 Matlab 和 TwinCAT3 配合使用有一些要求：
（1）推荐在安装过 VS 完整版的电脑上安装 Matlab
（2）推荐 Matlab®/Simulink® R2010b 以上版本进行安装
（3）首先安装 VS2010/2012/2013/2015 完整版以及 TC3，可以参考 C++安装篇
（4）其次安装 WDK 和相关配置，如果是 64 位操作系统还需要创建并安装测试证书，可以参考 C++安装篇

1.2 Matlab 安装
安装步骤省略，但建议用比较新版本的 Matlab（例如 2014a/b，2015a/b），当然 Matlab 版本必须高于 VS 版本，否则之后选择编译器的步骤会识别不到电脑中有 VS 的编译器。

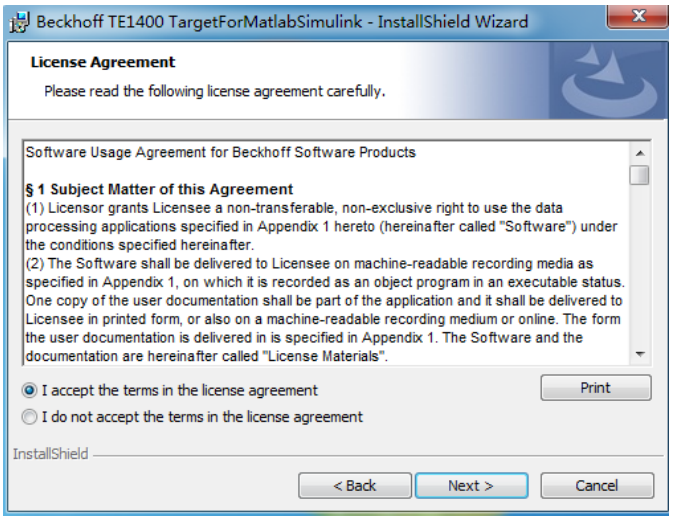
1.3 安装 TE1400-Target-for-Matlab-Simulink，下载链接：
http://www.beckhoff.com/english/download/tc3-download-te1xxx.htm?id=1957281419487556
TE1400 是 Matlab/Simulink 离线版的工具，可以不需要在 TwinCAT3 的工控机上安装 Matlab，在专门一台安装有 Matlab 的电脑上安装 TE1400 可以通过这个工具把 Simulink 工程文件转换成 TwinCAT3 可以直接调用的模块。
（1）下载好后开始安装，选择安装语言。



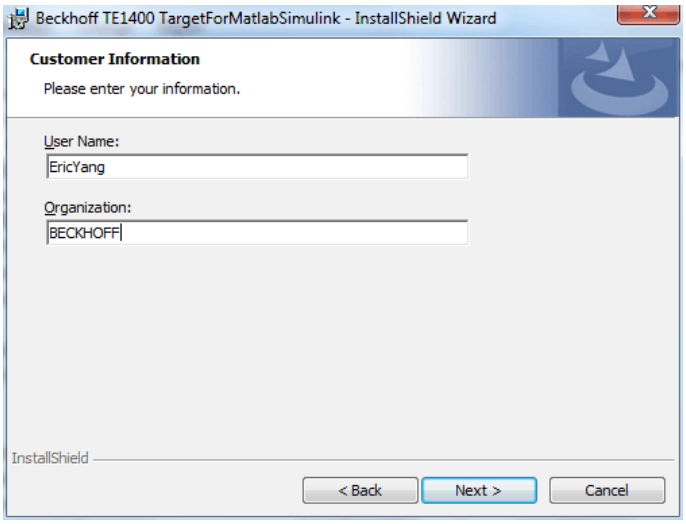
（2）点击 Next。



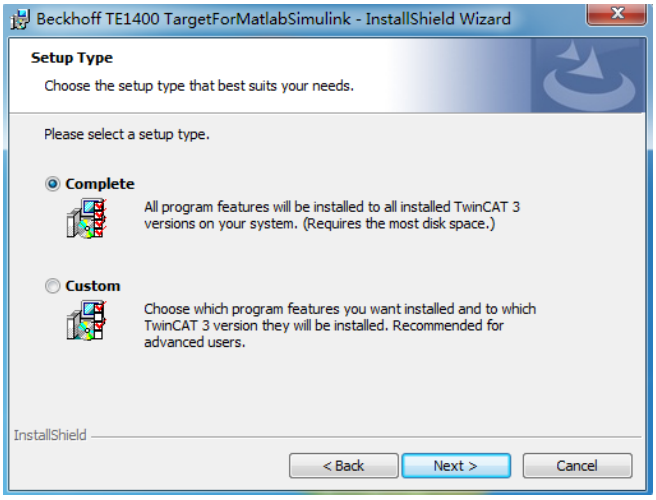
（3）选择 I accept the terms in the license agreement 后点击 Next。



（4）输入用户名和公司后点击 Next。

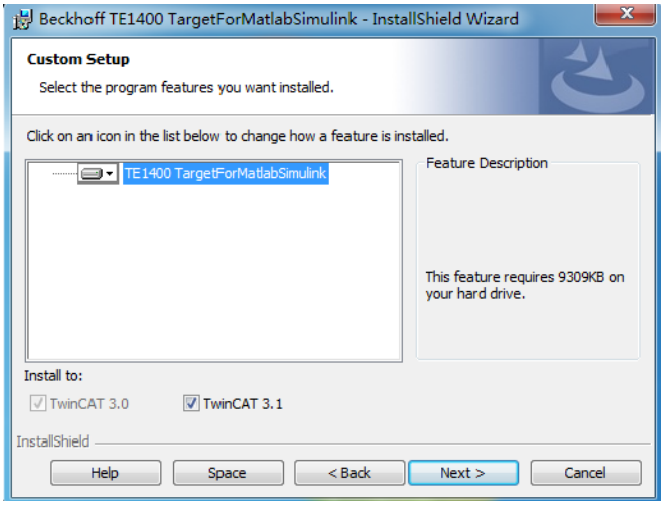


（5）选择 Complete 点击 Next

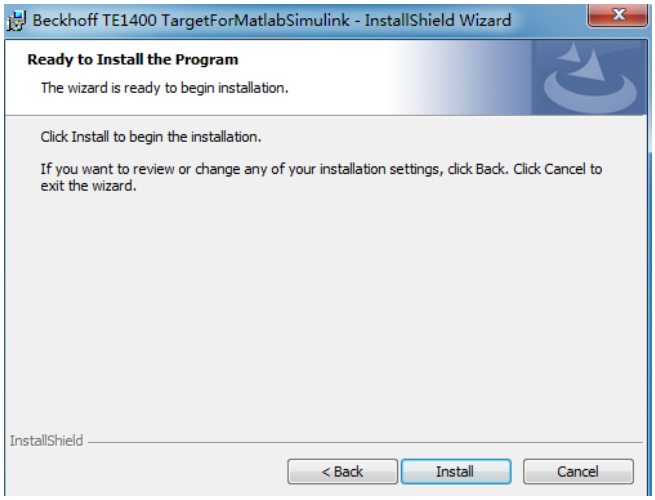


（6）如果选择 Custom 我们可以发现 TE1400 还是需要有 TC3 支持的，所以安装

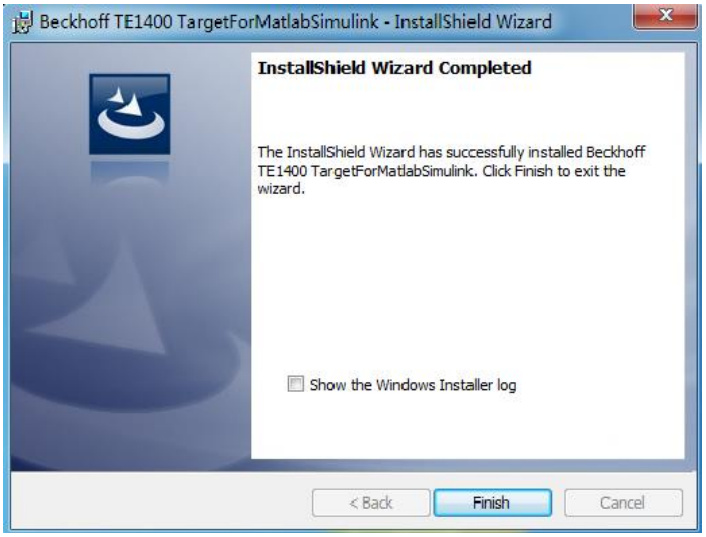
TE1400 的坏境必须有 TwinCAT3，如下图：



（7）随后点击 Install 就可以开始安装了。



（8）完成点击 Finish 即可，无需重启



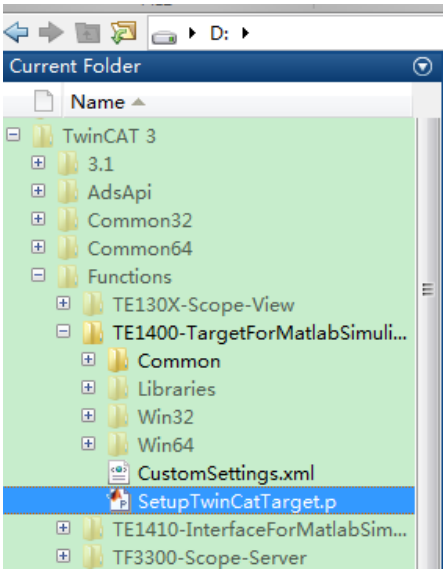
2. TE1400 使用

TE1400 作为 TwinCAT3 工程插件，其功能是利用 Simulink Coder 把 Simulink 中的模型生成实时 C/C++代码，并且进一步导出成自带输入输出接口的 TcCOM 模块，此模块可以在 TwinCAT3 项目中实例化，并且也可以对模块的参数进行修改，最终在 TwinCAT3 内核中实时执行。

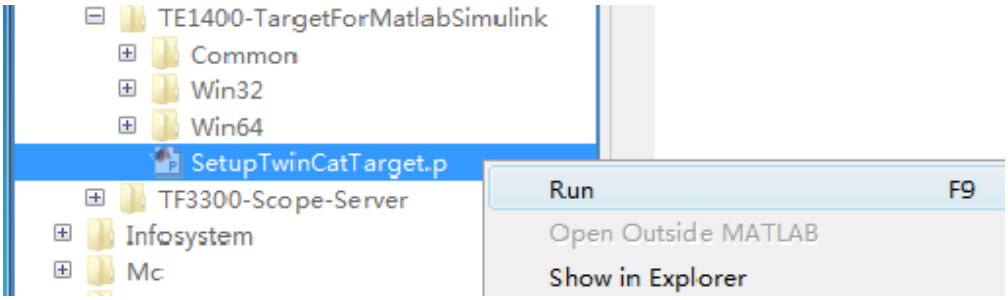
2.1 TE1400 生成模块 call by task 简介

（1）以管理员身份运行 Matlab

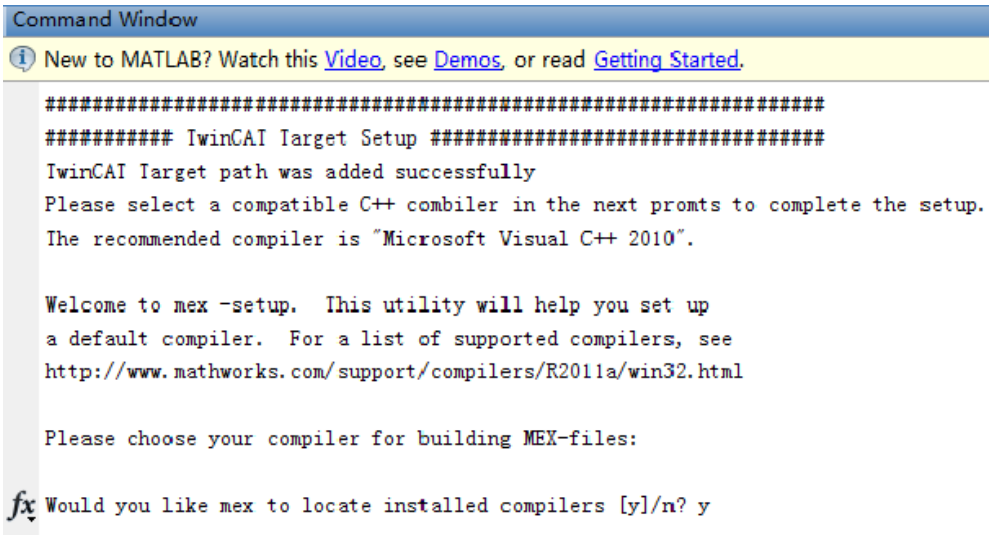
（2）登录后把当前文件夹改成 TwinCAT3 安装盘，找到 TwinCAT → Functions → TE1400 → SetupTwinCATTarget.p



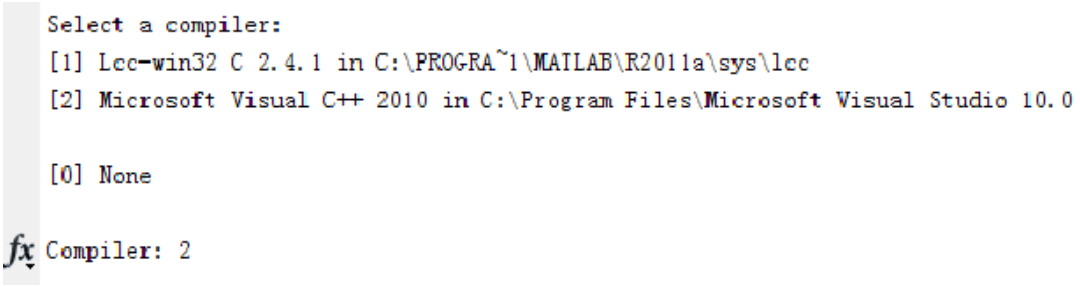
（3）找到这个文件后右键选择 Run，注意：这一步是为了选择 MatlabSimulink 编译的 Module 所需要的编译器种类，是第一次运行使用 Matlab+TE1400 的时候必须执行的，以后就不必每次都操作这一步



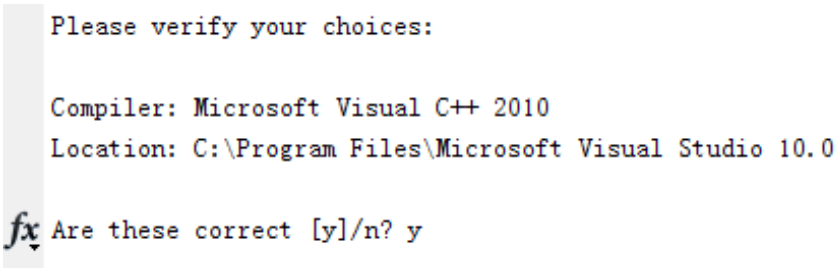
（4）运行后在 Matlab 主窗口提示让你选择是否用本地的编译器，因为本地有 VS 的编译器，Matlab 会根据系统中的信息列出本地已经存在的编译器。所以在这里选择 y 后敲回车



（5）随后 matlab 找到本地有两种编译器，一个是 matlab 本体的 lcc-win32 C2.4.1，另一个是 VS，选择 VS 所代表的数字，输入 2 敲回车



（6）最后让 matlab 让你确认编译器的选择，输入 y 敲回车



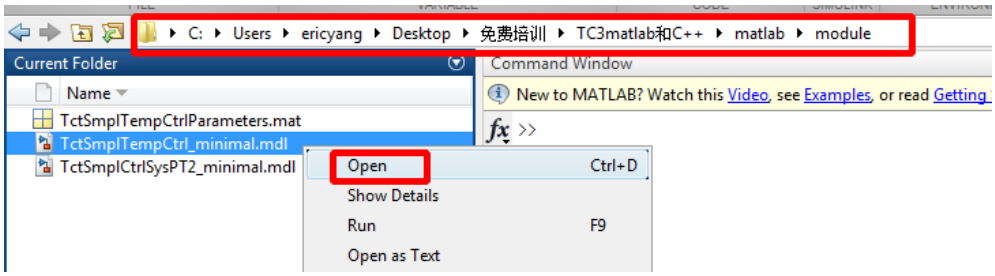
提示以下信息说明编译器选择完成

```
Trying to update options file: C:\Users\Administrator\AppData\Roaming\MathWorks\MATLAB\R2(
From template:              C:\PROGRA~1\MATLAB\R2011a\bin\win32\mexopts\msvc100opts.bat

Done . . .

********************************************************************************
  Warning: The MATLAB C and Fortran API has changed to support MATLAB
           variables with more than 2^32-1 elements.  In the near future
           you will be required to update your code to utilize the new
           API.  You can find more information about this at:
           http://www.mathworks.com/support/solutions/en/data/1-5C27B9/?solution=1-5C27B9
           Building with the -largeArrayDims option enables the new API.
********************************************************************************

IwinCAT Target setup was completed successfully
fx >>
```

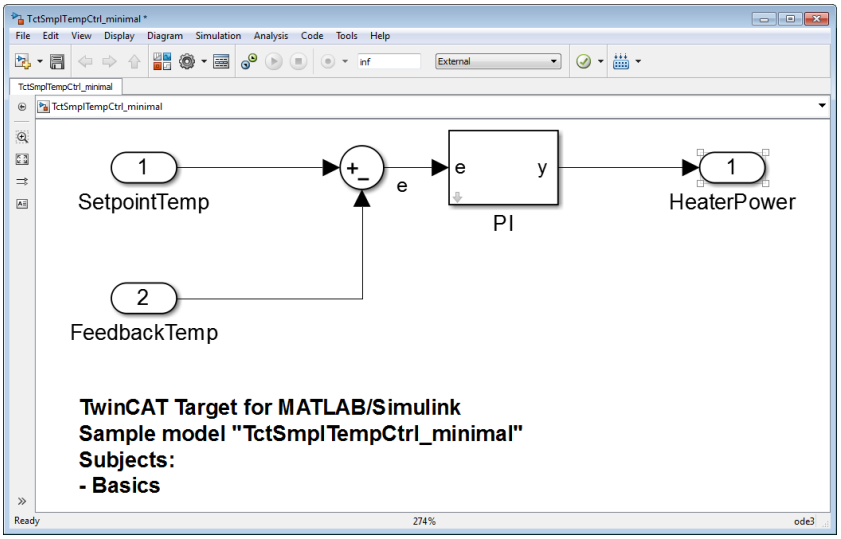
注意：如果你安装的是比较新版本的 Matlab，比如 2014b，那在选择编译器的时候都会自动匹配，只要运行.P 文件就会自动找到电脑中合适的编译器进行选择，如下图：

```
>> SetupIwinCatTarget
################################################################################
########### IwinCAT IE1400 1.2.1221.0 Setup ######################
IwinCAT IE1400 paths were added successfully.
IwinCAT IE1400 uses "Microsoft Visual C++" compilers to build IwinCAT modules from generated code.
It is recommended to use a "Microsoft Visual C++" compiler also for mex builds.
mex was configured to use the compiler "Microsoft Visual C++ 2013 Professional (C)".
fx >>
```

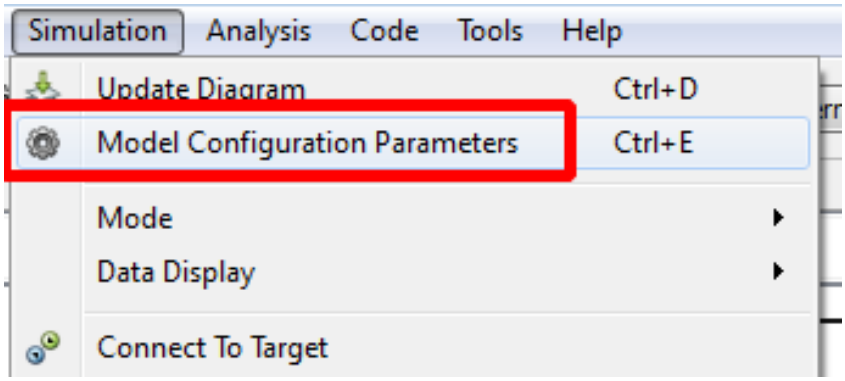
（7）在路径栏中找到模型路径，并且右键选择模型点击 open 打开模型



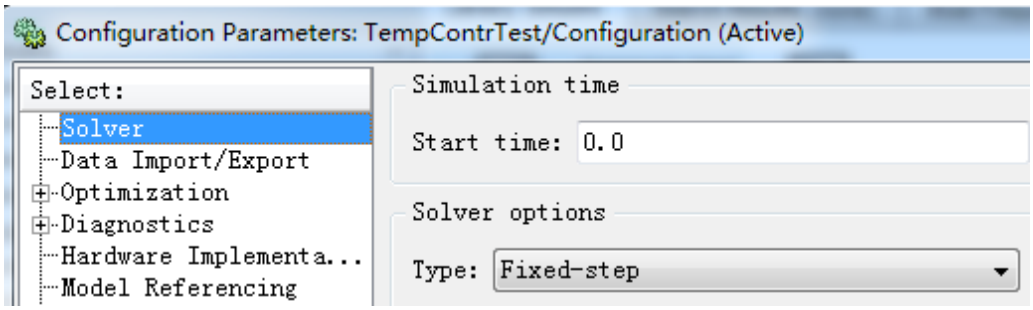
本次案例模型是一个简单的 PI 控制模型



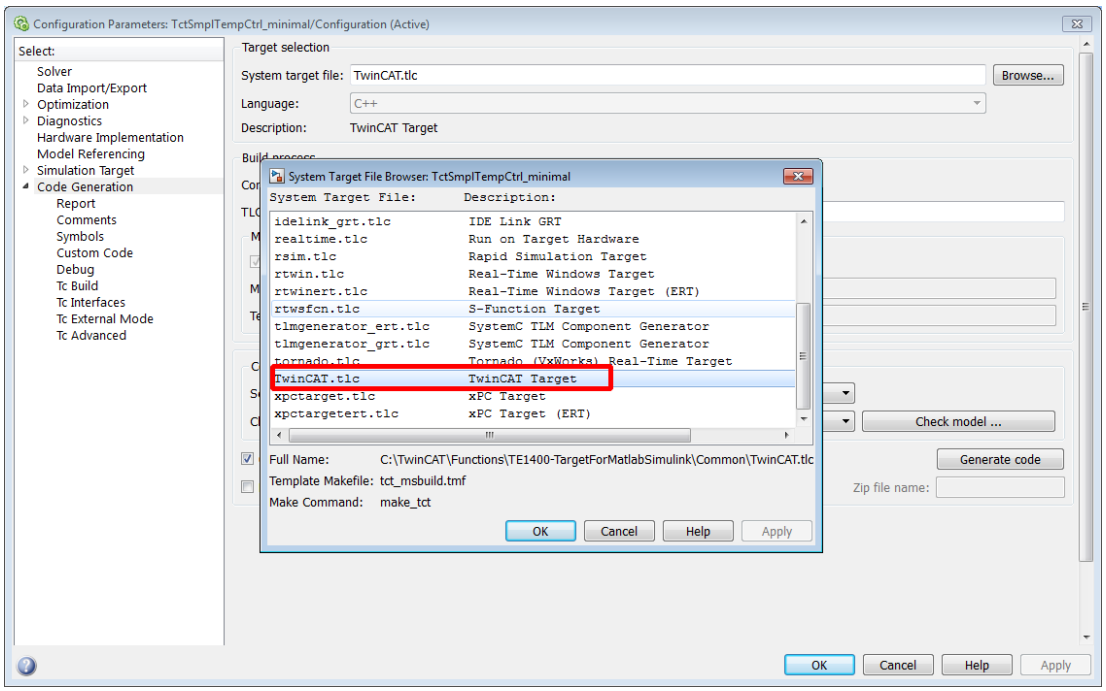
（8）打开 simulation 菜单栏，选择 configuration parameters 进行参数设定



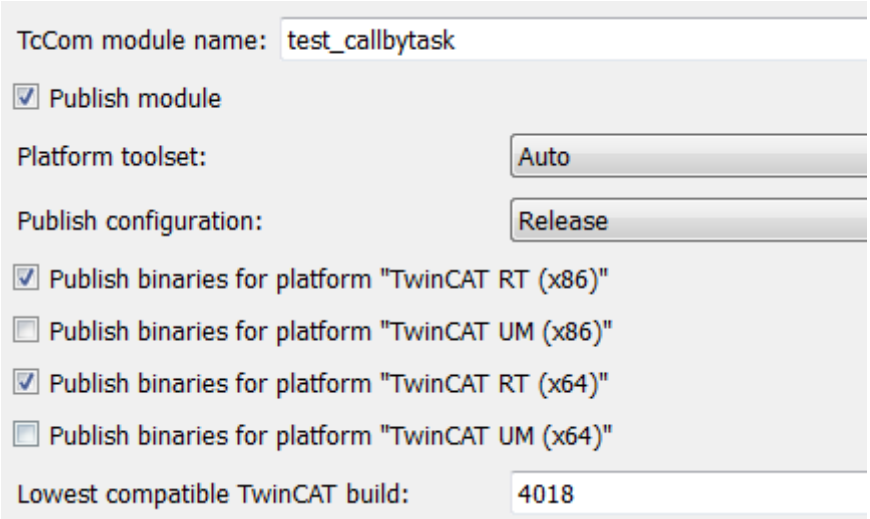
（9）进入参数设定后，选择右边的树形栏中的 Solver，把其中的 Type 改成 Fixed-step



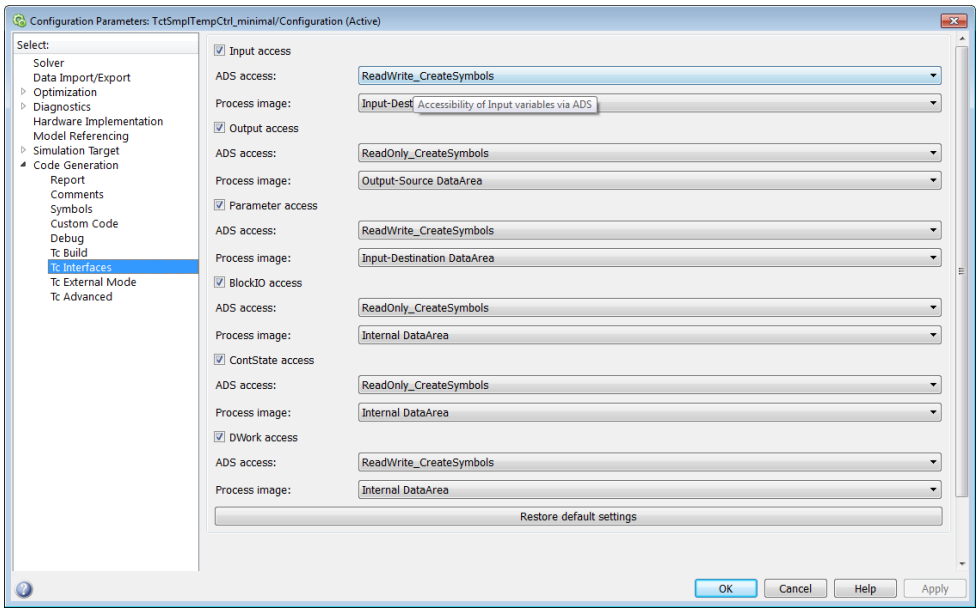
（10）之后选择树形栏中的 Code Generation，把其中的 System target file 改成 TwinCAT.tlc 点击 Browse 可以进行选择



（11）继续选择树形栏中的 Tc Build，输入模块名 test_callbytask，并且选择需要导出的模块平台种类，x86 还是 x64（因为我的电脑是 64 位所以我都勾选）



（12）TC interface 中设置模块中哪些部分支持 ADS 接口，并且也可以设置模块中不仅仅输入输出可以 mapping，也可以开放其他参数支持 mapping，比如 parameter



（13）最后选择树形栏中的 Tc Advanced，修改相应参数：

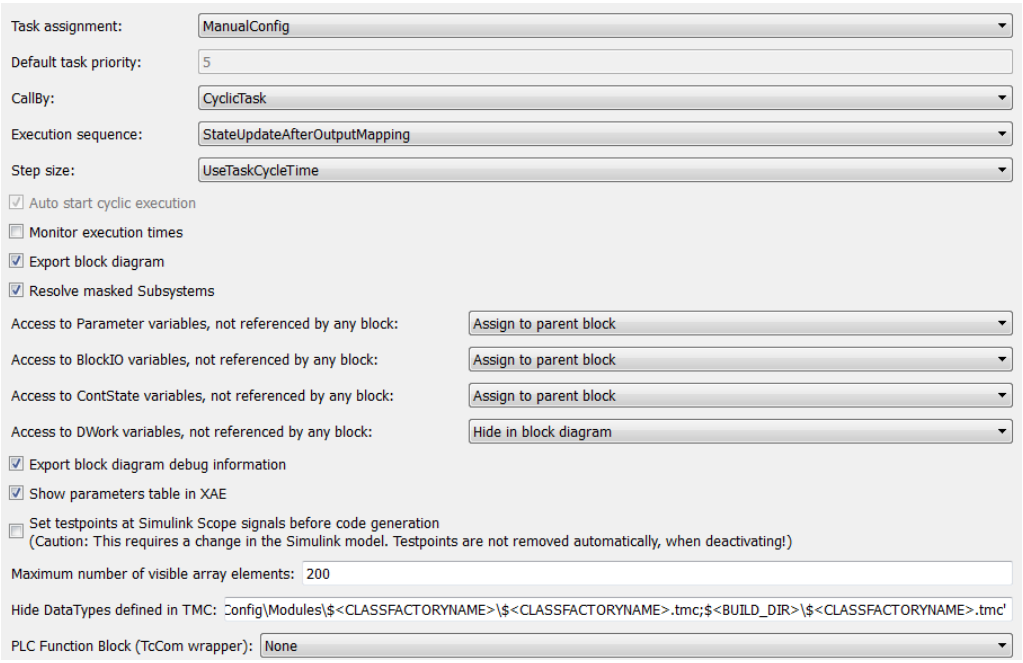
Task assignment：Manualconfig；

CallBy：CyclicTask；

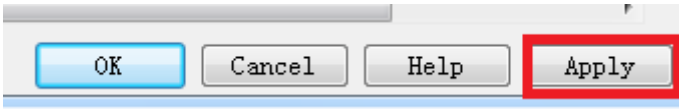
Step size：UseTaskCycleTime；

PLC Function Block：None；

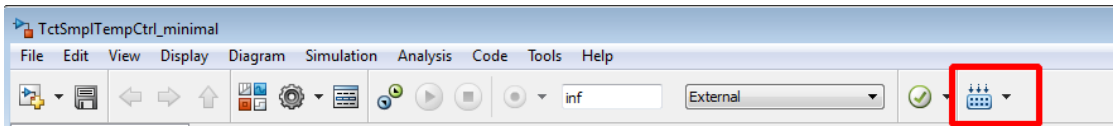
并且可以按照截图中勾选部分选项



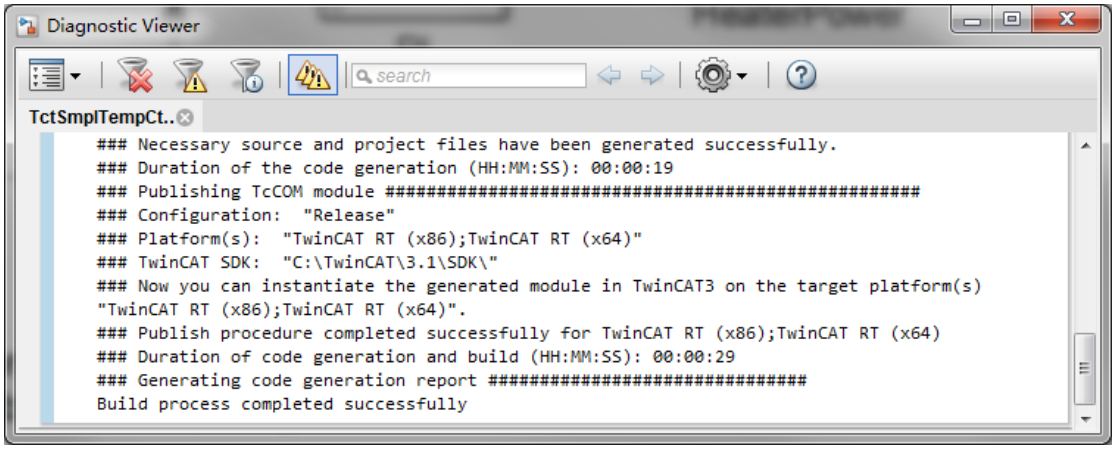
（14）以上操作完成后点击左下角的 Apply



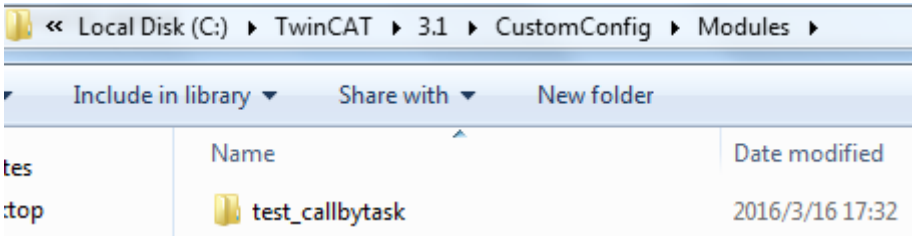
（15）随后回到模型文件，在工具栏中找到 build，编译并且转换成 TcModule



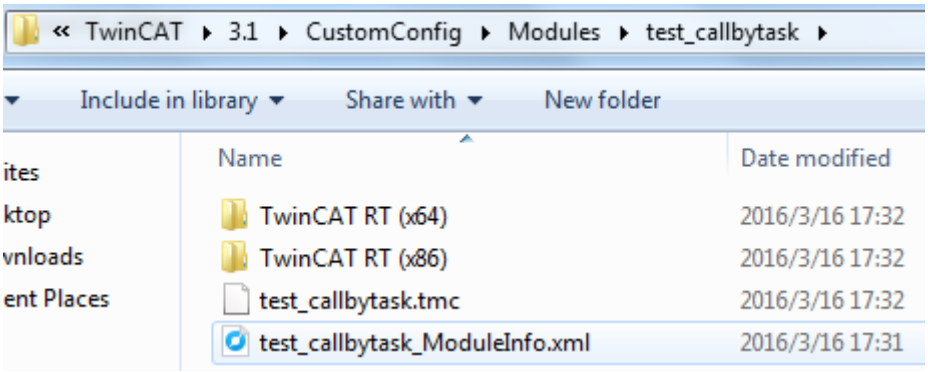
打开 Diagnostic View 中可以看到生成过程，提示 Module 生成完成

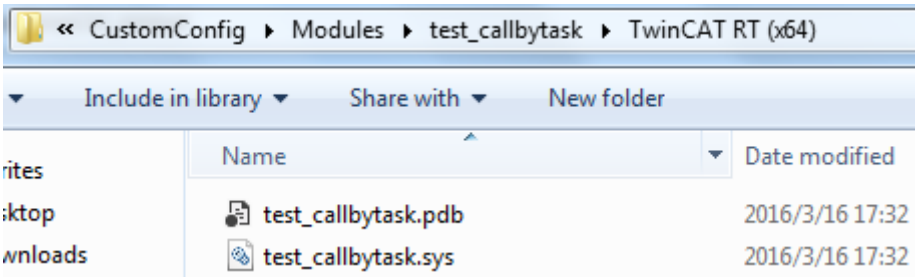


我们来看下生成的 Module 会在什么位置
可以发现在 TwinCAT/3.1/CustomConfig/Modules 路径下会生成名字和案例模型名字一样的文件夹 test_callbytask

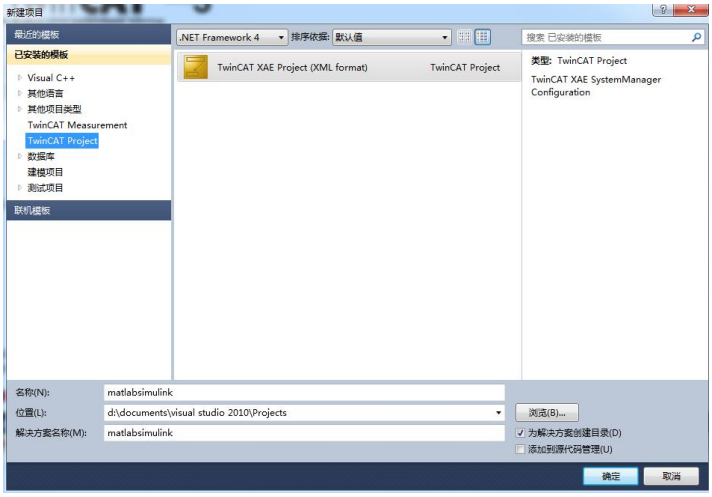


打开可以发现里面有.tmc 描述文件，x64 和 x86 的二进制文件，以及 1 个 xml 文件，这些文件都是经过编译后的离线文件，可以支持把整个文件夹拷贝出来，给那些没有 Matlab 的电脑中使用

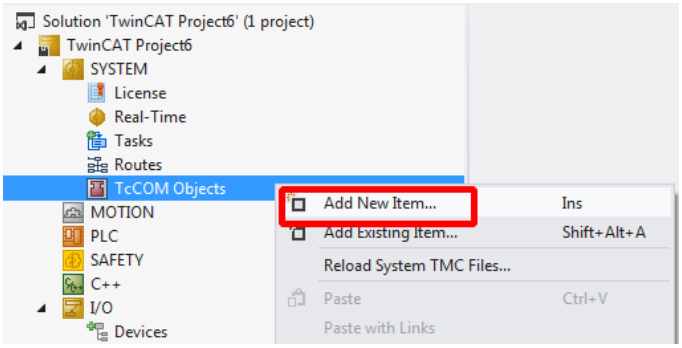




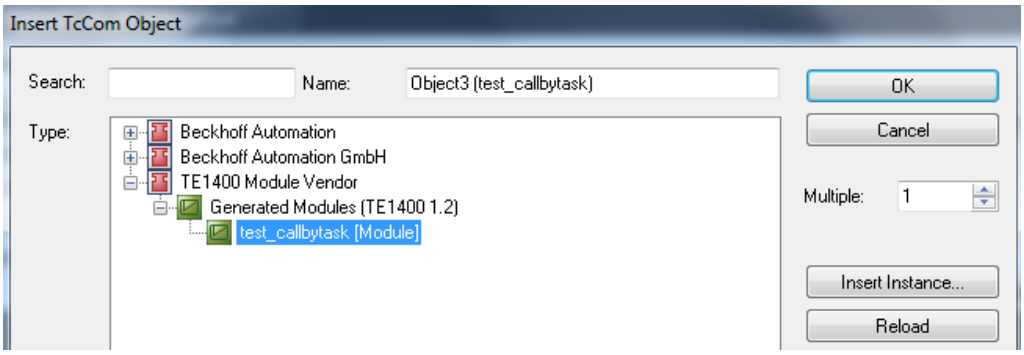
（16）创建 TC3 新项目



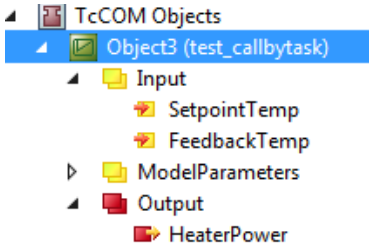
（17）打开 SYSTEM，右键 TcCOM Objects 添加新项



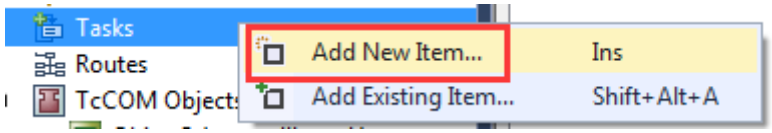
（18）TC3 会自动找到之前生成的模块，选中后点击 OK 进行添加



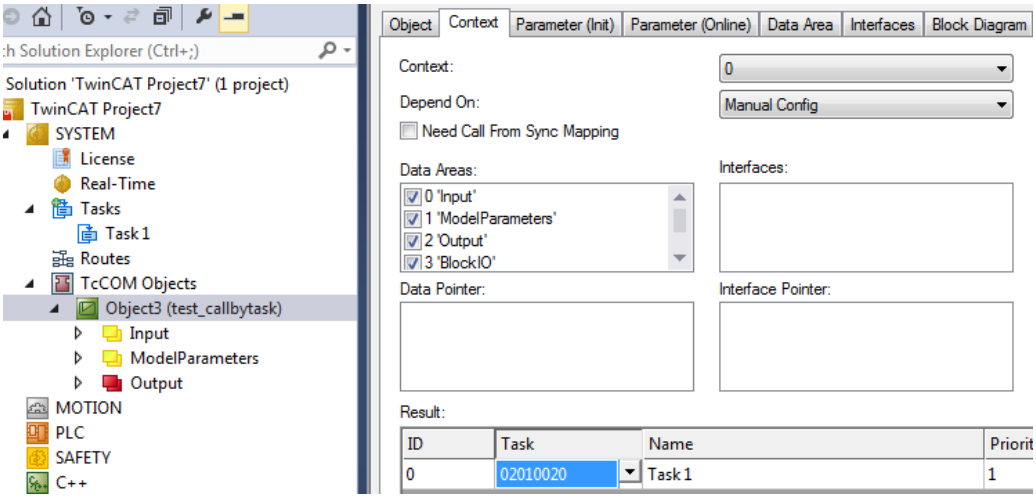
（19）添加好后我们可以发现 TcCOM Objects 下出现 matlab 生成的 Module，并且 3 个变量出现在 IO 位置，方便和 PLC 程序或者硬件 IO 进行变量连接



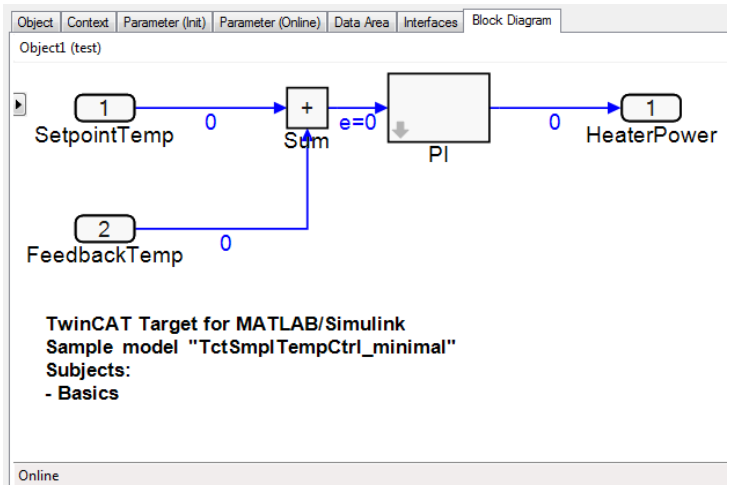
（20）右键 Tasks 添加新项



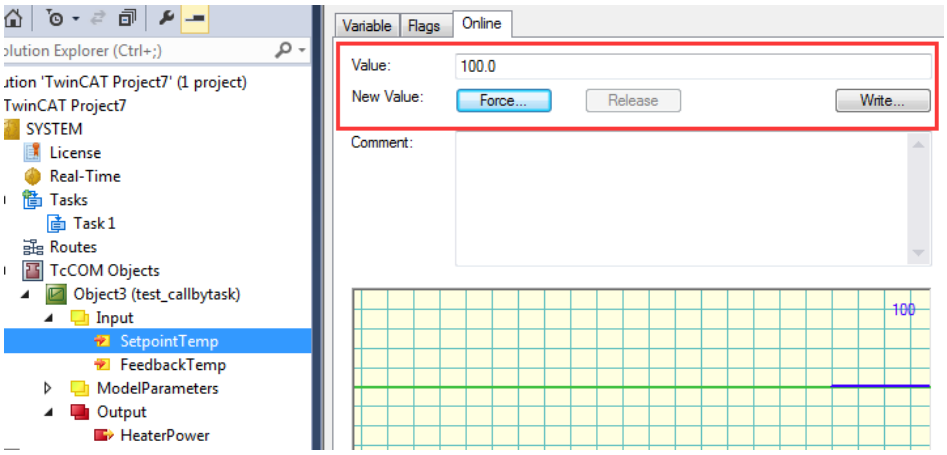
（21）分配 Task 给模块

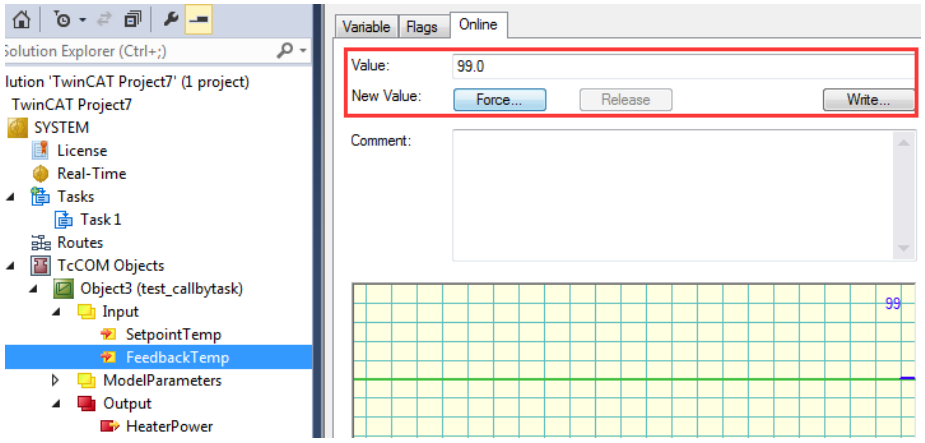


（22）active configuration 激活配置并下载程序，通过 block diagram 可以查看模块在线的状态

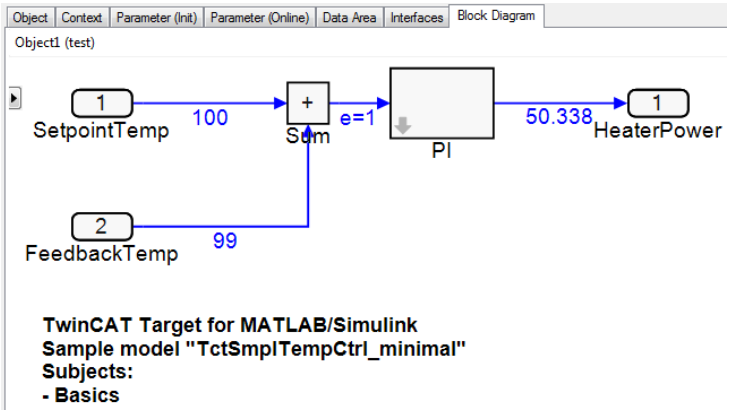


（23）随后分别给 2 个输入一个值 setpointTemp：100，feedbackTemp：99。

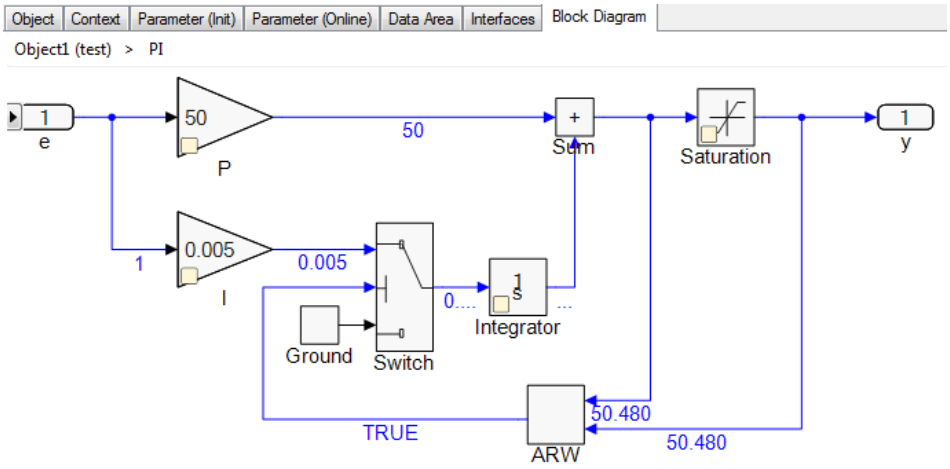




（24）此时观察模块可以发现已经有输出操作量了



（25）并且可以到 PI 这个 subsystem 中详细看到 PI 计算过程



注：虽然 TE1400 插件装上了，但用的是 Demo 版，所以对于使用有一些限制，查询 information system 可以看到如下：

Restrictions in the demo version

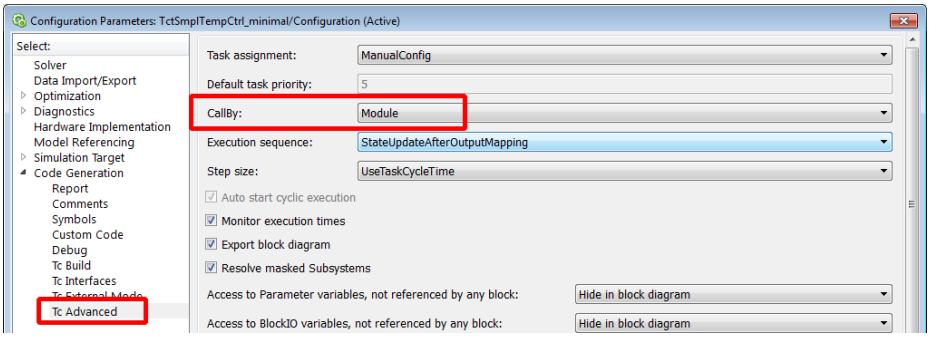
You can run the module generator without any license with the following limitations relating to the the complexity of the source Simulink model. Allowed are models with a maximum of:

- 100 blocks
- 5 input signals
- 5 output signals

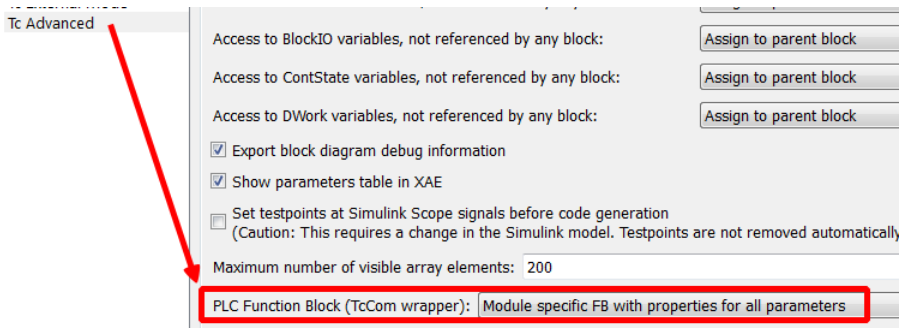
2.2 TE1400 生成模块 call by module 简介

紧接着上一篇，这次不希望模块自动周期执行，而是希望被 PLC 条件调用

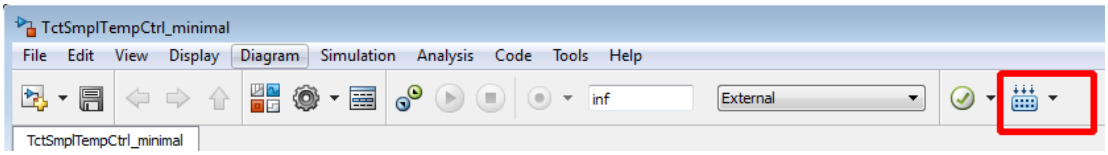
（1） 回到 Simulink 模型，在 Configuration 中找到 CallBy，设置为 Module。



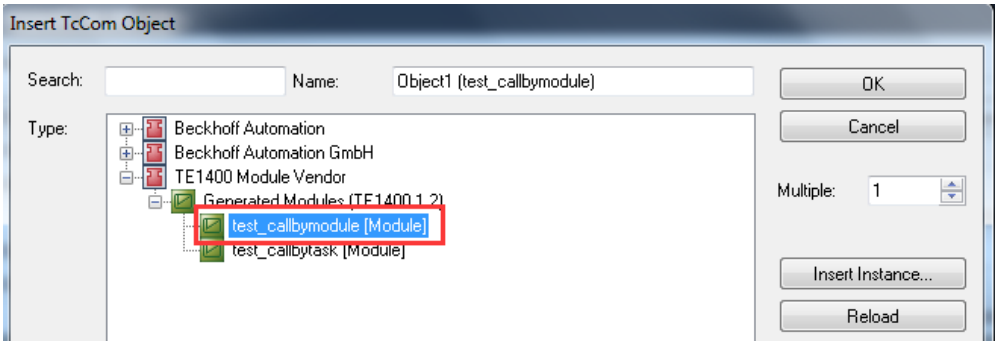
（2） 同样在 Tc Advanced 中找到 PLC Function Block，设置成 Module specific FB with properties for all parameters，这样就会在模块生成的时候自动创建 PLC 调用此模块的功能块



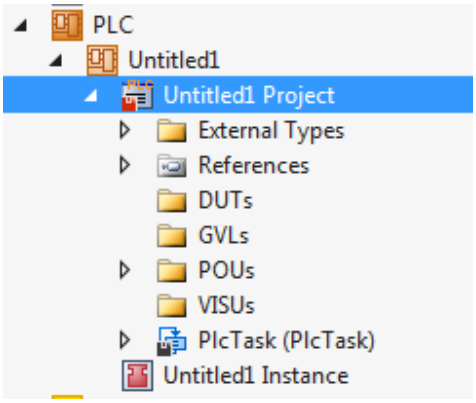
（3） 之后别忘了修改模块名字，随后就可以重新编译此模型



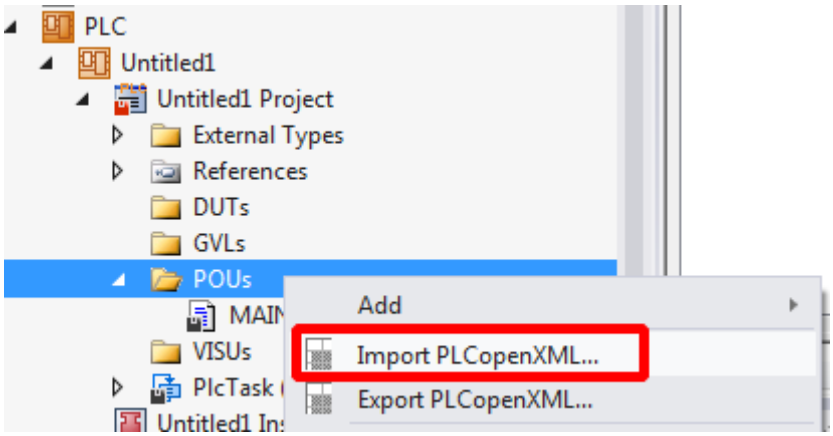
（4） 点击 reload 重新上载下模块，并且选择新的模块进行添加



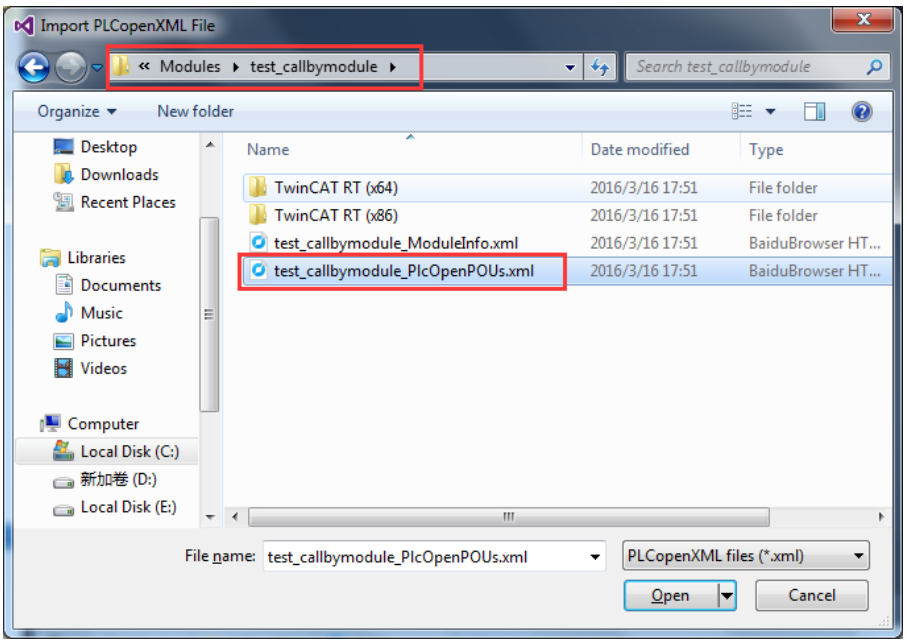
（5） 新建 PLC 项目



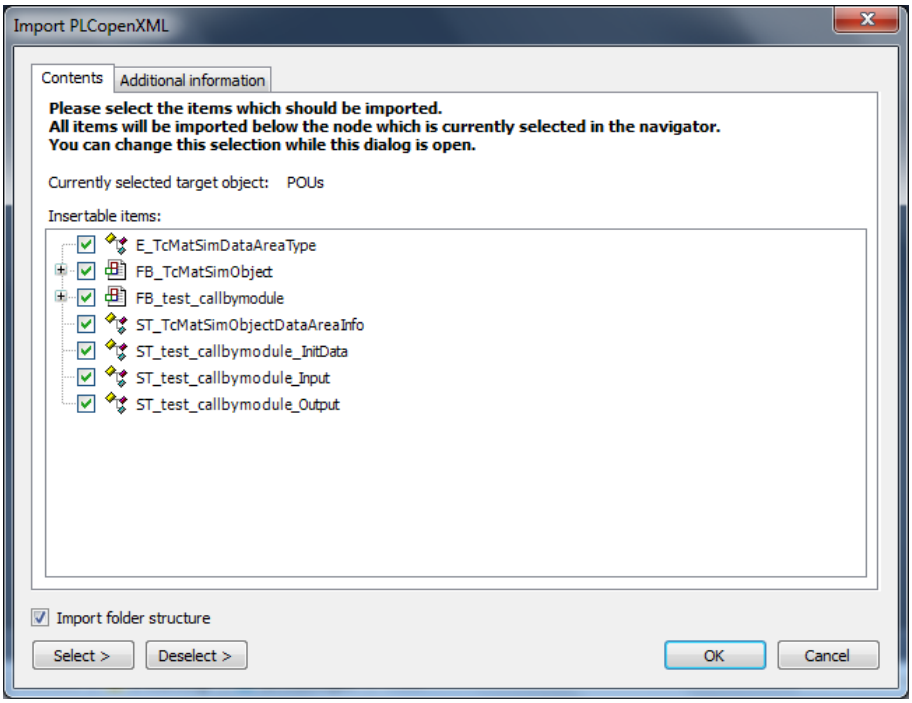
（6） 右键 POUs 点击 Import PLCopenXML。



（7） 在默认路径 C:\TwinCAT\3.1\CustomConfig\Modules\test_callbymodule 中找到 test_callbymodule_PlcOpenPOUs.xml 并且点击 open 加入到 PLC 项目中



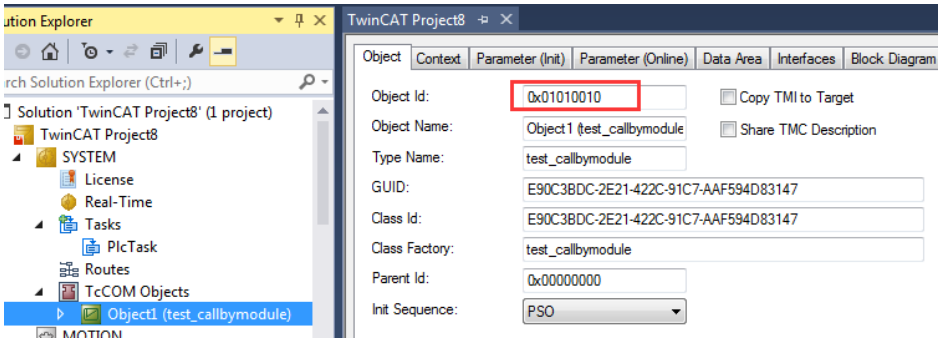
（8） 点击 OK 确认添加



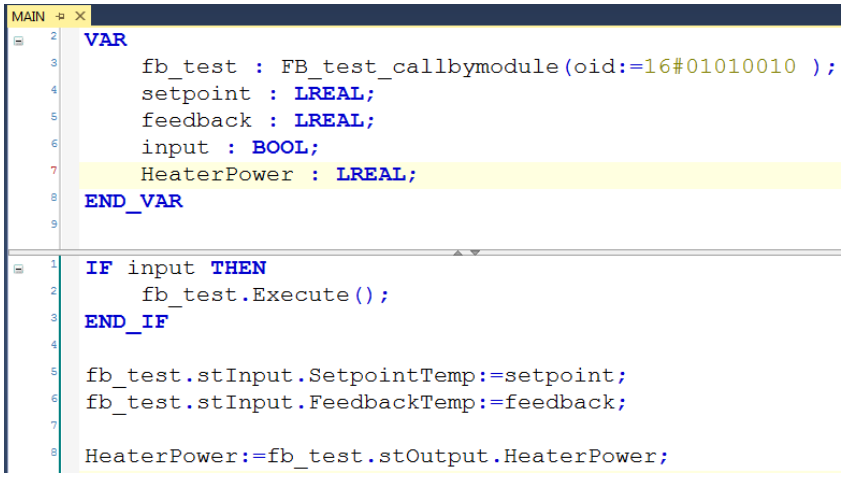
（9） 接下来开始写程序，首先对于模块接口功能块进行变量声明，在 oid 中输入 TcCOM 中模块的 Object Id

```
MAIN
1  PROGRAM MAIN
2  VAR
3      fb_test : FB_test_callbymodule(oid:=16#01010010 );
4  END_VAR
```

Object ID 可以在 TcCOM 中加载的模块中查看到



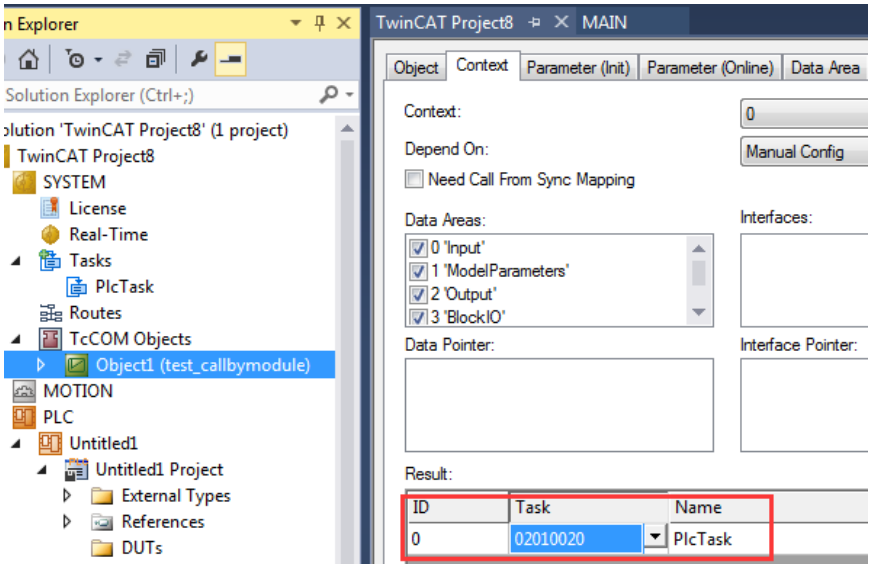
（10） 随后在程序中就可以调用相应的 method 进行模块调用，并且也可以直接访问到输入输出等参数进行赋值



```
VAR
    fb_test : FB_test_callbymodule(oid:=16#01010010 );
    setpoint : LREAL;
    feedback : LREAL;
    input : BOOL;
    HeaterPower : LREAL;
END_VAR
```

```
IF input THEN
    fb_test.Execute();
END_IF

fb_test.stInput.SetpointTemp:=setpoint;
fb_test.stInput.FeedbackTemp:=feedback;

HeaterPower:=fb_test.stOutput.HeaterPower;
```

（11） 最后别忘了分配模块 Task 为 PlcTask（必须和所调用 PLC 的 Task 一致）。



（12） 激活并下载程序后发现即使输入给了值，输出也没有变化。

MAIN [Online]

TwinCAT_Device.Untitled1.MAIN

Expression	Type	Value	Prepared value
fb_test	FB_test_callbymodule		
setpoint	LREAL	100	
feedback	LREAL	99	
input	BOOL	FALSE	
HeaterPower	LREAL	0	

（13） 只有触发了 input，使 execute 方法被调用才可以执行 simulink 模型

MAIN [Online]

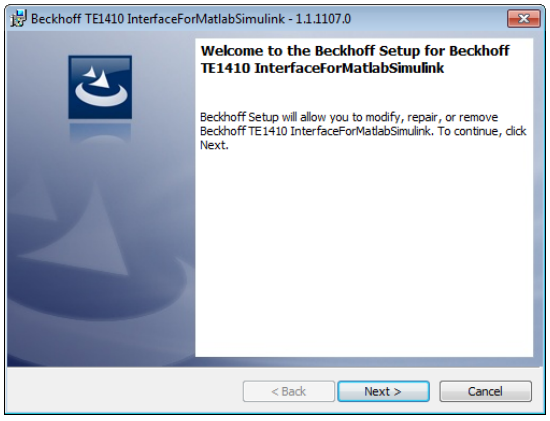
TwinCAT_Device.Untitled1.MAIN

Expression	Type	Value	Prepared value
fb_test	FB_test_callbymodule		
setpoint	LREAL	100	
feedback	LREAL	99	
input	BOOL	TRUE	
HeaterPower	LREAL	50.0067	

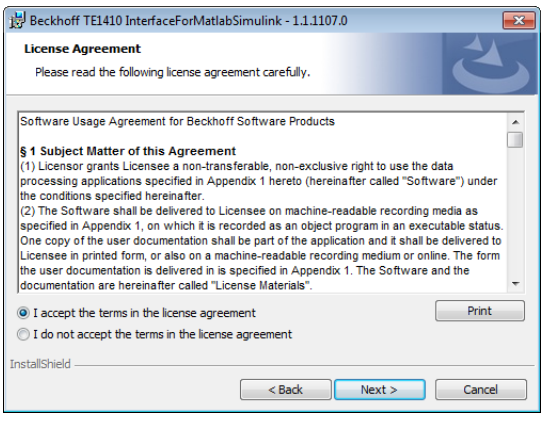
3. TE1410 使用

3.1 TE1410 安装步骤

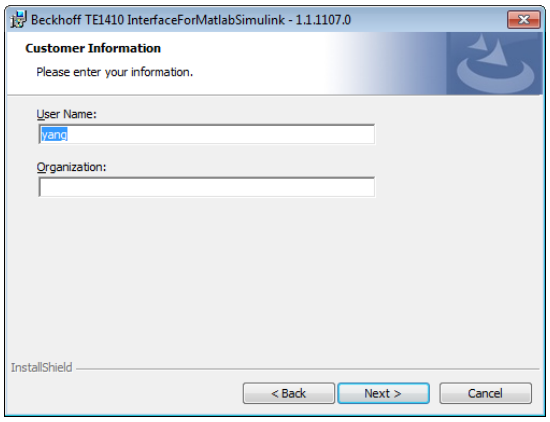
（1）首先安装 TE1410 软件



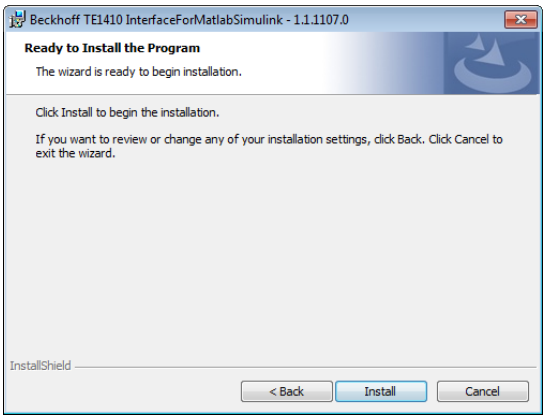
（2）点击 I accept，并且 next



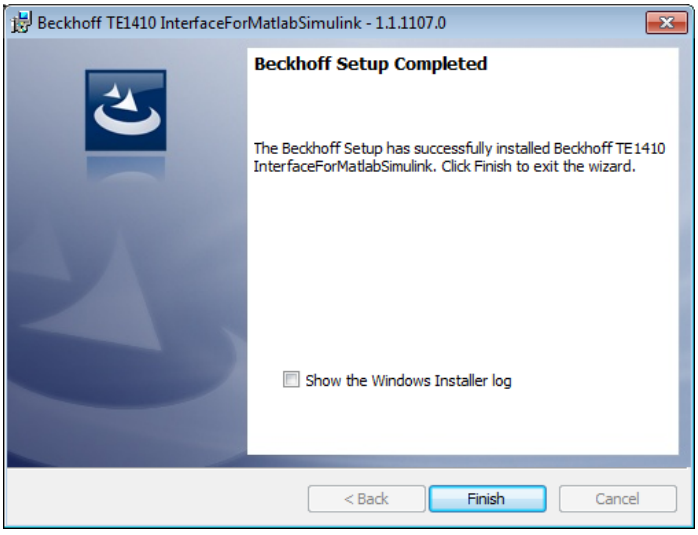
（3）下一步



（4）点击安装

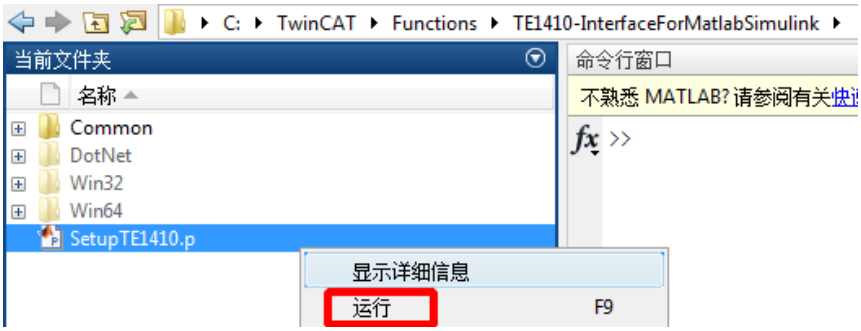


（5）安装完成后点击 finish

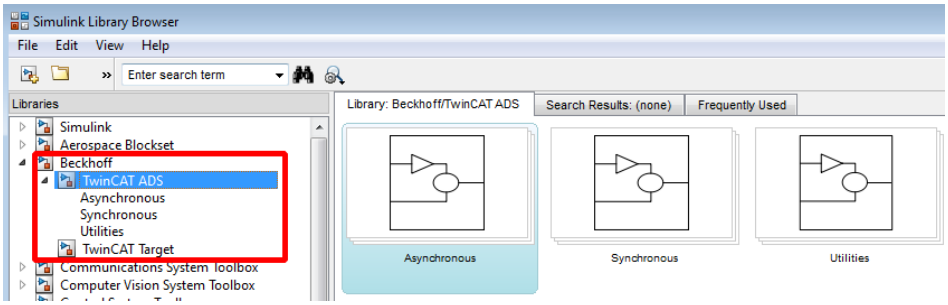


3.2 TE1410 使用简介

（1）打开 Matlab，找到 TE1410 安装路径，执行里面的 SetupTE1410.p 文件



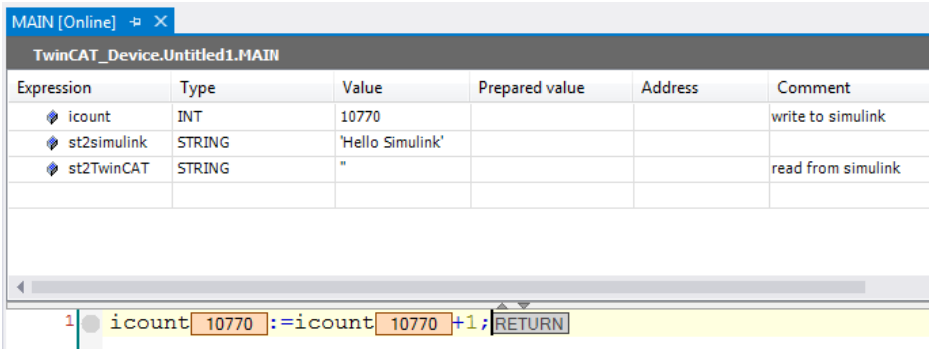
（2）随后重新打开 Matlab，在 Simulink library 中就可以发现集成了 Beckhoff 的 ADS 模块，利用这些 ADS 接口的模块就可以和安装有 TC3 的控制器进行 ADS 通信。



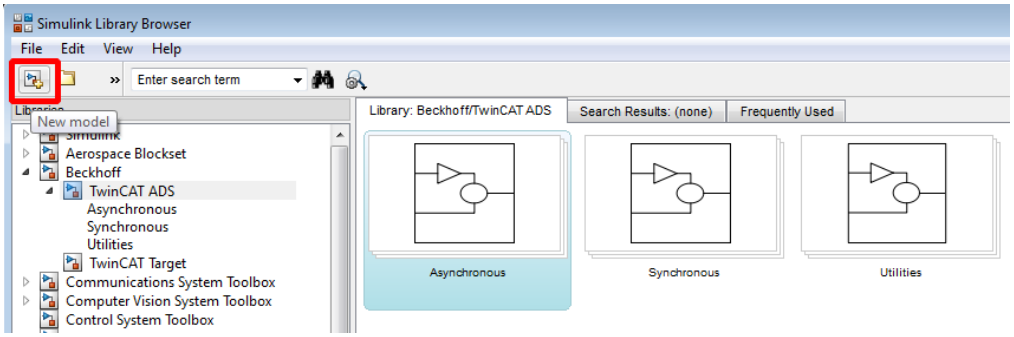
（3）首先打开 TC3，新建 PLC 项目，并且添加以下变量：
icount 和 st2simulink 是发送给 simulink；st2TwinCAT 是从 simulink 读取

```
MAIN
PROGRAM MAIN
VAR
    //write to simulink
    icount: INT;
    st2simulink: STRING:='Hello Simulink';
    //read from simulink
    st2TwinCAT: STRING;
END_VAR

icount:=icount+1;
```

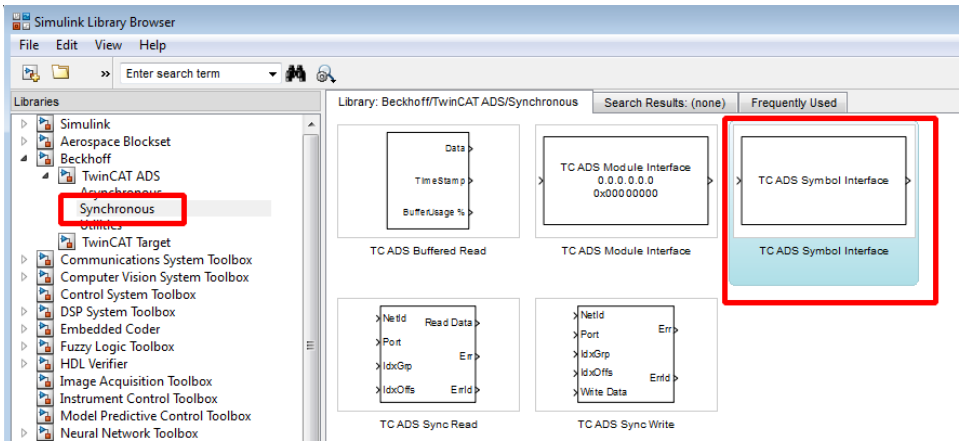
（4）运行程序并且 login

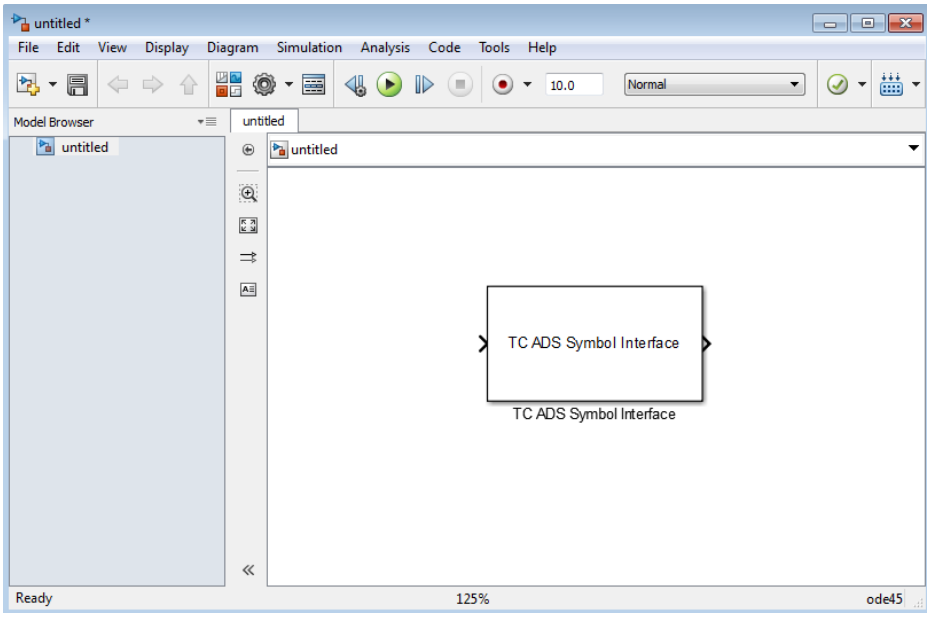


（5）打开 simulink library browser，点击工具栏的 new model 新建 simulink 模型

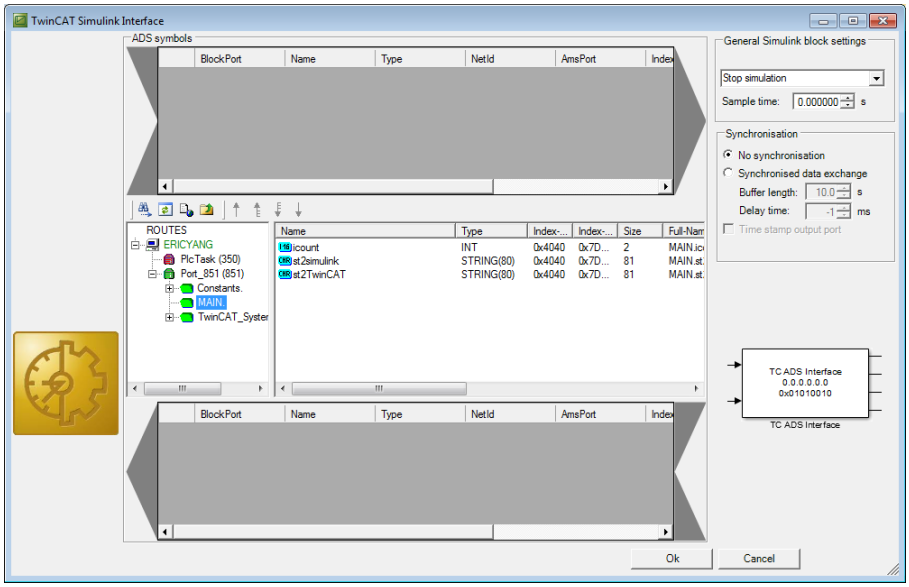


（6）在 beckhoff→TwinCAT ADS→Synchronous 库中找到 TC ADS Symbol Interface 添加到新的 simulink 模型中

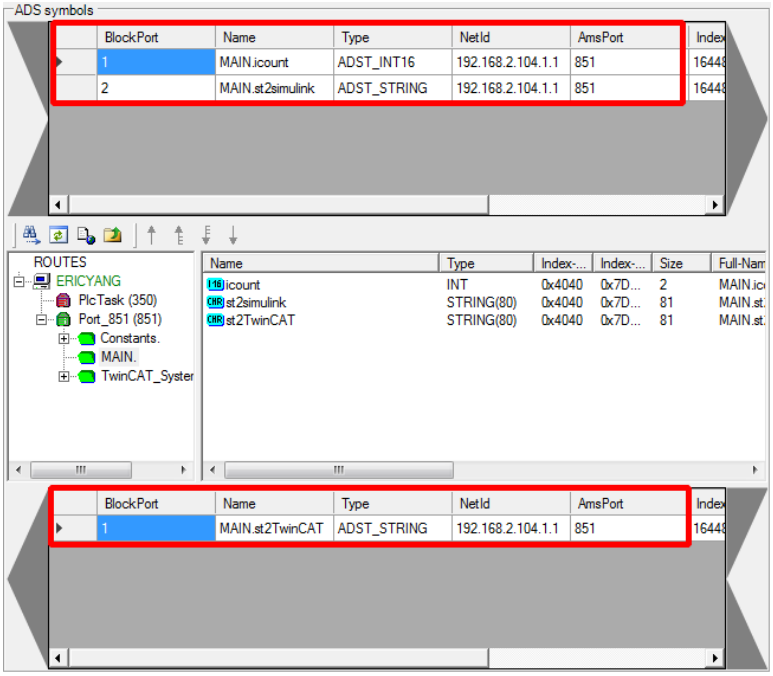




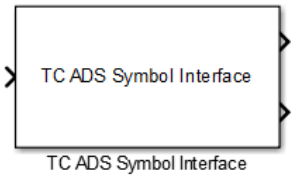
（7）双击 TC ADS Symbol Interface，在 ROUTES 中找到相应的控制器或者本地电脑，并且在 PLC 端口（默认起始 851）中找到之前添加的 3 个变量



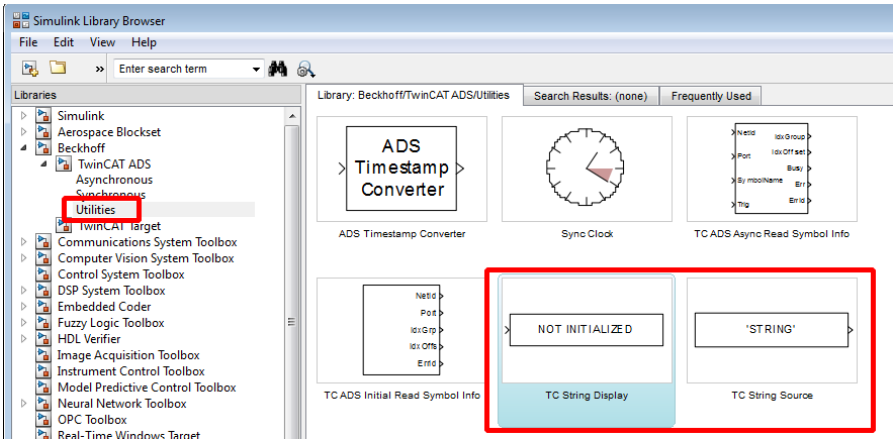
（8）随后把 icount 和 st2simulink 添加到上栏代表写给 simulink 模型，st2TwinCAT 添加到下栏代表从 simulink 读取。

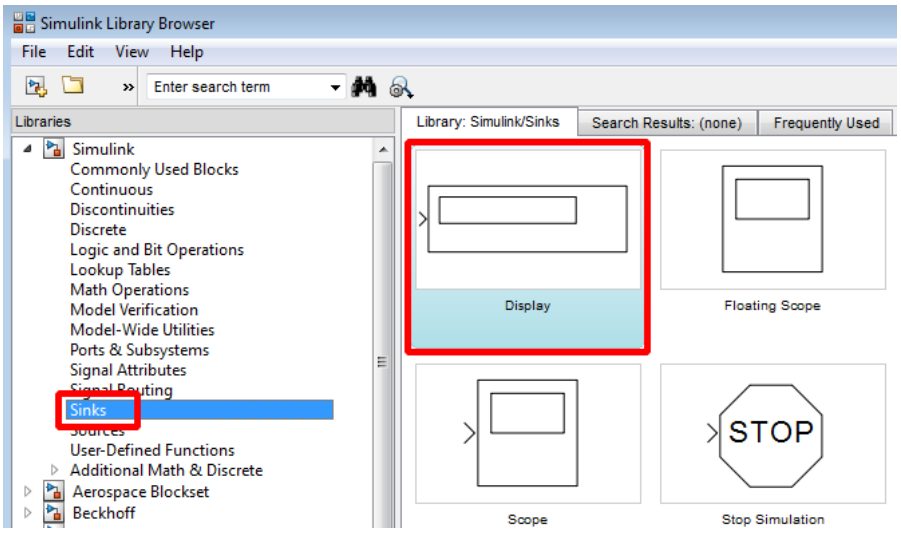


（9）设置好点击 OK 后发现 TC ADS Symbol Interface 有 1 个输入和 2 个输出

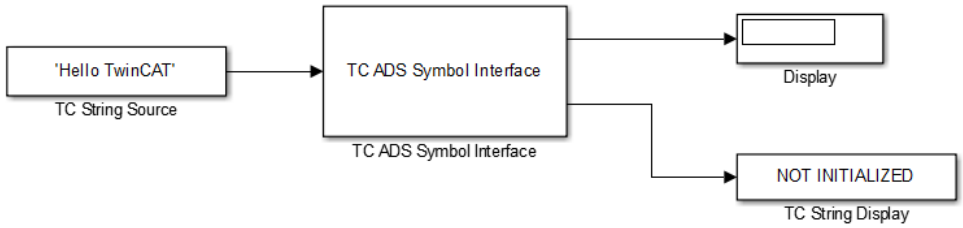


（10）随后继续在 simulink library toolbox 中找到 3 个模型，分别是 Beckhoff→TwinCAT ADS→Utilities 中的 TC String Display 和 TC String Source；以及 simulink→sinks 中的 Display

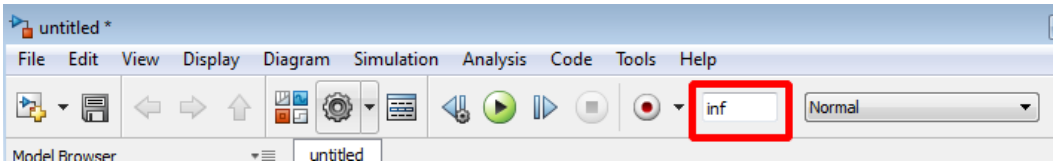




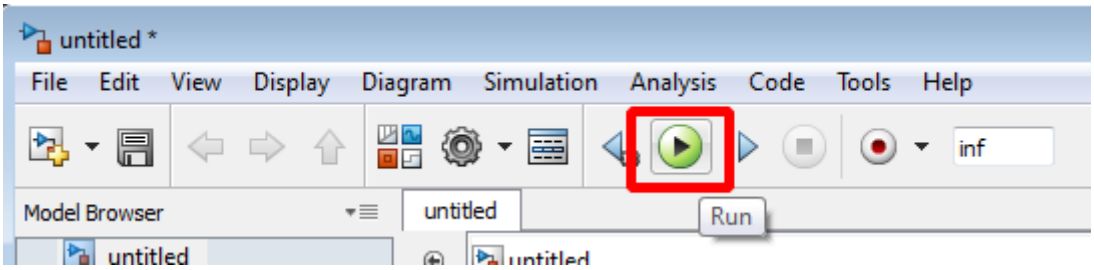
（11）把这 3 个变量分别加入模型中，并且双击 TC String Source 修改字符串为’Hello TwinCAT’



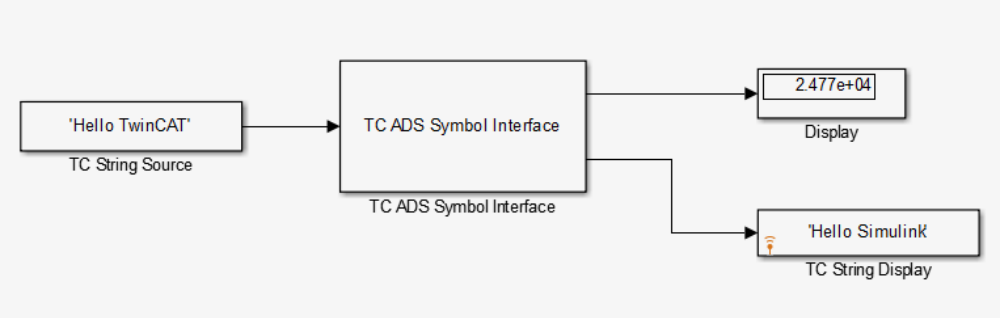
（12）随后修改 simulink 仿真停止时间为 inf(无穷大)



（13）最后点击工具栏中的 run



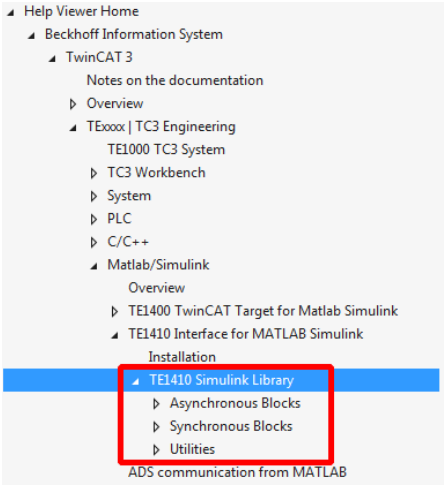
（14）可以发现和 TwinCAT3 中的变量读写成功



（15）当然也可以在 TC3 中观察到 st2TwinCAT 被写成功

Expression	Type	Value	Prepared value	A
icount	INT	26950		
st2simulink	STRING	'Hello Simulink'		
st2TwinCAT	STRING	'Hello TwinCAT'		

（16）当然在 simulink library→beckhoff 中还有很多功能块，基本都是实现 ADS 通信的功能，具体可以参考帮助文档 TE1410 中有详细说明



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　　传真：021-66315696　　邮编：200072

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　　传真：010-82200039　　邮编：100035

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　传真：020-38010303　　邮编：510623

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　　传真：028-86202582　　邮编：610016

请用微信扫描二维码
通过公众号与技术支持交流

倍福中文官网：
http://www.beckhoff.com.cn/
倍福虚拟学院：
http://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