BECKHOFF

TwinCAT 3 入门教程

Version 4.17

毕孚自动化设备贸易（上海）有限公司

2021 年 10 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：

第一章介绍 TwinCAT3 软件架构，如何选择合适的 Visual Studio，以及如何安装帮助系统。

第二章介绍了 TwinCAT3 试用版授权以及完整版授权激活方式，同时介绍了两种全新硬件授权设备的介绍和使用。

第三章介绍了 TwinCAT3 中如何扫描硬件，以及虚拟层和物理层直接的连接如何实现。

第四章围绕 IEC61131-3 的概念展开了说明，讲述了 IEC61131-3 标准的核心概念，语法以及 IEC61131-3 新标准扩充的部分。

第五章介绍如何创建一个 TwinCAT3 项目，并且选择 ST（结构文本）语言进行简单编程，调用功能块，在线检测与调试的过程。

第六章介绍如何选择 PLC 中自带的 HMI 功能编辑一个完整的界面，并且实现全屏显示，用户管理，网页浏览等功能。

第七章全面介绍 TwinCAT3 中 Measurement 功能的使用，包括如何创建一个有效的示波器，如何进行 YT 和 XY 的曲线、Bar 柱状图查看，以及 reporting，cursors，trigger 和 saving data 等功能。

第八、九章讲述了 TwinCAT3 库管理和源代码管理，可以学习在 TwinCAT3 平台如何实现库的创建和管理，以及源代码上载和下载，自启动项创建的过程，针对多版本切换也有详细介绍。

第十章主要针对于原本一些 TwinCAT2 的客户，希望实现代码移植转换到 TwinCAT3 中，那可以通过本章节学习到转换的步骤，以及一些注意点。

第十一章介绍了高速实时以太网——EtherCAT，对 EtherCAT 基础性能和网络错误信息诊断进行了介绍，同时也讲解了如何在工控机上手动安装 EtherCAT 驱动的步骤。

V4.0→V4.17 更新内容：

第一章

- TwinCAT3 所支持 Visual Studio 版本更新说明

第七章

- 全新版本 Scope 使用修改

第三章

- 3.1.4024.0 或更高版本连接目标控制器步骤

错别字纠正

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虚拟学院获取到，当然本书所有配套的案例程序也会在此虚拟学院中供所有读者免费获取。

虚拟学院地址：http://tr.beckhoff.com.cn/

欢迎对本书的结构、内容提出意见和建议，请发邮件至：
support@beckhoff.com.cn

最后感谢吴静雯对第六章的编写，江星睿对第七章的编写。

技术支持部
2018 年 2 月 14 日

目录

一、 TwinCAT3 安装

1. TwinCAT3 软件概述及安装

TwinCAT3 Full 版本分 XAR 和 XAE 两部分。

XAE：eXtended Automation Engineering。

XAR：eXtended Automation Runtime。

XAE 是基于 Visual Studio 作为开发环境，进行多种语言的编程和硬件组态。

XAR 是实时运行环境，对 TwinCAT 模块加载、执行、管理、实时运行与调用。

此例程是基于 WIN7 系统进行 TwinCAT3 的安装。4020 版本的 TwinCAT3 自带 VS2013 Shell。**TwinCAT3 4020 版本可以支持 VS2015 并且支持 WIN10 系统。**

TwinCAT3 4022.14 版本可以支持 VS2017。

这里我们就介绍如何安装 TwinCAT3 4020 Full 版本和 infosys。

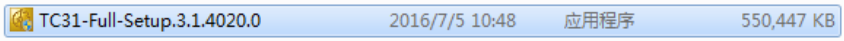
1.1 TwinCAT3 Full 版安装步骤（本例以 WES7 32 位系统为例）。

（1）首先安装 TwinCAT3 Full 版本，安装包可以从 beckhoff 官方网站进行下载。登录以下链接：

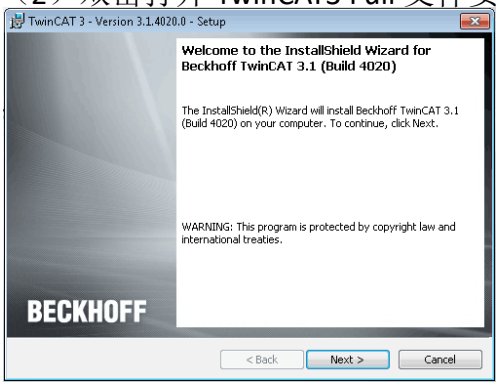
http://www.beckhoff.com/

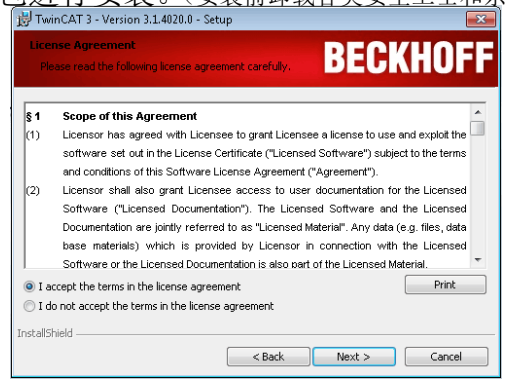
登录之后在 Download/Software/TwinCAT3/TE1xxx|Engineering 目录下找到 TwinCAT 3.1 – eXtended Automation Engineering(XAE)文件夹中的安装包：

TC3.1-Full-Setup.3.1.4020.0

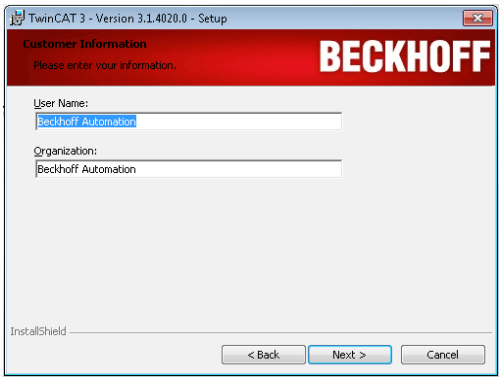


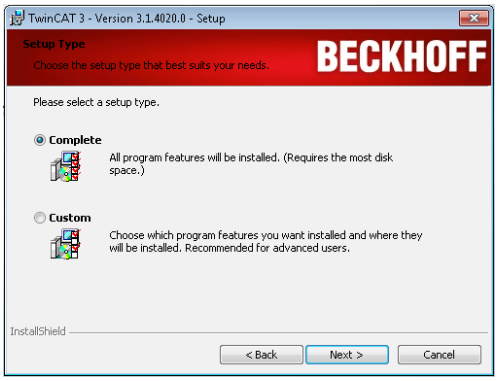
（2）双击打开 TwinCAT3 Full 文件安装包进行安装。（安装前卸载各类安全卫士和杀毒软件）



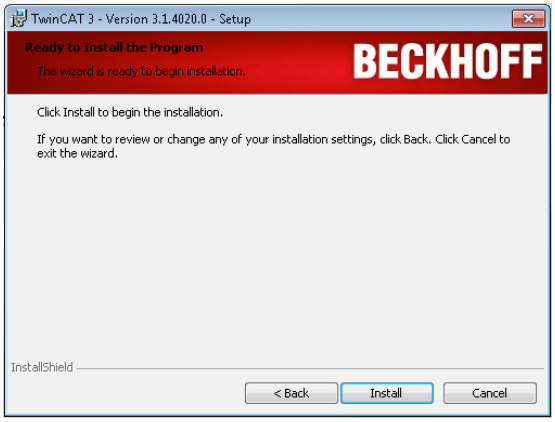


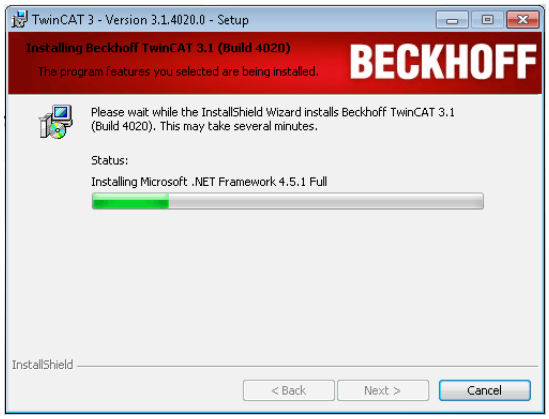
（3）这里可以选 complete 即可。

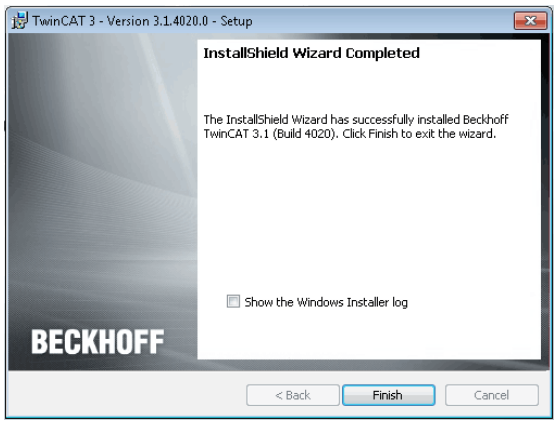


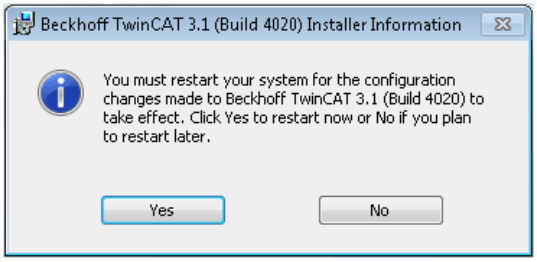


（4）点击 Install 进行安装，安装好后选择 Yes 重启。

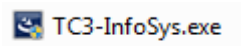


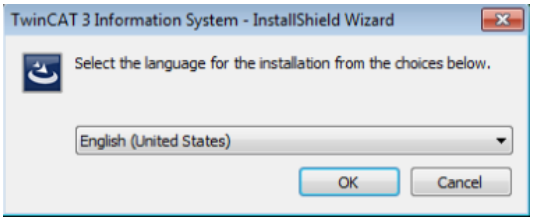


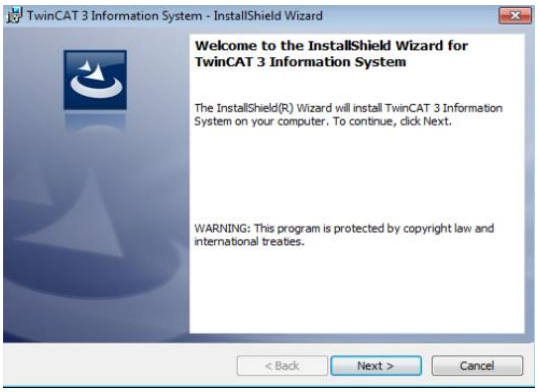


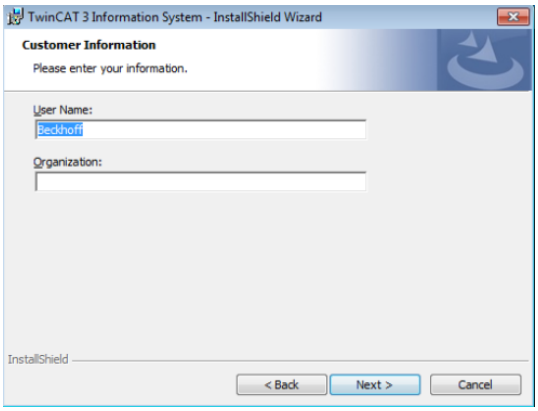


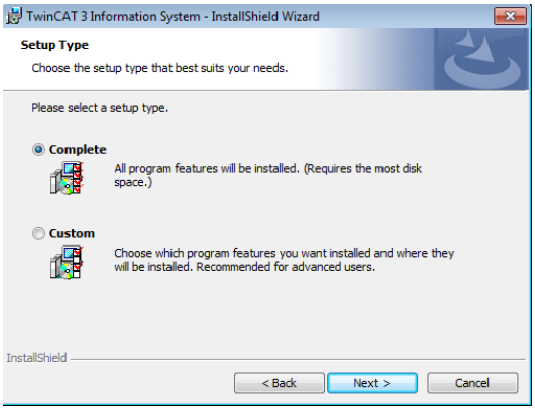
（5）最后安装帮助文档，点击：





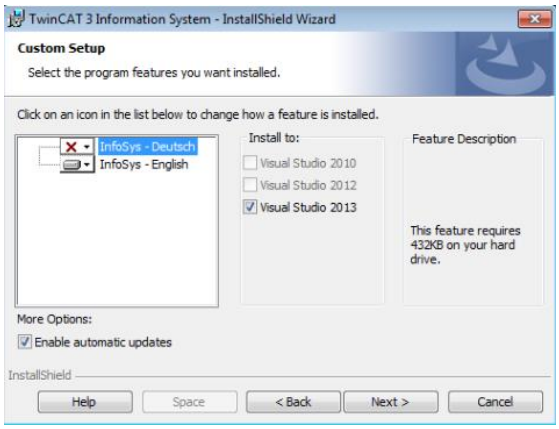


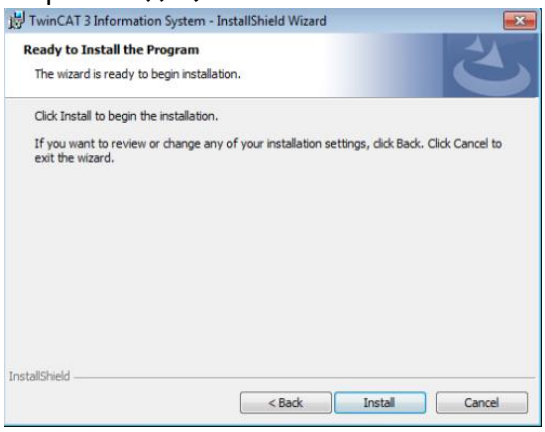




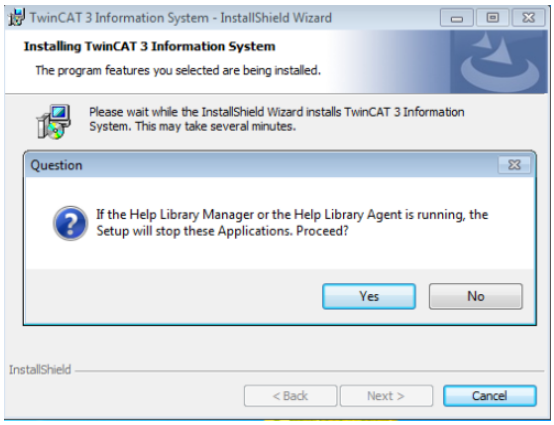
（6）到了这一步我们可以直接选 complete 并且下一步

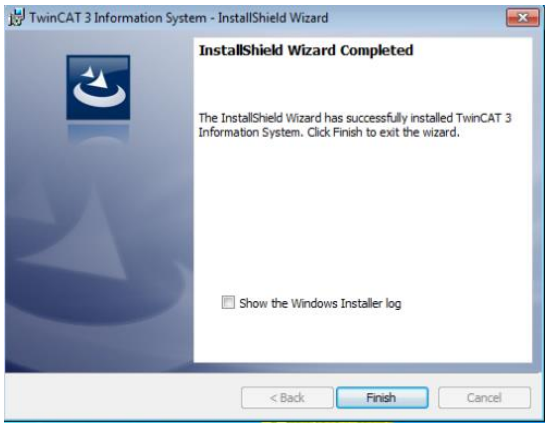
（7）也可以选择 custom，这样你就可以只选择你所需要的帮助文档的语言，并且把 Visual Studio 2013 和 Enable automatic updates 打勾



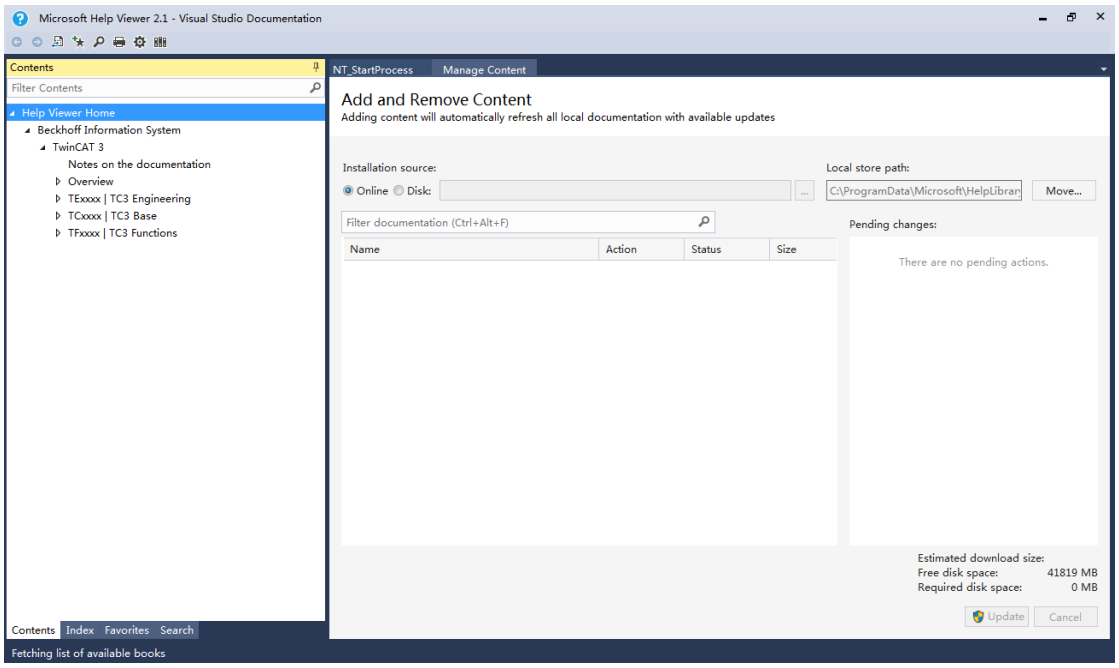


（8）选择 Yes

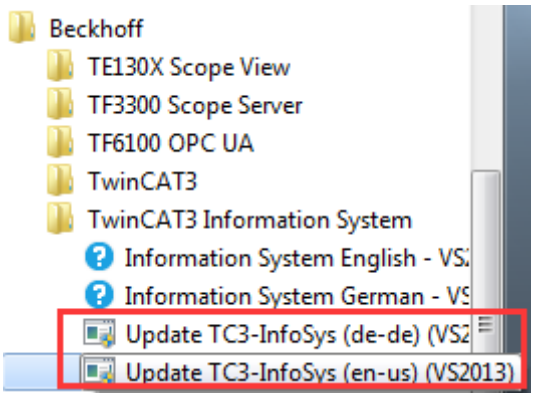




（9）这样就完成了 TwinCAT3 的安装，并且可以离线查看帮助文档(帮助文档如果链接网络可以在线更新)，完成安装后打开 TC3，点击 HELP→View Help，就可以看到 Beckhoff Information System。



如果希望手动更新帮助文档，可以在开始菜单中找到 Update TC3-InfoSys（de-us）点击进行更新。



注意：集成在 Visual Studio shell 中的 TwinCAT3 不能进行 C++和 matlab/Simulink 的开发工作，但可以执行其他 PC 开发好的 C++和 matlab/Simulink 的模块。

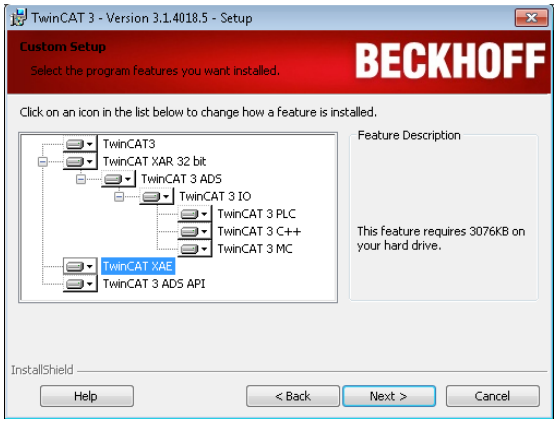
1.2 TwinCAT3 XAR 版安装。

通常在嵌入式 PC 中只需要安装 XAR 版本的 TwinCAT3 即可，因为嵌入式 PC 中硬盘通常比较小，而且只需要负责模块的执行，不需要在本机中对模块进行开发。Beckhoff 官方网站下载 Twincat3 XAR 版，下载完成后，打开软件安装包。随后的步骤和 full 版本安装(1)-(4)一样。

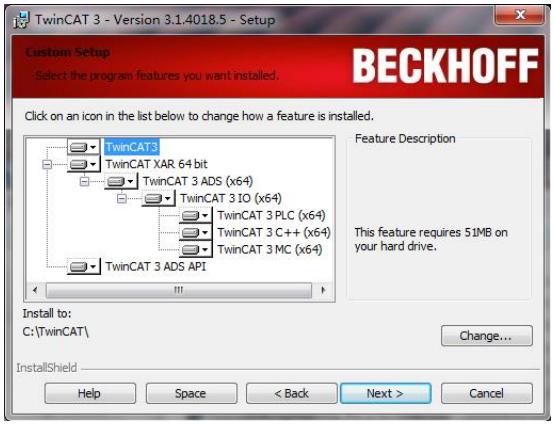
1.2.1 安装提示。

首先让我们来看下这 2 个安装包分别有什么内容：

FULL：



XAR：



（1）通过用户自定义安装可以发现：其实你只需要 Full 安装包即可，Full 安装包也可以只安装 XAR。

（2）上图中 32bit 是软件自动识别系统是 32 位系统还是 64 位，因此 3.1 版以后都可以支持 32 位和 64 位。

（3）安装路径建议安装在全英文路径下，如果不是用户自定义安装默认在 C:盘下。

1.3 安装中出现的弹窗

（1）出现警告，提示需要以管理员身份运行。

答：右键安装包并以管理员身份运行。

（2）如果以管理员身份运行了还是跳出如上弹窗。

答：当前用户没有管理员权限，可以在控制面板中查看当前账户是否具备管理员权限。如果不具备管理员权限，请更换有管理员权限的账户进行安装，或者修改当前账户使之具备管理员权限。

（3）出现报错，提示“Could not write value to key”。

答：某些杀毒软件阻止了服务导致安装失败，安装前卸载杀毒软件，或将 TwinCAT 添加至杀毒软件白名单。

（4）出现警告，提示“Could not install .net framework 4.5”导致安装无法继续的情况。

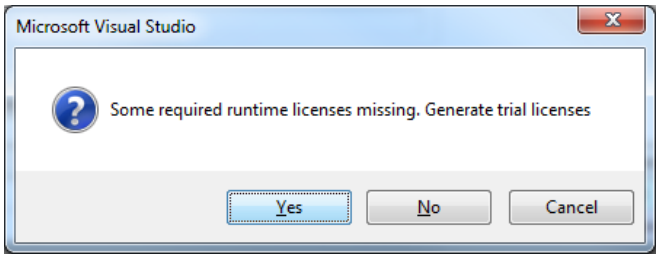
答：XP 系统无法安装.net framework 4.5，对于 Visual Studio 来说也就无法安装 2012 等以上版本。只能安装 VS2010，再装 3.1.4018.X 版本的 TwinCAT3。

二、 TwinCAT3 的授权激活

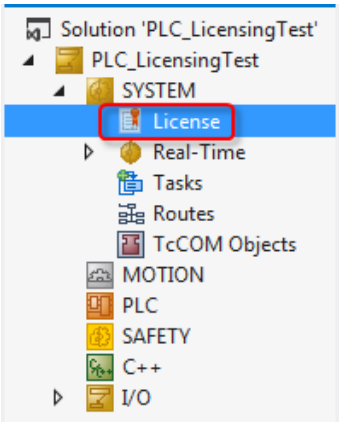
1. 试用版授权激活

在 TwinCAT3 中可以很方便激活试用版 7 天授权。在不连接网络的情况下，可以根据实际需求重复激活使用。

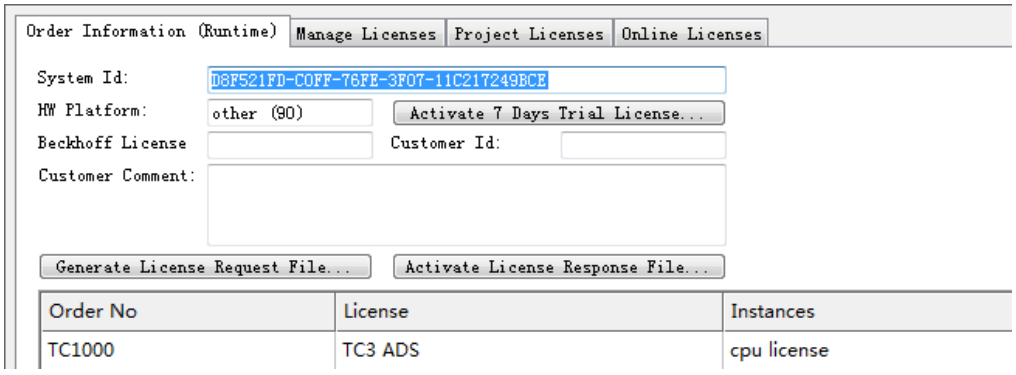
为此 TwinCAT3 开发环境可以自动检测项目中所需要的授权，并且在运行项目的时候进行提示，比如当项目在本地或者目标控制器中被激活的时候，会提示那些试用版授权应该被激活。



（1）在 TwinCAT3 开发环境中也可以手动激活 7 天试用授权，具体步骤如下：
点击 system 下的 license

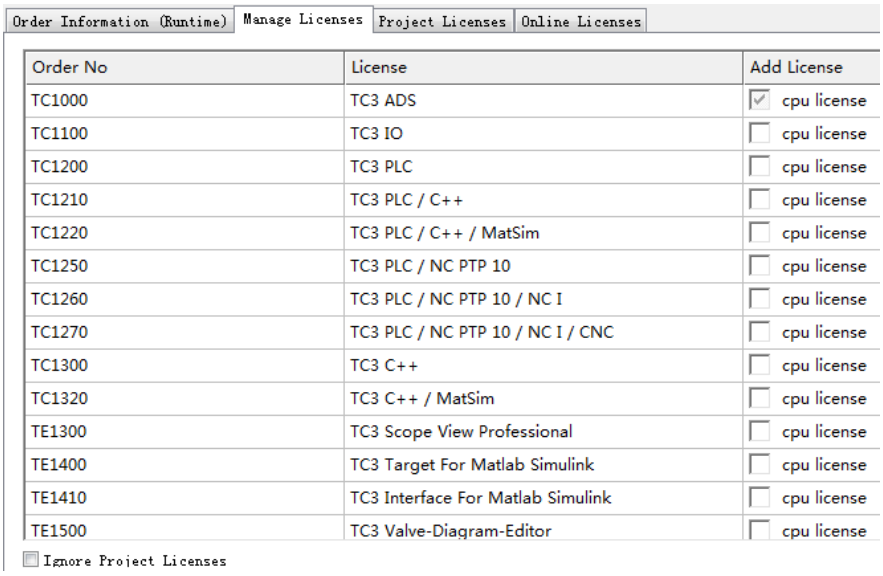


（2）在菜单中有 4 个选项

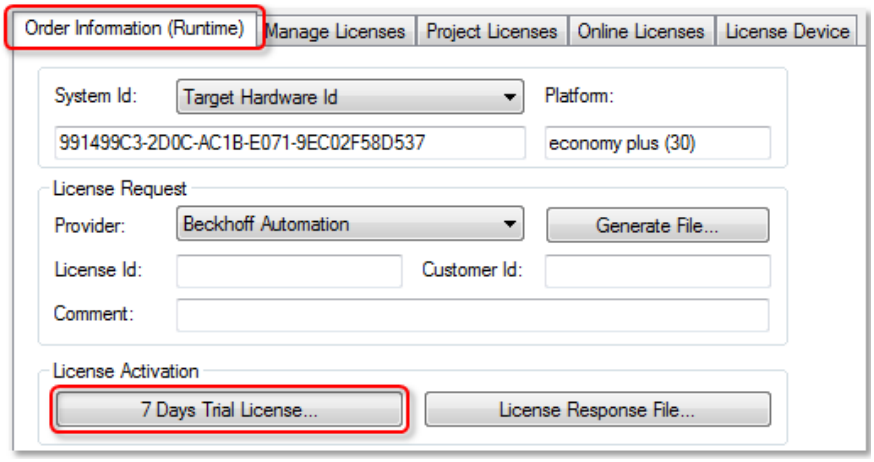


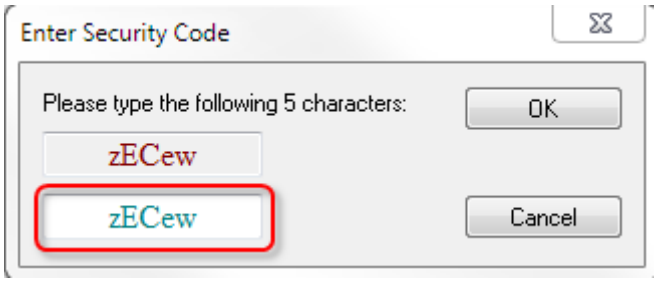
Order Information（Runtime）：试用版和正版激活选项
Manager Licenses：可以手动勾选所需要的 licenses
Project Licenses：当前项目所用到的 licenses
Online Licenses：激活过的 licenses 或者曾经激活过的 licenses

（3）首先选择 manager licenses，在 Add License 中勾选所需要的 License

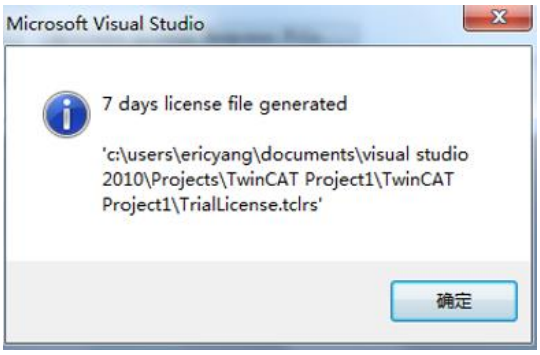


（4）之后回到 order information 选型，点击 Activate 7 Days Trial License，随后会弹出验证码方框，输入 5 位验证码，大小写有区分

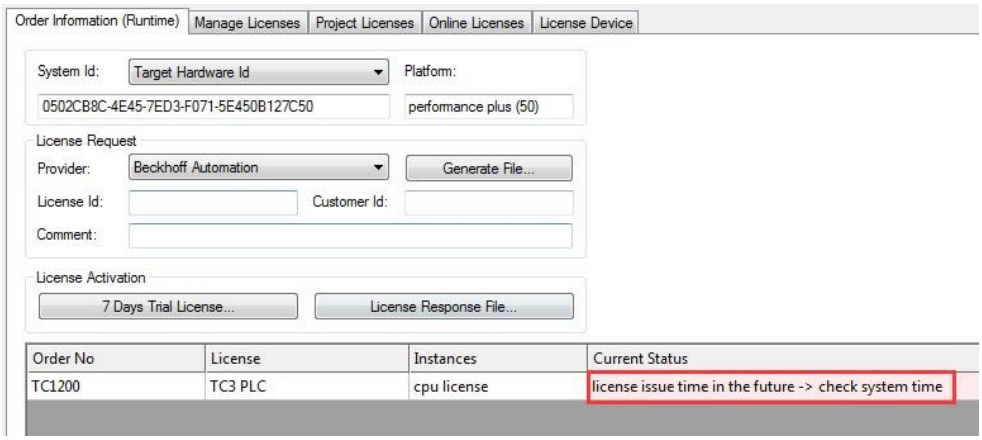




（5）输入正确后点击 OK 会弹出窗口告知 7 天的试用版 license 已经生成，这样我们就可以有 7 天的授权可以用，如果过期了再次用同样的方法激活就可以了



Q：在对目标控制器操作 7 天试用版激活的时候，发现目标控制器始终无法切换到 run-mode，并且在 license 选项卡中发现所做 7 天授权当前状态如下：license issue time in the future？

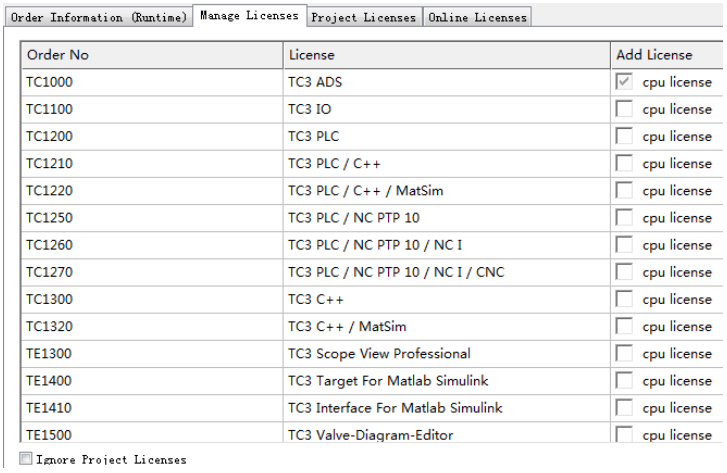


A：主要原因是目标控制器的系统时间和本地 PC 不一致导致，解决方式只需要把目标控制器系统时间手动修改成和本地 PC 一直即可。

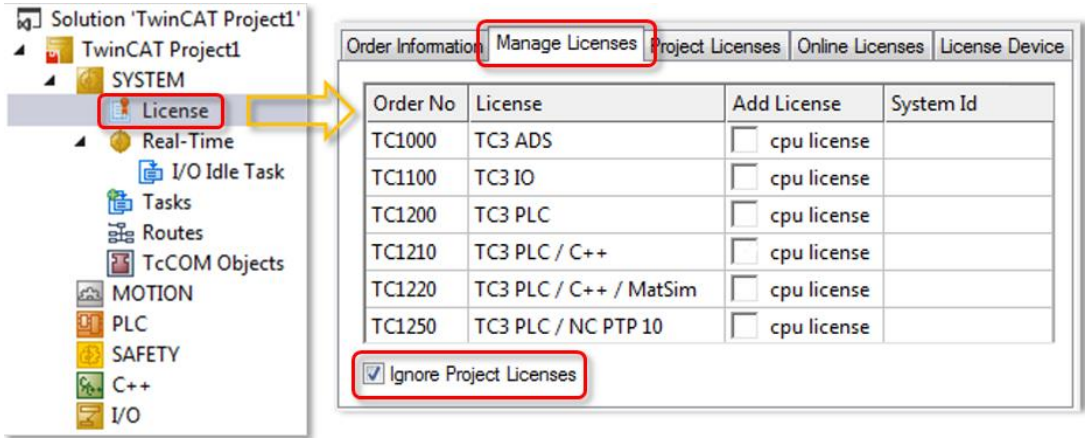
2. IPC/EPC 中的完整版授权激活

通常 TC3 授权和硬件一起购买是不需要我们自己完成授权工作的，只有当授权单独购买，或者后续补丁授权，就需要自己完成这些步骤。

（1）首先选择 manager licenses，在 Add License 中勾选所需要的 License



（2）如果希望自由选择某些 license 不受当前项目应用所影响，可以通过勾选左下角的 Ignore Project Licenses，这样有助于后期补丁某些授权从而完成激活



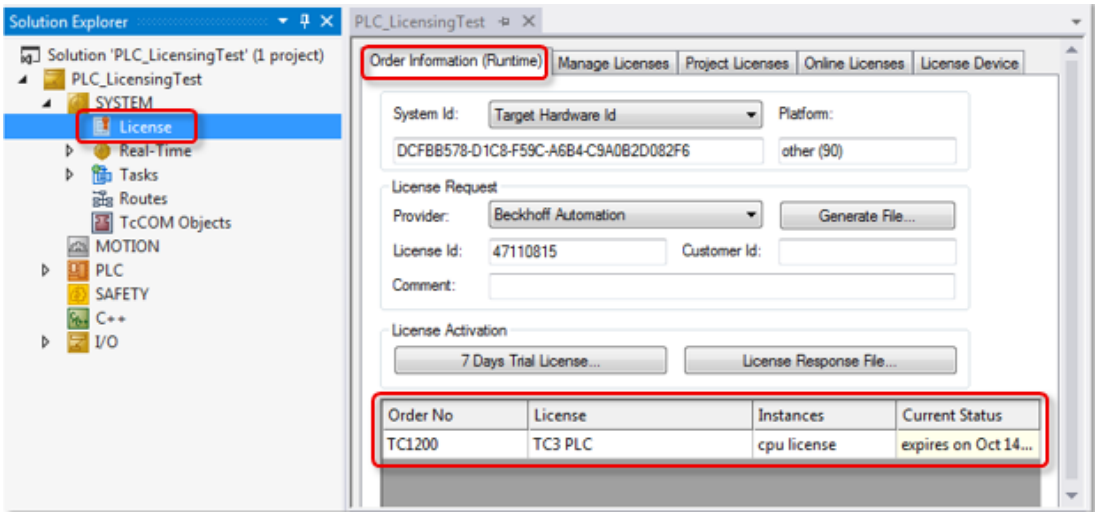
（3）随后回到 Order Information：

System Id：无需填写，只需要确认授权所激活的 PC 或者目标控制器，如果是目标控制器需要连接目标控制器，这里就会显示目标控制器的 System Id，否则就是本地 PC

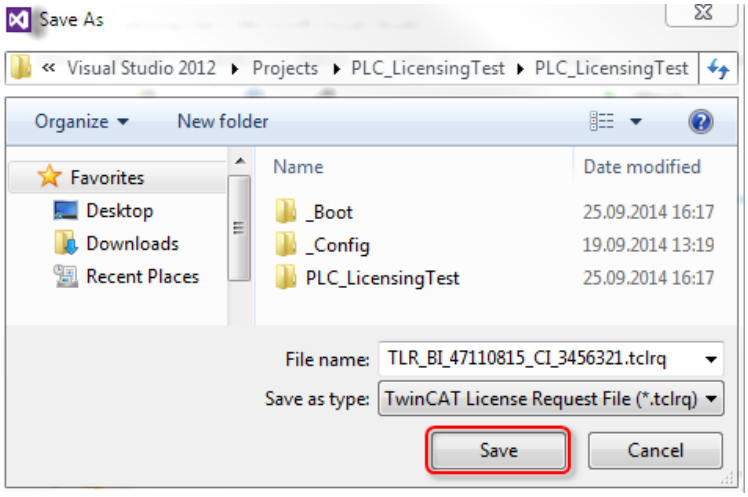
Platform：无需填写，只需要确认订单

License Id：输入购买 License 时候的订单号，如果不知道可以咨询倍福销

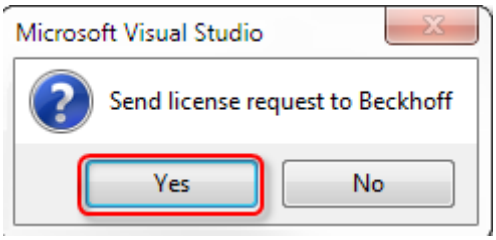
Customer Id：可以任意输入一些数字或者字母（也可以空着不填），通常会填公司名称或者用户名



（4）随后点击 Generate File，系统会生成一个 license 的请求文件，保存到任意位置

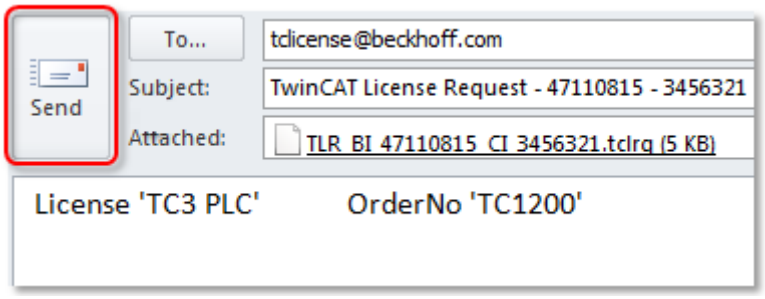


（5）随后弹出对话框询问是否要把此文件发送到 beckhoff

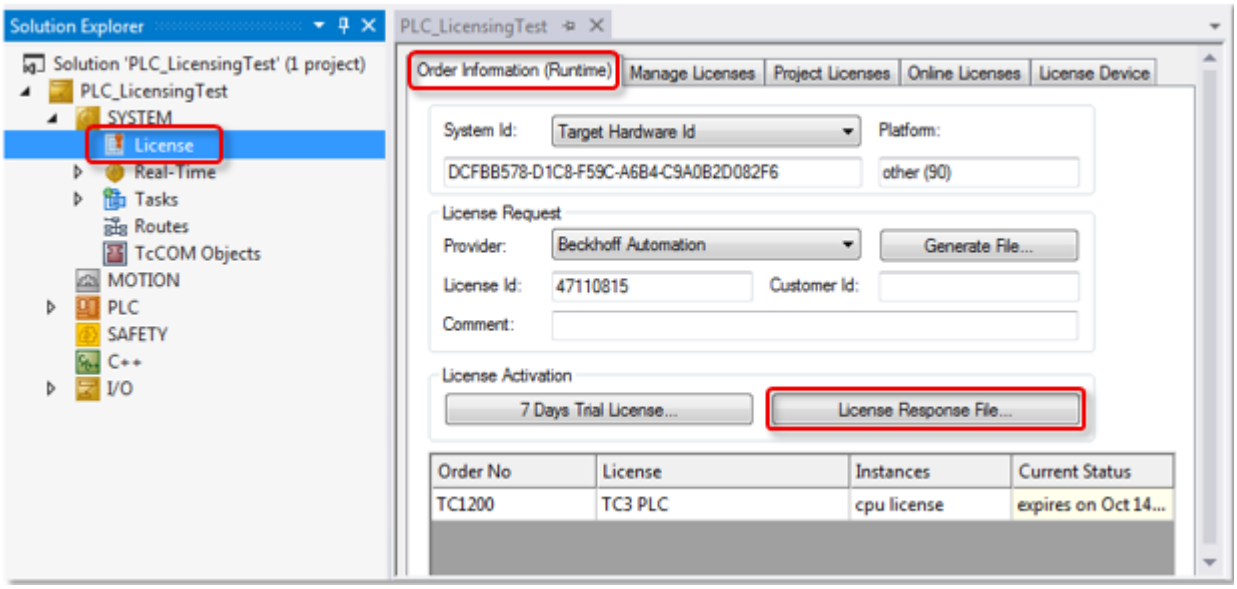


点击“Yes”

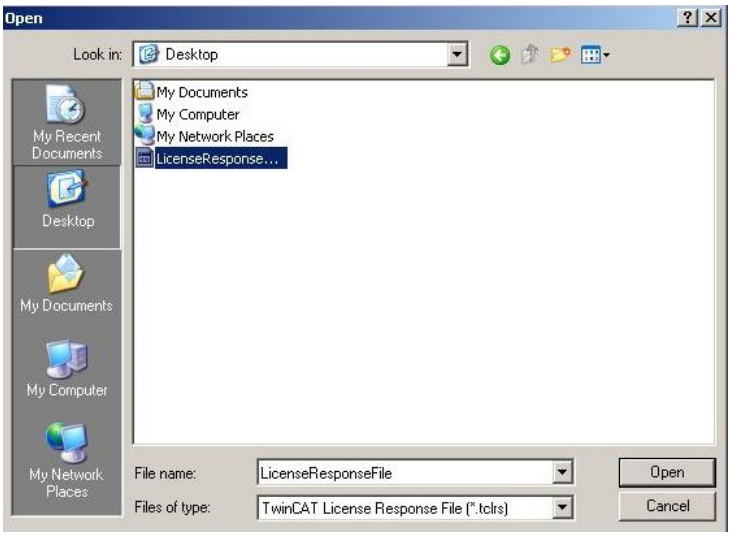
（6）随后系统就会以邮件方式帮您发送到一个固定邮箱 tclicense@beckhoff.com，邮件内容包含了你所申请的 license 和申请文件，申请文件中包括了您的订单号信息，您的用户名信息，您所要激活的这台电脑的 HW 等，当然如果在当前电脑不方便发邮件，或者没有网络，也可以把此文件拷贝出来，在其他电脑中发送此文件到固定邮箱中



（7）如果德国收到邮件后核实相关信息没有错误后，就会返回一个邮件给你，并返回一个激活文件给你，随后点击 Activate License Response File 把激活文件导入就注册正式版成功



（8）导入返回的激活文件



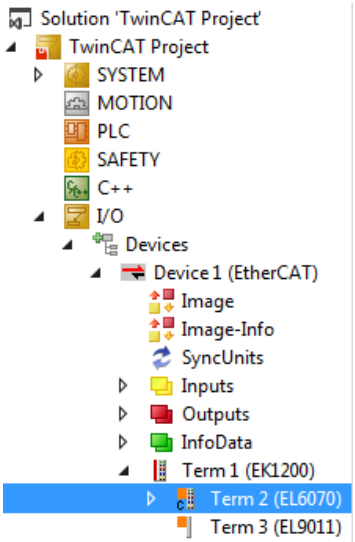
（9）提示以下对话框就说明注册成功，点击 OK 即可

（10）最后注意下这个 license 文件会自动复制到 C:\TwinCAT\3.1\Target\License 中，如果目标控制器是 CE 操作系统，此文件在 \Hard Disk\TwinCAT\3.1\Target\License 中，请备份以免激活文件丢失。

3. Dongle 的使用与授权的激活（TC3.1.4020 老版本）

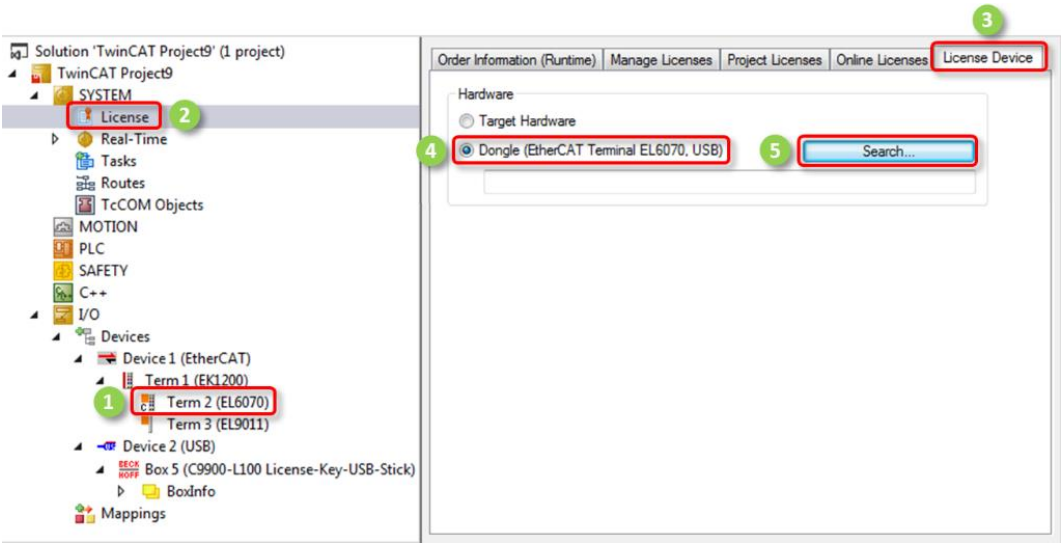
3.1 从 TC3.1.4018.26 开始支持 EL6070 这种模块式组件作为授权工具

（1）首先在 EtherCAT 总线中扫描到 EL6070

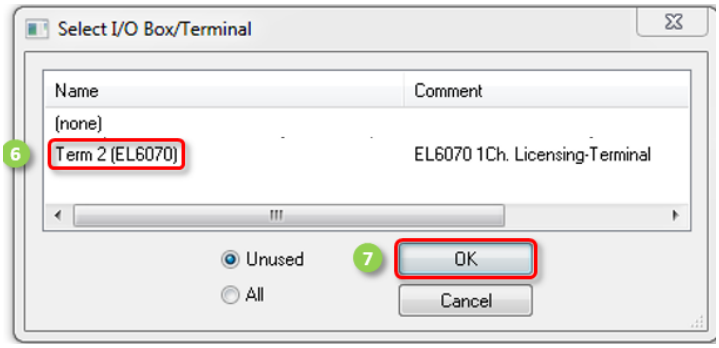


（2）在 SYSTEM→License 中选择 License Device

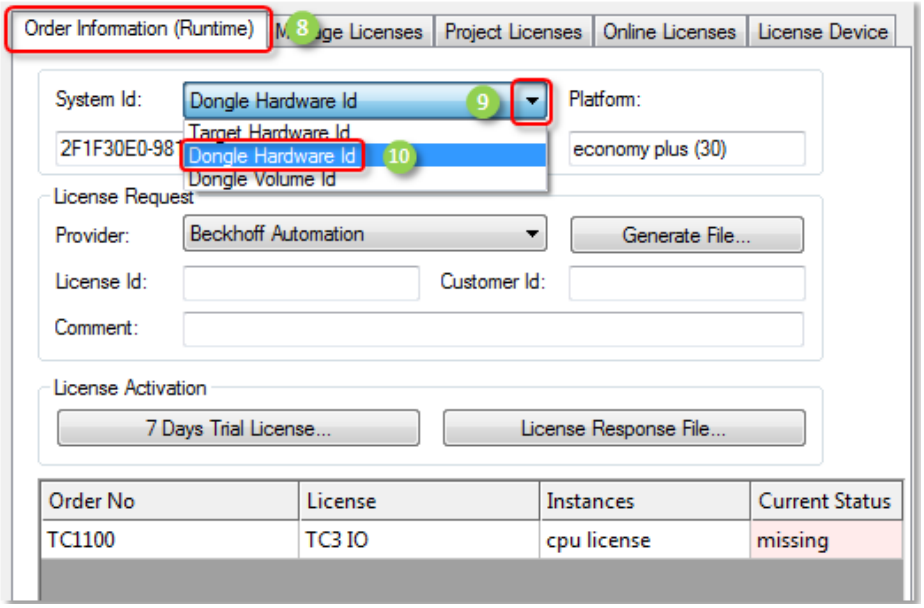
选择 Dongle 后，点击 Search



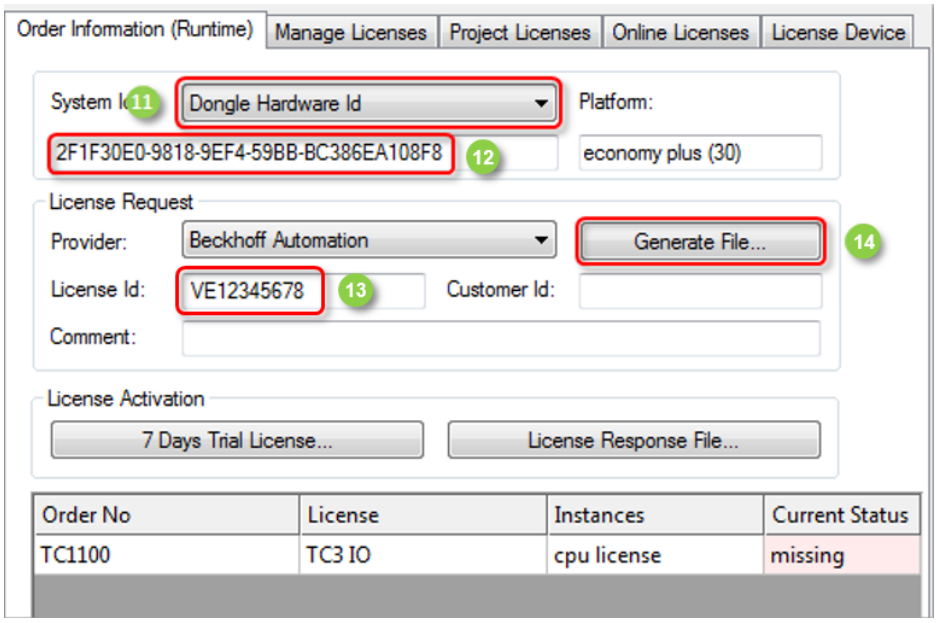
（3）弹出窗口选择 EL6070，点击 OK



（4）回到 Order Information，在 System Id 中下拉框选择 Dongle Hardware Id

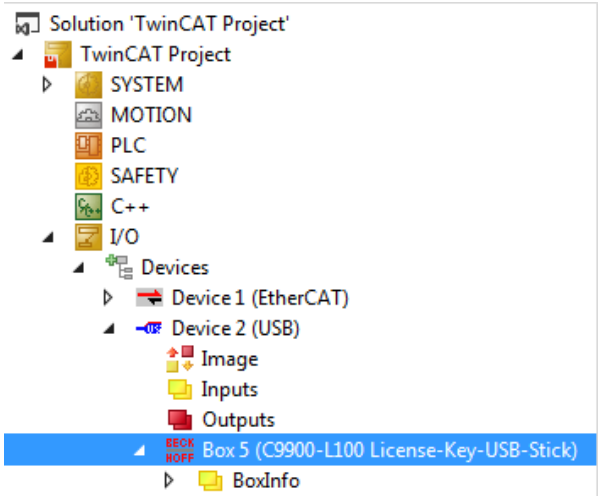


（5）最后输入 License Id，点击 Generate File 生成授权申请文件发送给固定邮箱，就可以得到授权响应文件

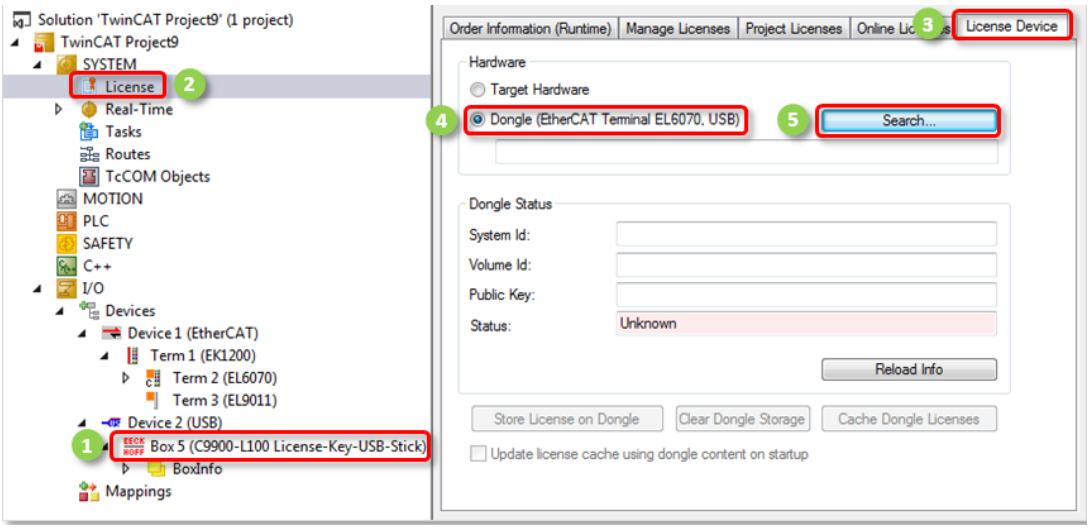


3.2 从 TC3.1.4018.26 开始支持 C9900-L100 这种 USB 类型组件作为授权工具

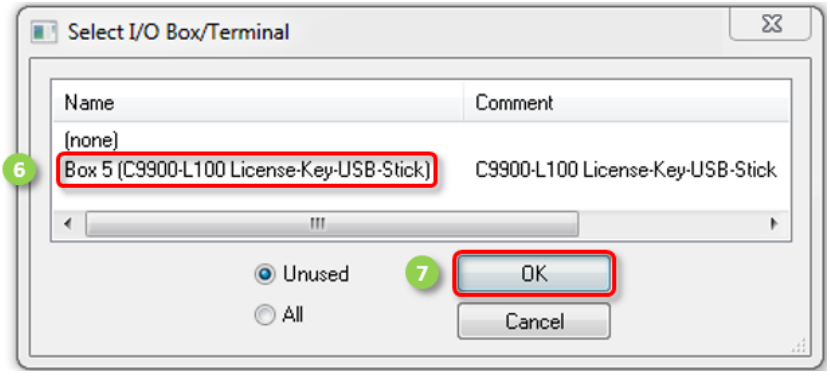
（1）首先在 USB 接口中扫描 C9900-L100



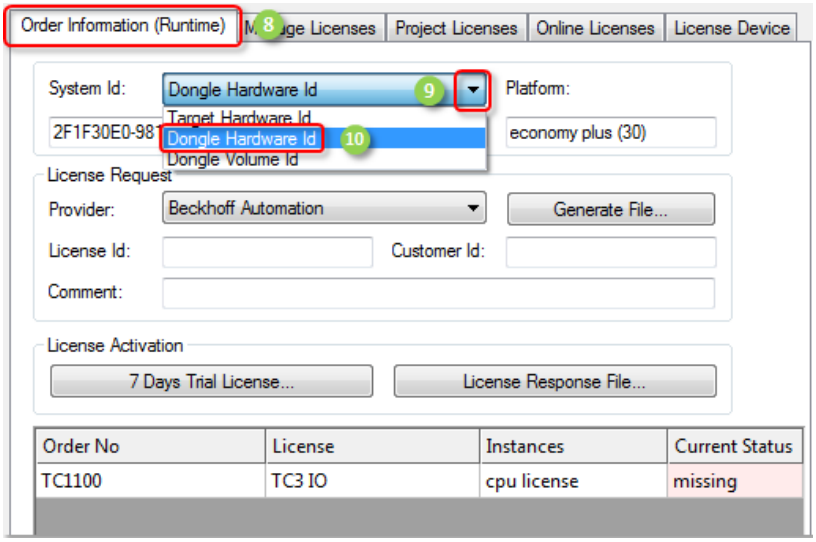
（2）在 SYSTEM→License 中选择 License Device
选择 Dongle 后，点击 Search



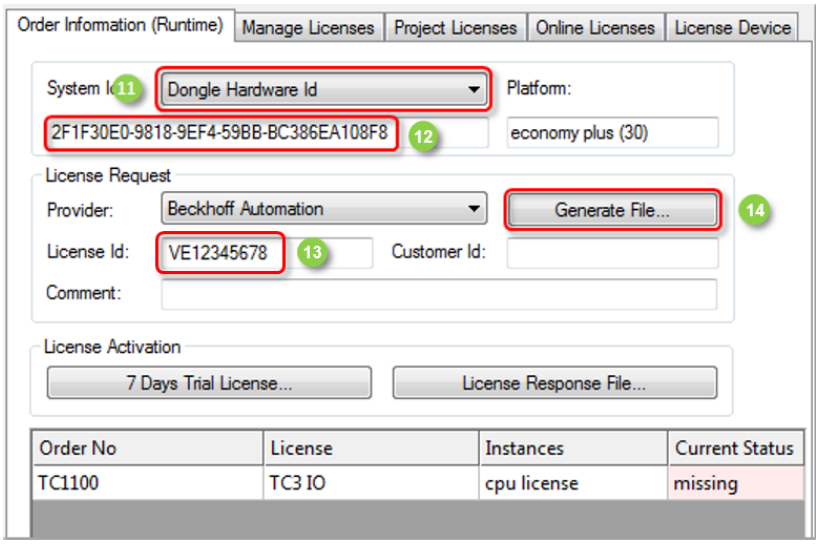
（3）弹出窗口选择 C9900-L100，点击 OK



（4）回到 Order Information，在 System Id 中下拉框选择 Dongle Hardware Id

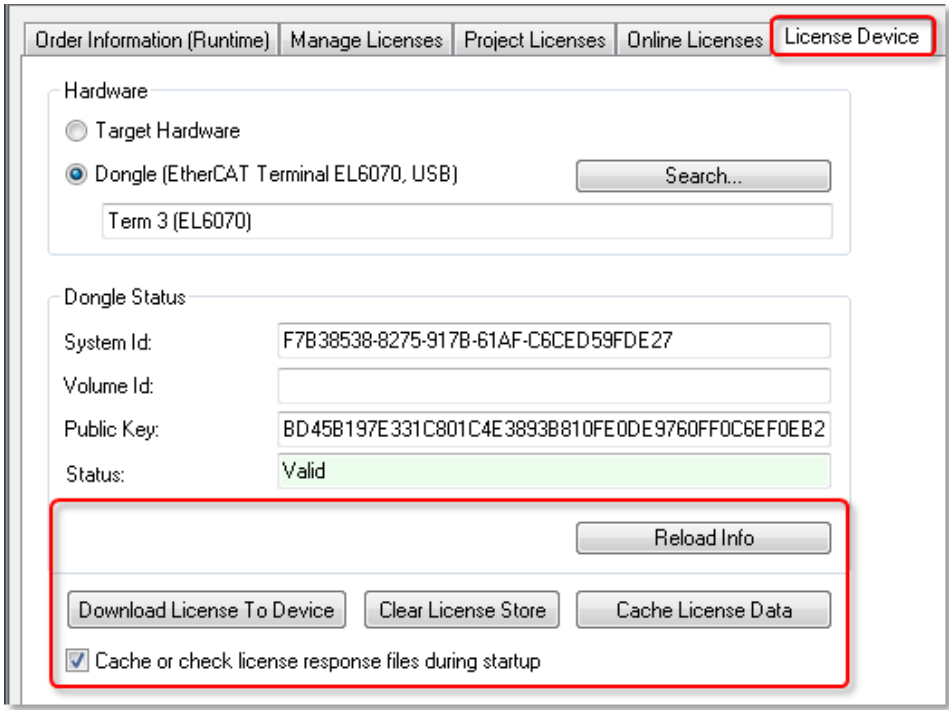


（5）最后输入 License Id，点击 Generate File 生成授权申请文件发送给固定邮箱，就可以得到授权响应文件



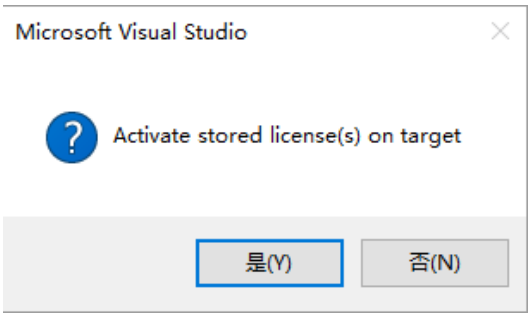
3.3 EL6070 和 C9900-L100 如何使用

（1）得到授权响应文件后，我们需要对 dongle 进行授权，返回到 License Device



因此只需要通过 Download License To Device 就可以把 license 下载到 EL6070 或 C9900-L100 中，使其真正变成移动式授权设备。

（2）完成所有步骤后，每次在 IPC 中插入 C9900-L100 或者在 EPC 后加入 EL6070，并且成功扫描上来后，在 License Device 中都会自动弹出窗口激活 license 到目标系统中，点击“是”就会自动把 dongle 中所存储的 license 全部拷贝到目标文件夹下 C:\TwinCAT\3.1\Target\License 中，如果目标控制器是 CE 操作系统，此文件在 \Hard Disk\TwinCAT\3.1\Target\License 中

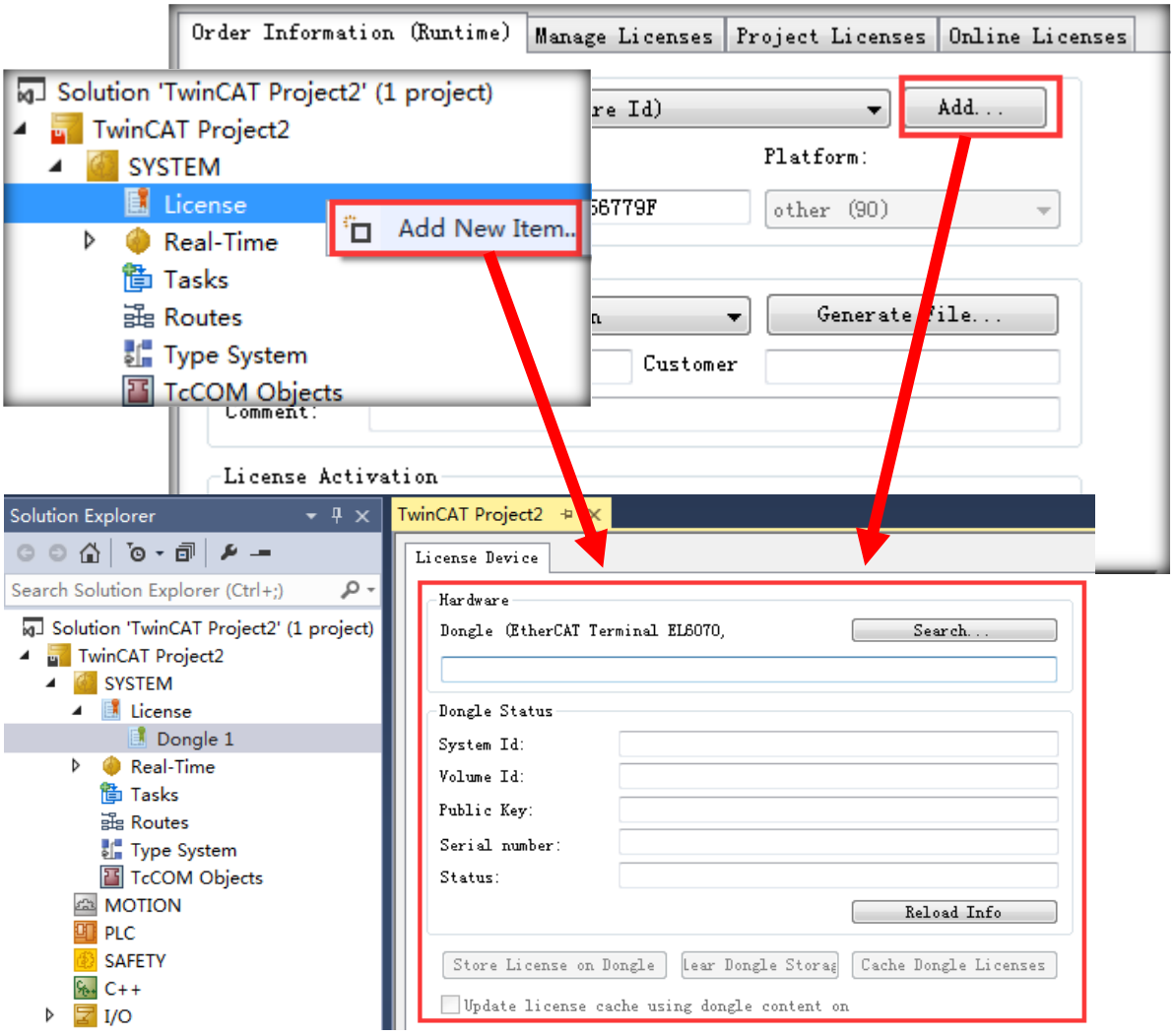


因此只需这 2 步操作就可以轻松利用 dongle 对目标系统完成授权。

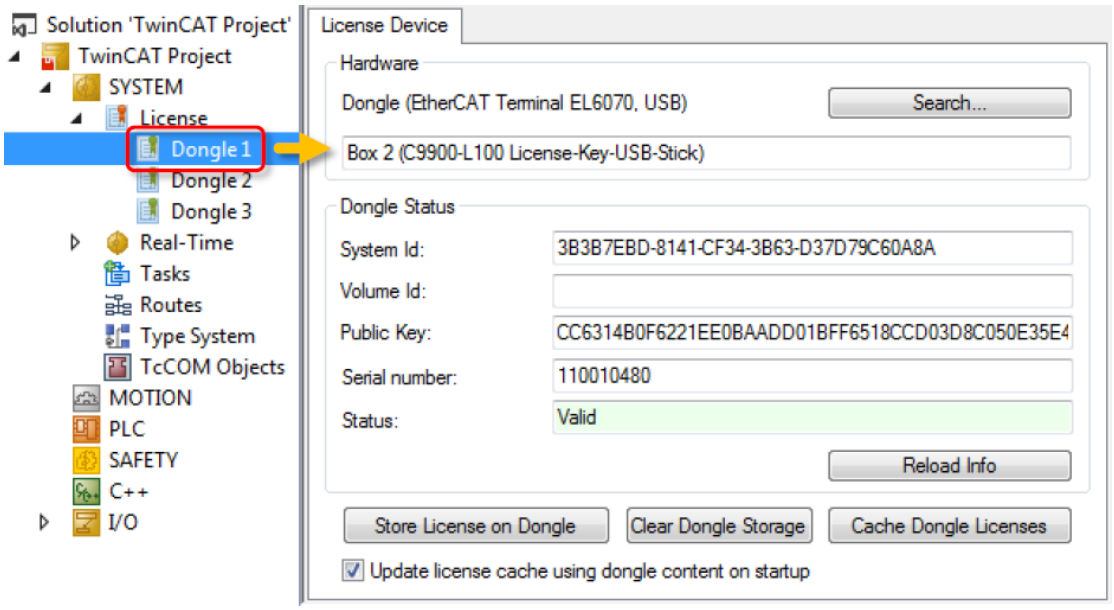
4. Dongle 的使用与授权的激活（TC3.1.4022 新版本）

4.1 从 4022 版本开始 Dongle 可以实现只需配置一次即可自适应，并且配置过程少许发生了变化

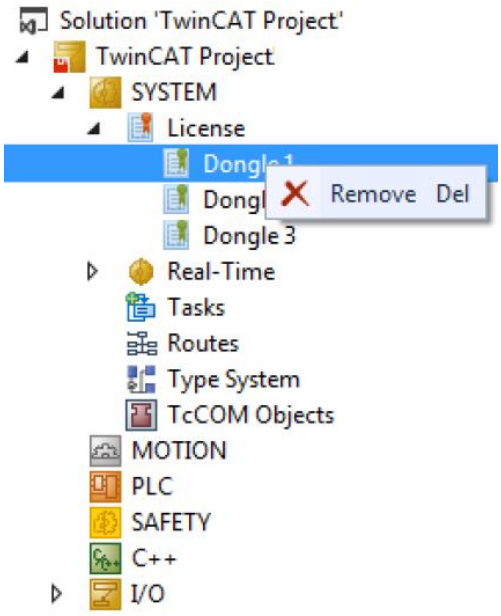
（1）首先需要在 license 中添加一个 dongle，可以右键 license 进行添加，也可以直接在 license→Order Information 中点击 Add 完成添加



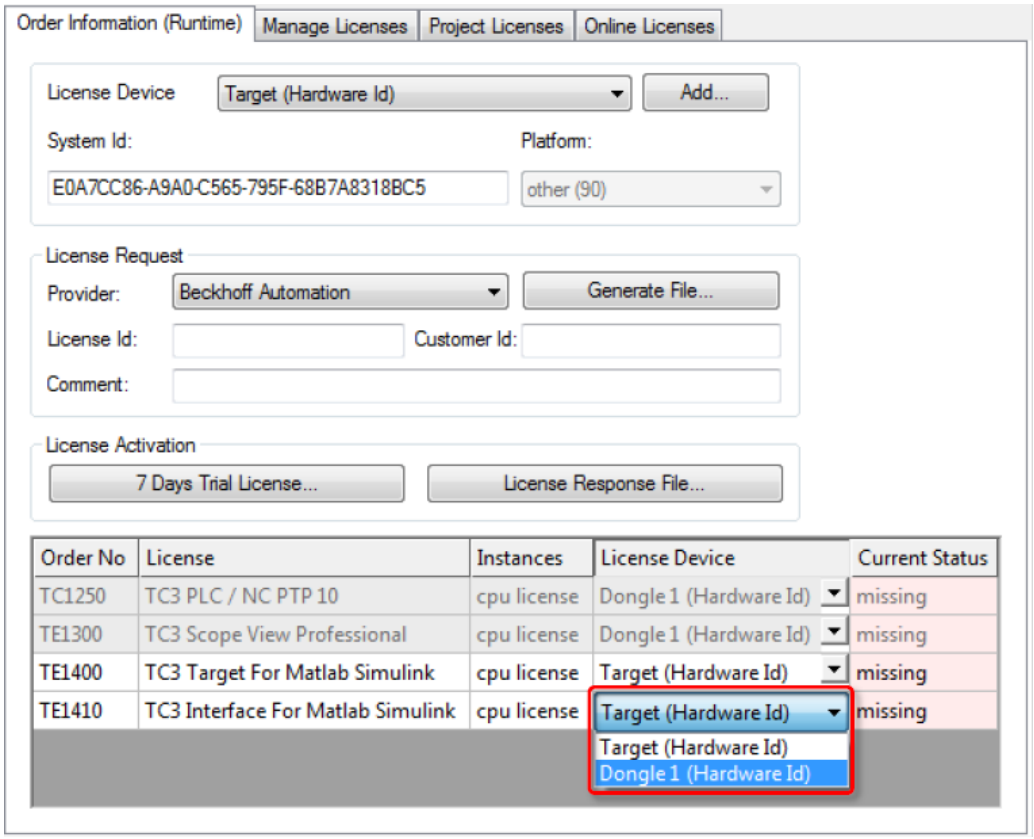
（2）随后点击 Search 选择已经扫描到的 Dongle（也可以同时配置多个 Dongle）



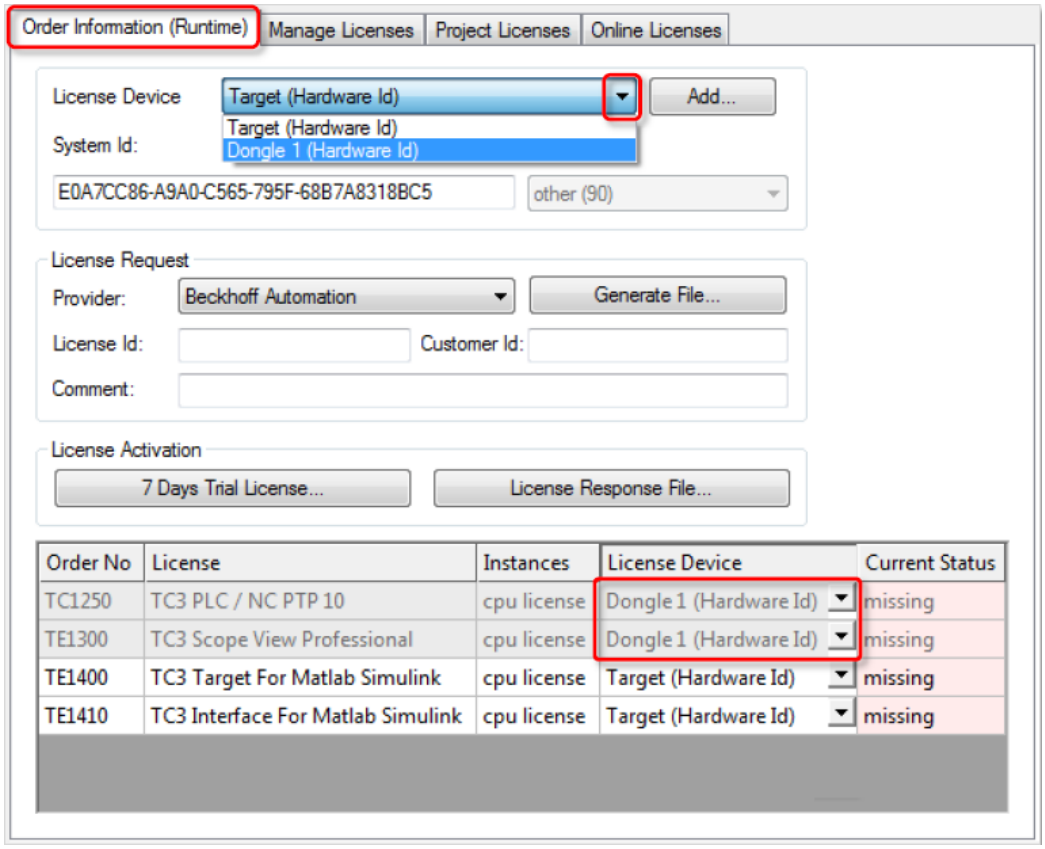
（3）通过右键 Dongle 也可以删除



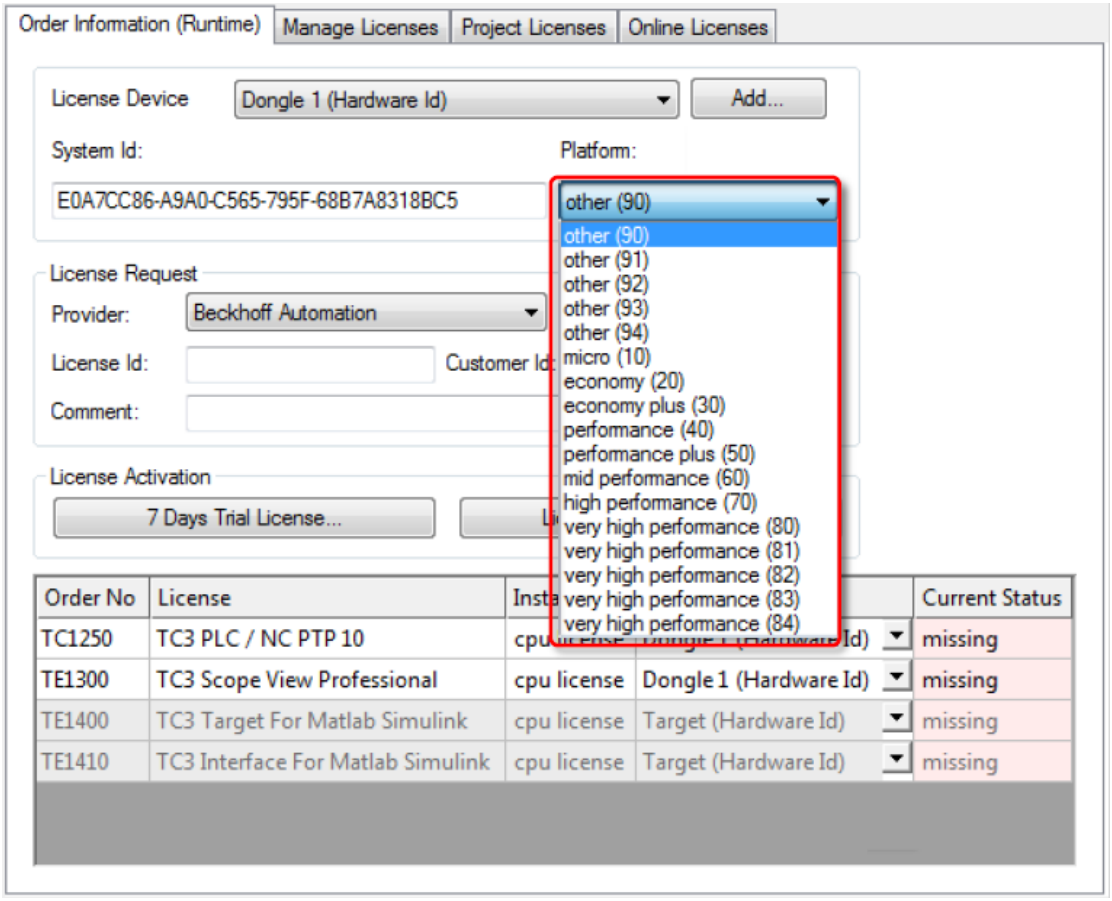
（4）回到 Order Information，可以针对所勾选的授权指定其激活在什么硬件设备中（Dongle 还是 EPC/IPC），选中任意授权都可以在 license Device 下拉框选择 Target（Hardwar Id）还是 Dongle1（Hardware Id）



（5）在 Order Information 最上面的 License Device 中也选择 Dongle1（Hardware Id），这样就可以把 System Id 从控制器切换到 Dongle

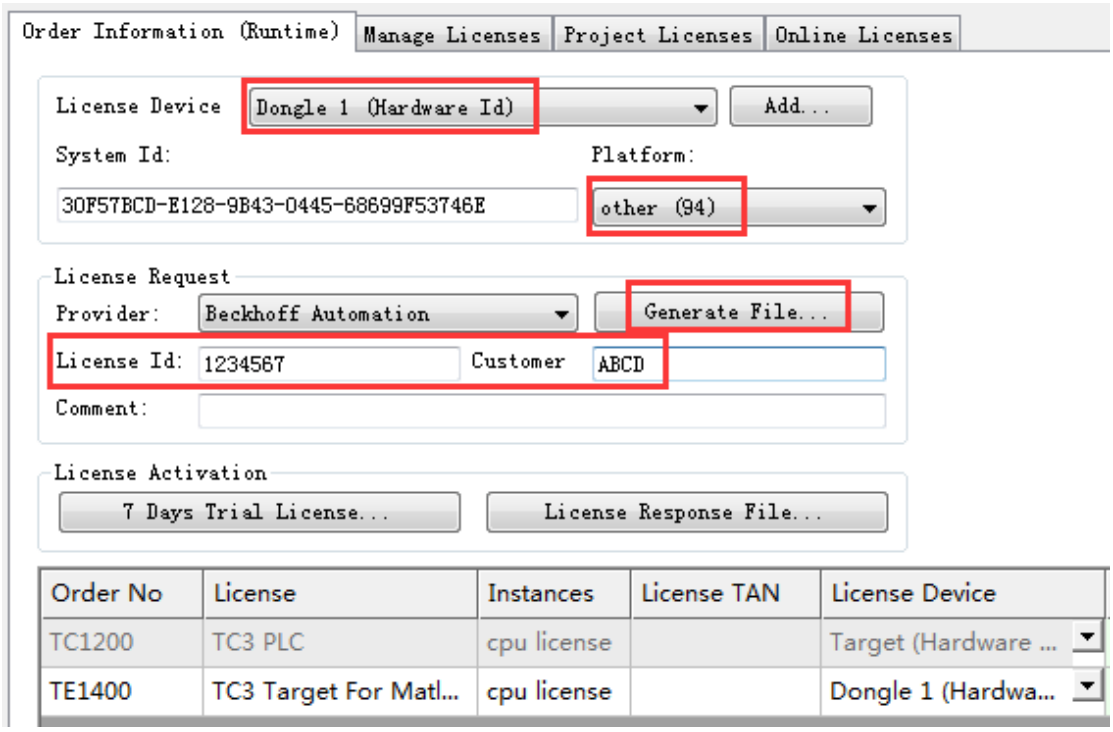


（6）新版本在配置 dongle，尤其是 platform 十分方便，可以通过下拉框选择所购买的等级。



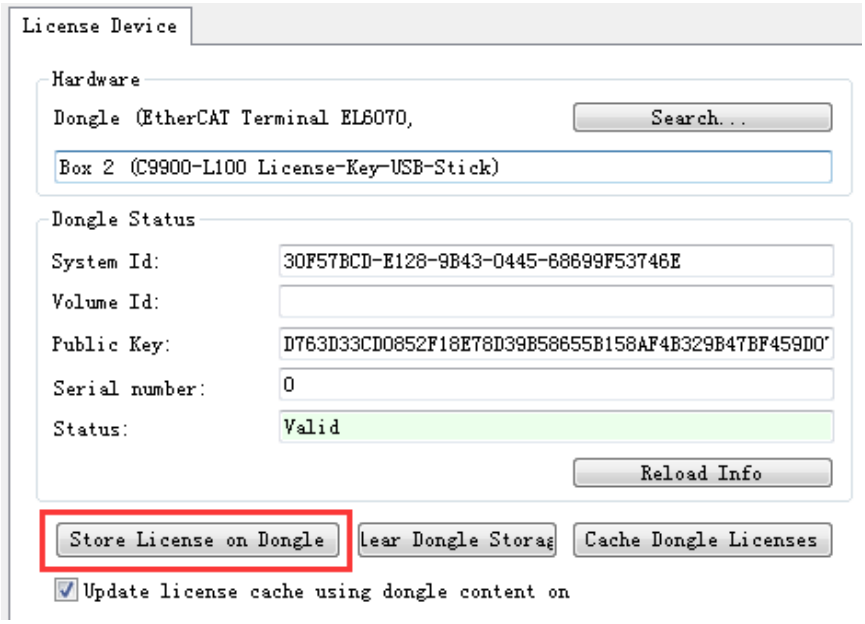
PS：TE（Engineering 授权选择 94 完成授权申请）

（7）最后输入 License Id，点击 Generate File 生成授权申请文件发送给固定邮箱，就可以得到授权响应文件



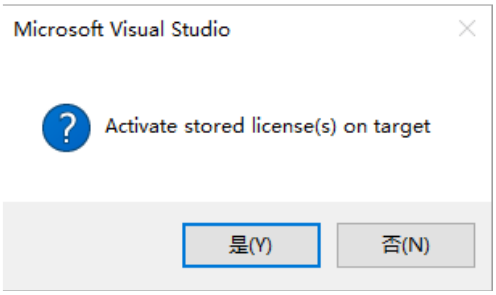
4.2 EL6070 和 C9900-L100 如何使用

（1）授权响应文件后，我们需要对 dongle 进行授权，返回到 Dongle



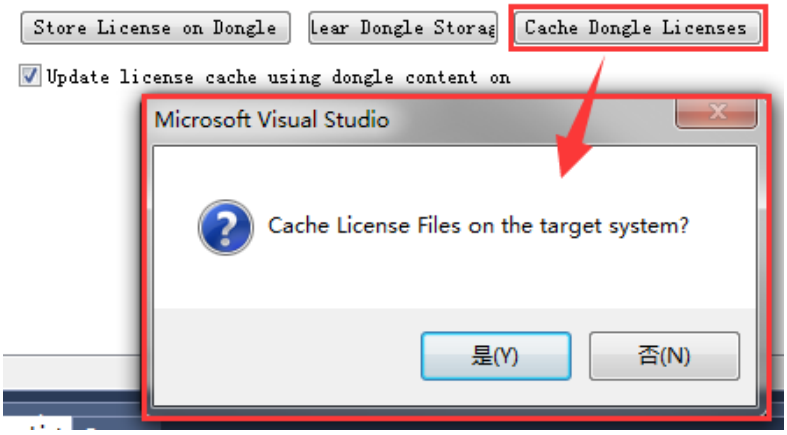
点击其中的 Store License on Dongle，把授权响应文件导入到 Dongle 即可完成 dongle 的授权绑定工作

（2）完成所有步骤后，每次在 IPC 中插入 C9900-L100 或者在 EPC 后加入 EL6070，并且成功扫描上来后，在 License Device 中都会自动弹出窗口激活 license 到目标系统中，点击“是”就会自动把 dongle 中所存储的 license 全部拷贝到目标文件夹下 C:\TwinCAT\3.1\Target\License 中，如果目标控制器是 CE 操作系统，此文件在\Hard Disk\TwinCAT\3.1\Target\License 中



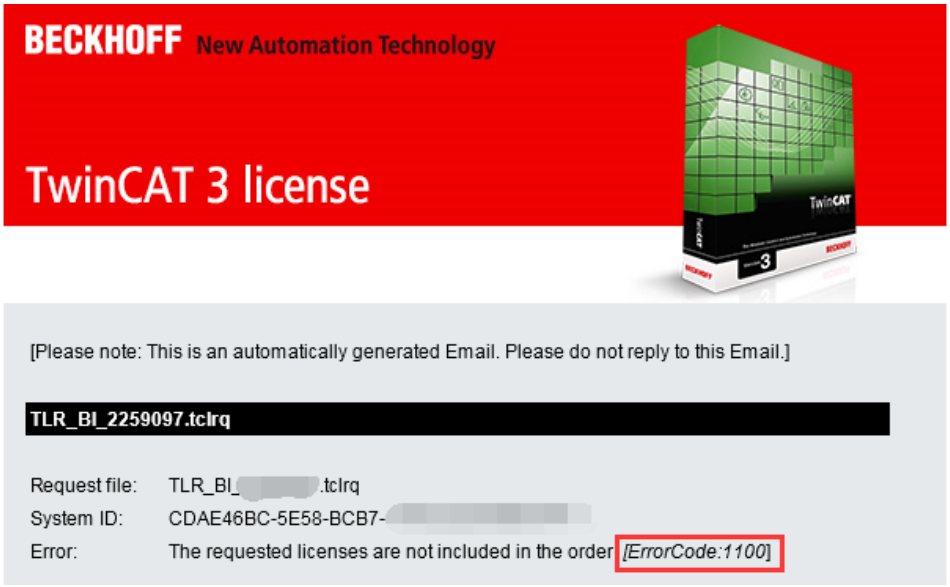
（3）或者手动点击 Cache Dongle Licenses 也可以把授权文件导入到目标设备中。

新版本针对第一次使用的目标设备只需配置一次即可，之后就可以即插即用无需配置。



5. 申请授权报错汇总

无论对于 IPC/EPC 还是对于 dongle 进行授权单独申请的时候，都需要发送申请文件到德国固定邮箱，从而得到响应文件，才可以完成授权激活，但在这个过程中，由于各种原因，难免会有申请失败的情况，不过德国服务器也会针对这个问题返回一些错误代码，比如如下截图：



下面列表就是一些常见错误，以及原因和解决方式

Error ID	Explanation	Procedure
999	The number of product licenses requested in the request file is no longer available for the specified order number.	Please ensure that you have ordered the correct number of licenses, and that these have not already been activated on other devices. Check that no licenses are included in the License Request file, which are not included in the order. *(The License Request file is an XML file and can easily be opened with suitable editors, e.g. Notepad++.)* For general queries relating to your order please contact your Beckhoff sales contact, referring directly to the Beckhoff order number. If the licensing attempt is linked to a hardware replacement, please contact Beckhoff Service.
1000	The order number specified in the request file does not exist.	Please ensure that you have used the correct order number in the "License ID" field. Generate another TwinCAT 3 request file and send it to tclicense@beckhoff.com. For general queries relating to your order please contact your Beckhoff sales contact, referring directly to the Beckhoff order number (VExxxx).
1100	The licenses requested in the request file are not included in the specified order number.	Please ensure that you only request licenses, which you have actually ordered. In case of queries please contact your Beckhoff sales contact, referring directly to the Beckhoff order number (VExxxx).
1200	The transmitted request file does not contain an order number.	Please generate another request file. Ensure that the Beckhoff order number (VExxxx) pertaining to your TwinCAT 3 order was entered in the "License ID" field.
1300	The transmitted request file does not contain an system ID.	Please check whether the correct target system is set! Once the target system has been set, generate another request file and send it to tclicense@beckhoff.com. If the error message persists, please contact Beckhoff Support.
1400	The transmitted request file does not contain a license.	Please check that the required licenses are listed in the "Order Information" tab. All the licenses required by TwinCAT 3 solution are automatically entered there by default. Should this not be the case, please enter the licenses manually via the Manage Licenses tab.
1500	The request file you sent is not readable.	Please generate another TwinCAT 3 request file via TwinCAT XAE and send it to tclicense@beckhoff.com. If the error message persists, please contact Beckhoff Support.

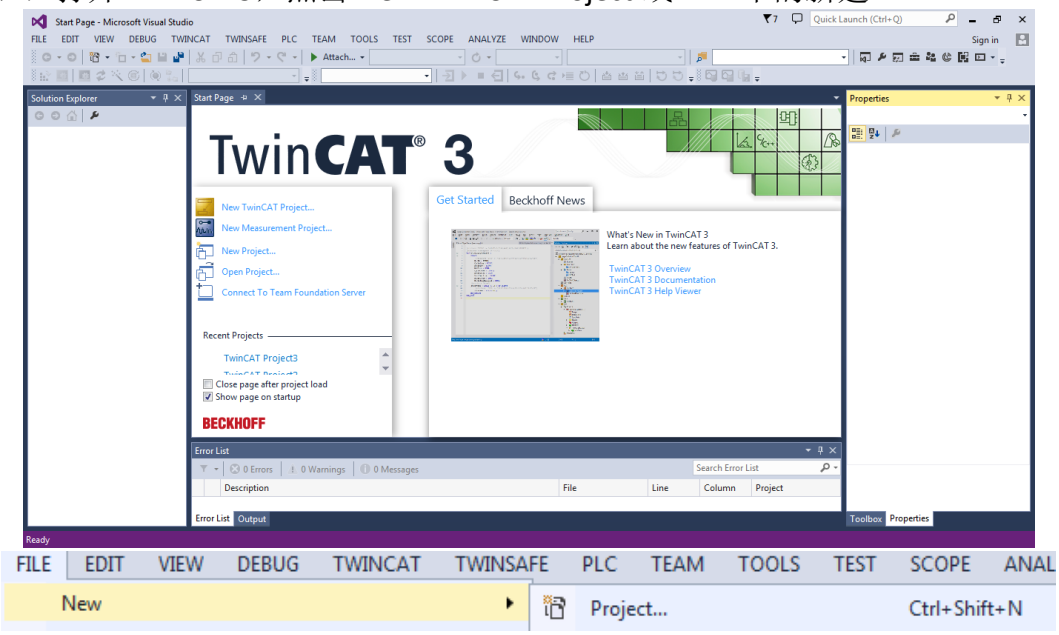
注意：如果返回的 Error ID 在上面列表中不存在，可以等待一段时间，尝试再次发送申请文件到德国固定邮箱

三、 TwinCAT3 扫描 IO 变量连接

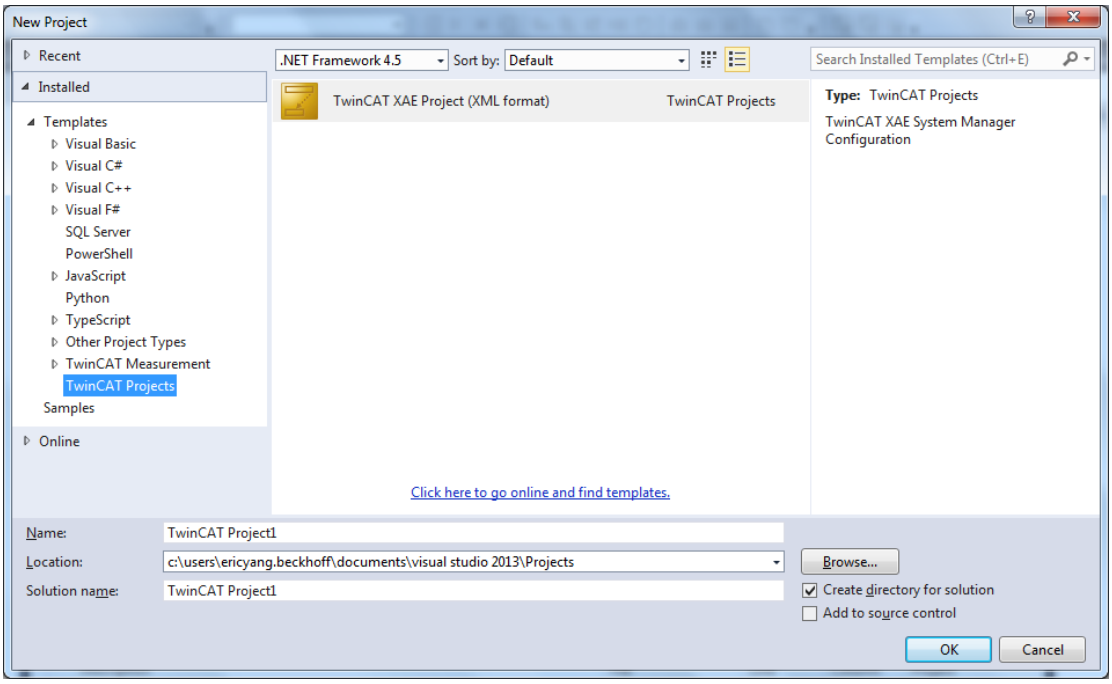
1. 连接目标控制器（TC3.1.4022.30 等老版本）

如果不是本地连接 IO，而是希望对远程目标控制器所连接的 IO 进行配置，那需要进行以下操作。

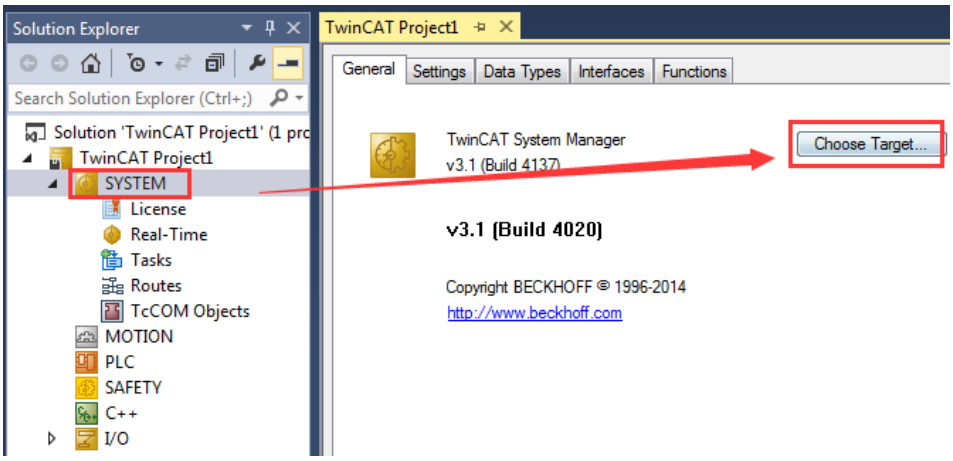
（1）打开 TwinCAT3，点击 New TwinCAT Project 或 FILE 中的新建



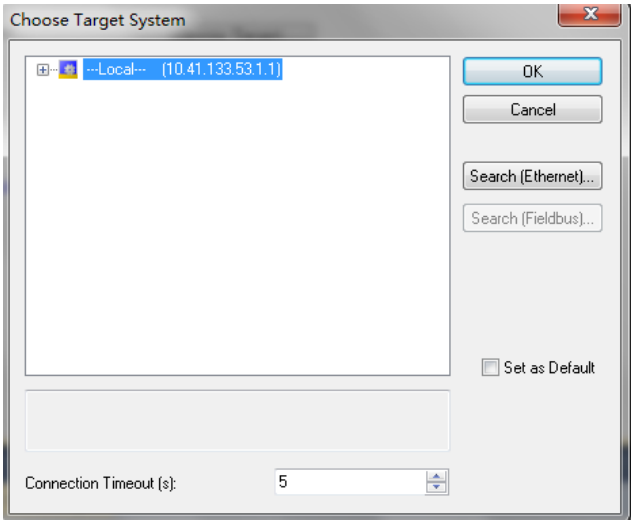
（2）选择 TwinCAT Project 中的 TwinCAT XAE Project(XML format)，并把名字改成英文，例如下图中 TwinCAT Project1，还有别忘了给这个项目一个路径，不然确定按钮是灰色的。



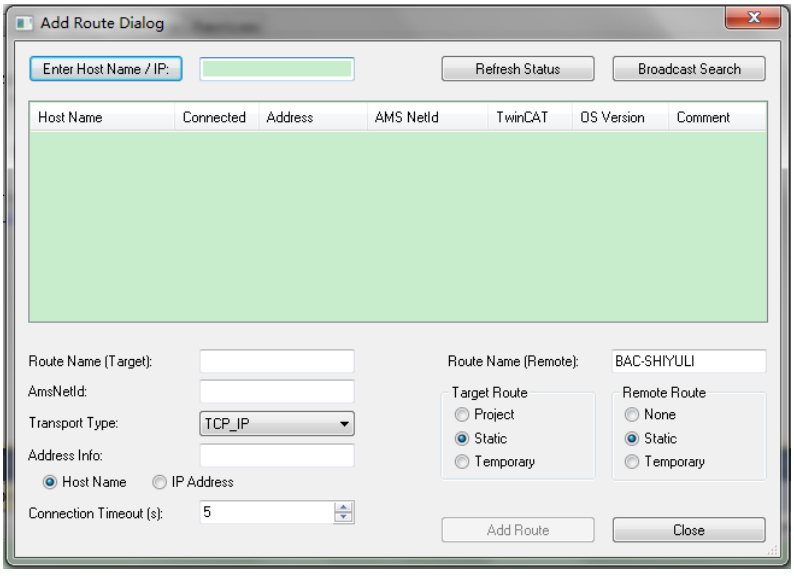
（3）打开 SYSTEM 选型，点击 Choose Target。



（4）在弹出的窗口中点 search（Ethernet）。

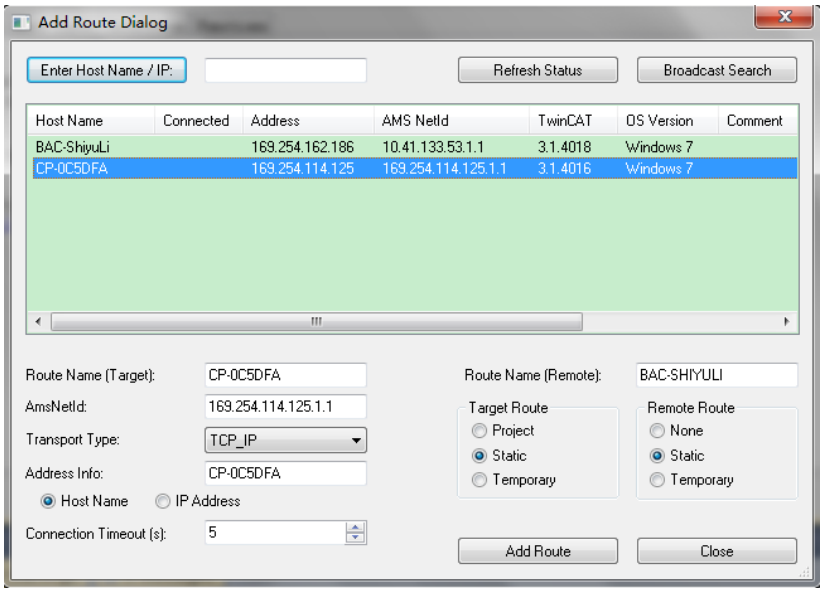


（5）点击 Broadcast Search 进行广播搜索。（多用于不知道目标控制器 IP 地址时）

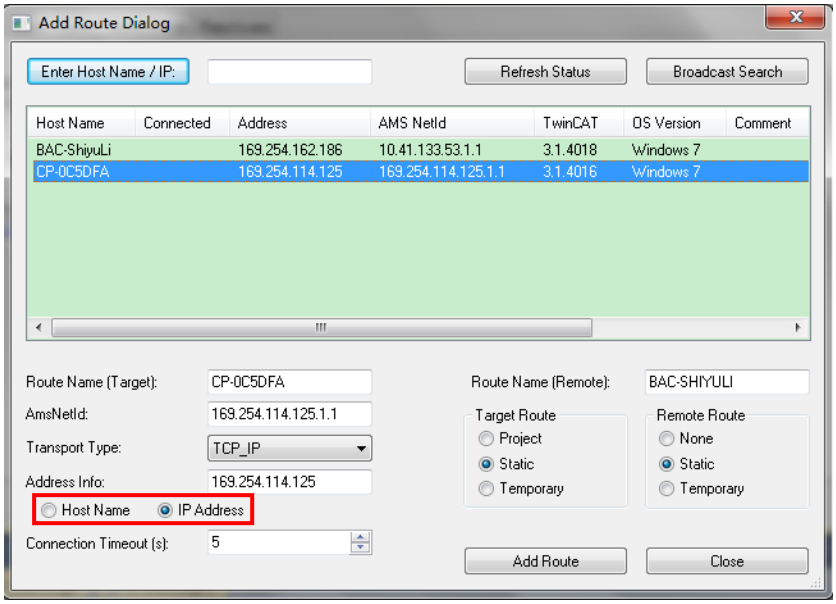


（6）如果已知目标控制器的IP地址，也可以将IP地址填入Enter Host Name/IP右边的输入栏，然后点击Enter Host Name/IP也可以搜索到目标控制器。

（7）当点击 Broadcast Search 或者 Enter Host Name/IP 之后，可以查看是否有搜索到目标控制器，并显示控制器相关信息：
Host Name:控制器的主机名。嵌入式控制器都是 CX-…；工控机都是 CP-…
Connected:连接状态。X 表示连接上，空白表示未连接上
Address：目标控制器 IP 地址
AMS NETID：目标控制器 AMS 地址
TwinCAT：目标控制器 TwinCAT 版本
OS Version：目标控制器 Windows 版本，例如 CE6.0/CE7.0/Win XP/ Windows 7 等
例如下图：



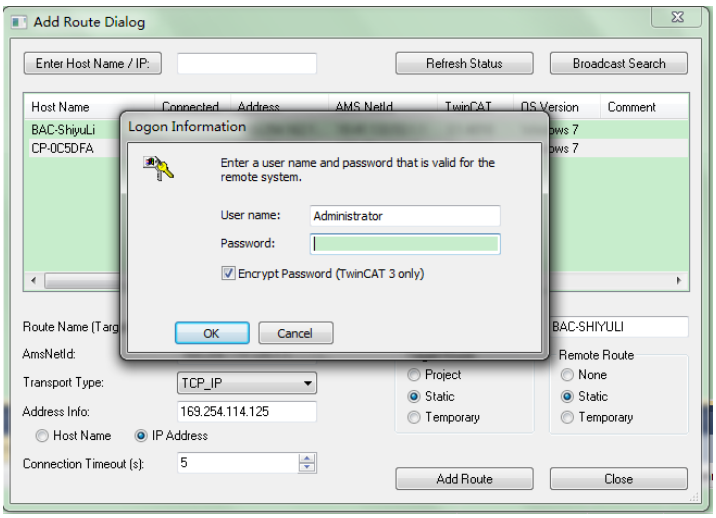
目标控制器的主机名为：CP-0C5DFA
IP 地址为：169.254.114.125
AMS NETID 为：169.154.114.125.1.1
内部 TwinCAT 版本为：3.1.4016
操作系统为：WIN 7
（8）选择搜索到目标控制器之后选择添加方式（Host Name 或 IP Address），并点击 Add Route



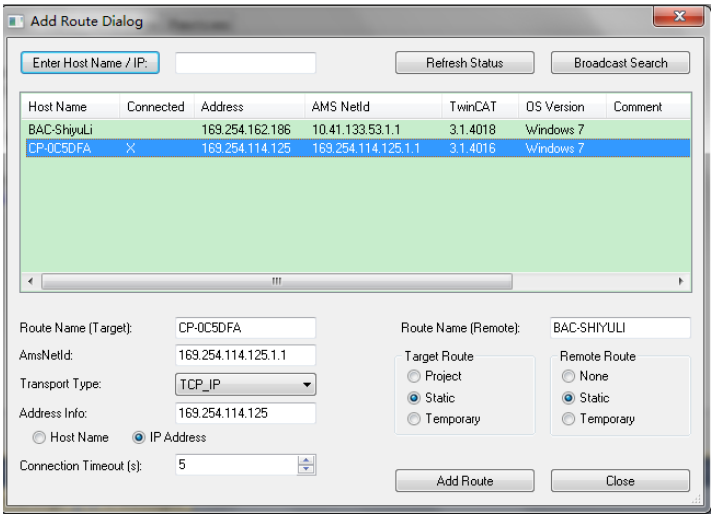
（9）输入用户名和密码

出厂设置：Windows XP/WIN7 ，用户名：Administrator，密码：1；

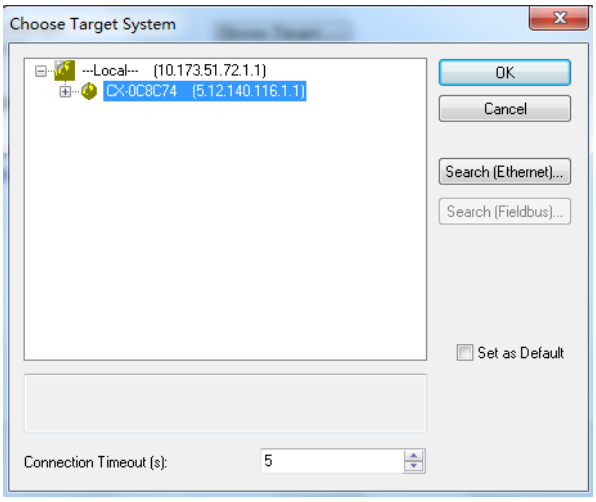
出厂设置：Windows CE ，密码为空白。



（10）添加成功之后 Connected 列显示 X 标记：

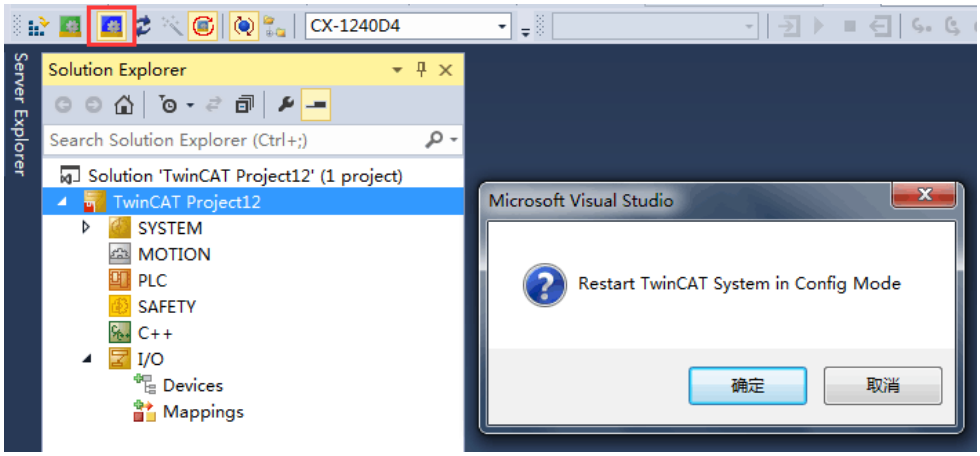


（11）点击 Close，返回前一窗体，此前添加的目标控制器就会出现在这个列表中，选中，点击 OK。



看到工具栏中是目标控制器的名称说明已经连上目标控制器

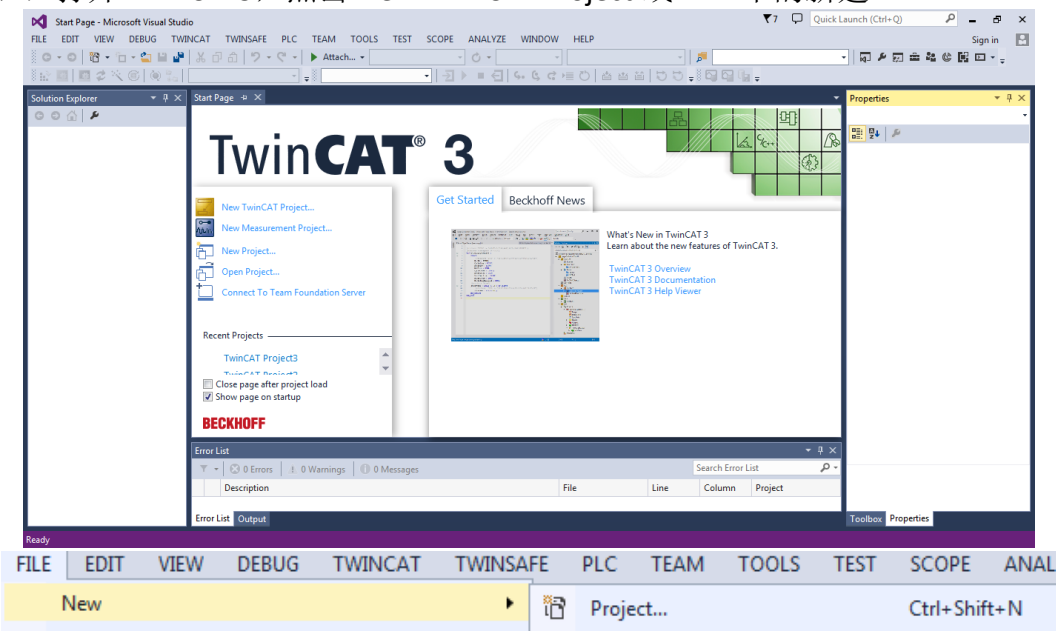
（12）点击工具栏中的蓝色图标把目标控制器切换到 Config Mode，点击确定



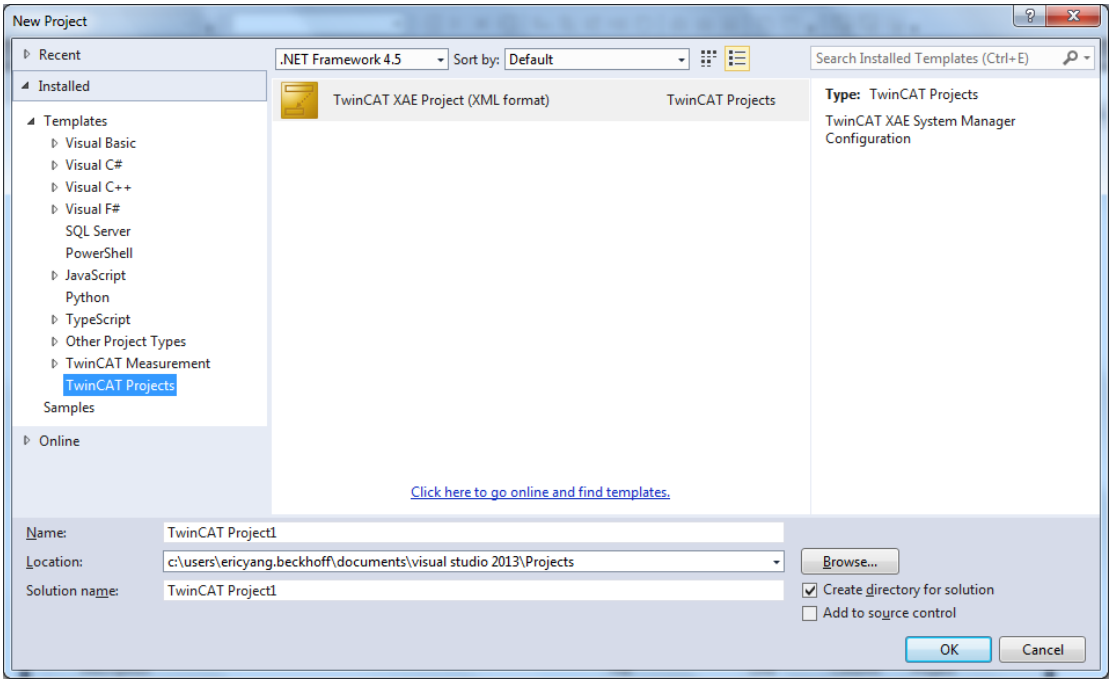
2. 连接目标控制器（TC3.1.4024.0 新版本）

如果不是本地连接 IO，而是希望对远程目标控制器所连接的 IO 进行配置，那需要进行以下操作。

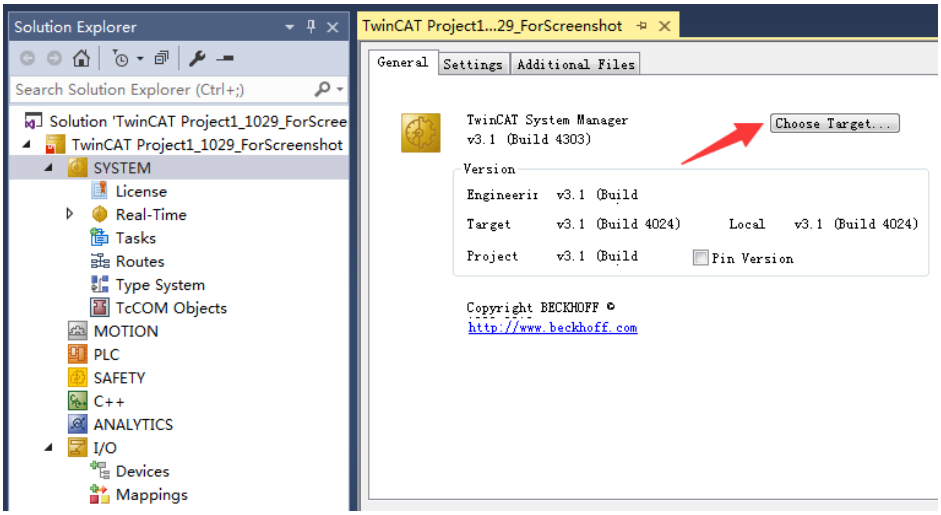
（1）打开 TwinCAT3，点击 New TwinCAT Project 或 FILE 中的新建



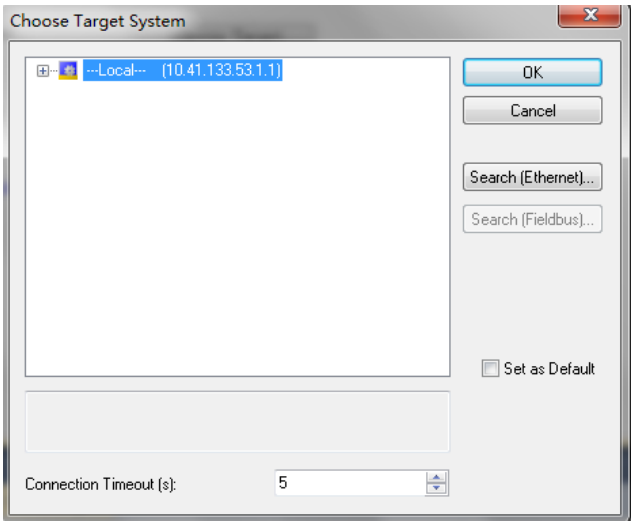
（2）选择 TwinCAT Project 中的 TwinCAT XAE Project(XML format)，并把名字改成英文，例如下图中 TwinCAT Project1，还有别忘了给这个项目一个路径，不然确定按钮是灰色的。



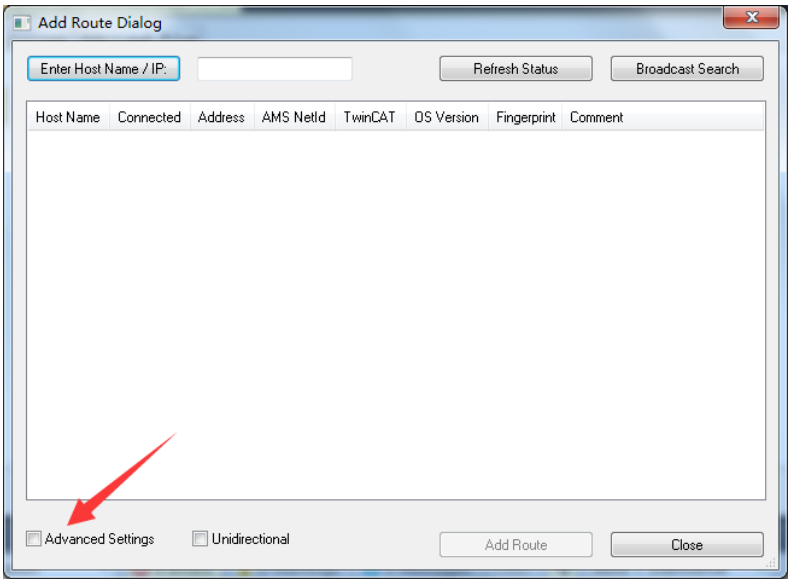
（3）打开树形窗口中的 SYSTEM，在 General 选项卡中点击小按钮 Choose Target。



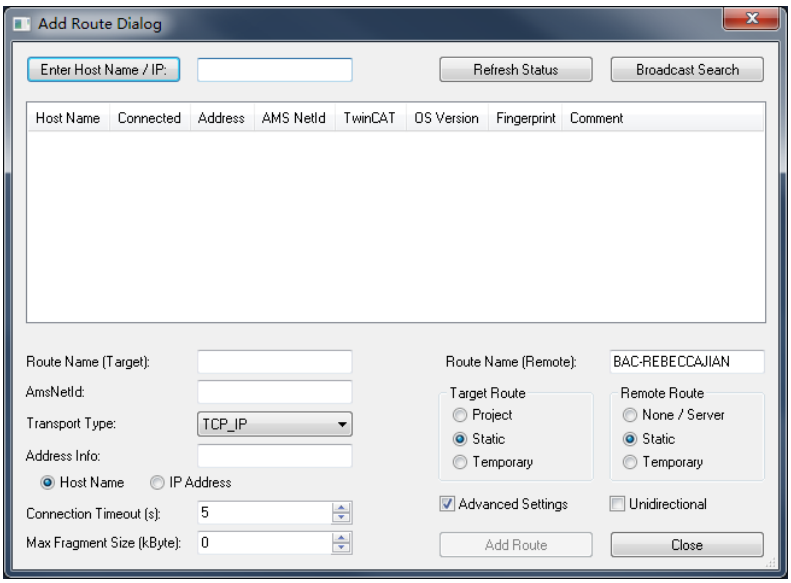
（4）在弹出的窗口中点 search（Ethernet）。



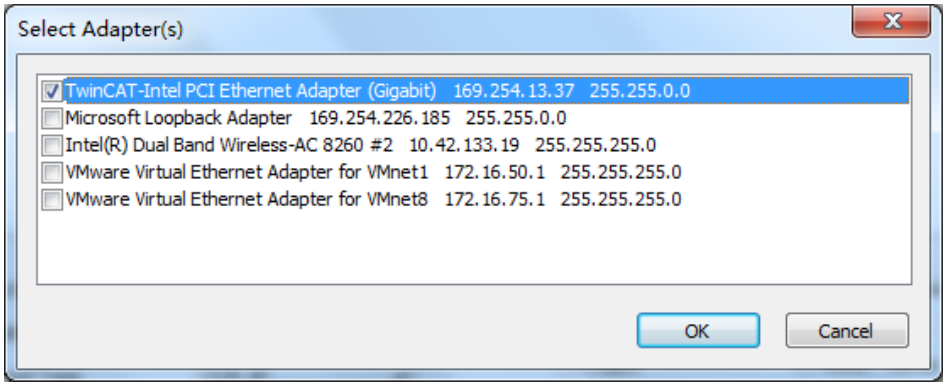
（5）出现 Add Route 对话框以后，我们将右下角的 Advanced Setting 勾选起来



（6）点击 Broadcast Search 进行广播搜索。（多用于不知道目标控制器 IP 地址时）

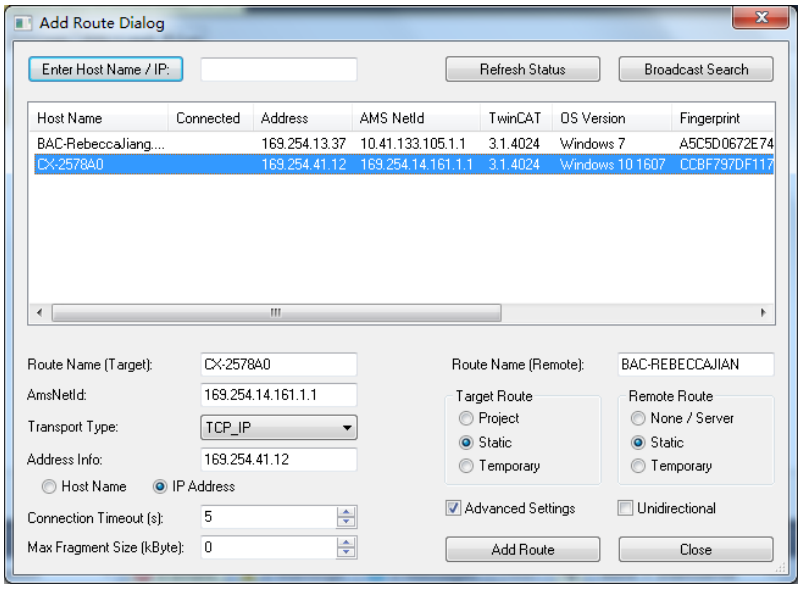


（7）然后可以选择已经和控制器连接的网卡适配器

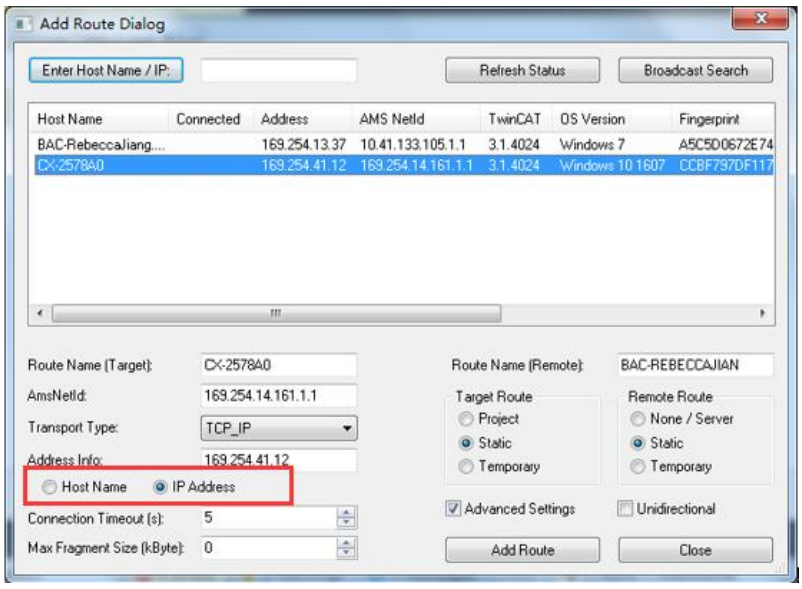


（8）如果已知目标控制器的IP地址，也可以将IP地址填入Enter Host Name/IP右边的输入栏，然后点击Enter Host Name/IP也可以搜索到目标控制器。

（8）当点击 Broadcast Search 或者 Enter Host Name/IP 之后，可以查看是否有搜索到目标控制器，并显示控制器相关信息：
Host Name:控制器的主机名。嵌入式控制器都是 CX-…；工控机都是 CP-…
Connected:连接状态。X 表示连接上，空白表示未连接上
Address：目标控制器 IP 地址
AMS NETID：目标控制器 AMS 地址
TwinCAT：目标控制器 TwinCAT 版本
OS Version: 目标控制器 Windows 版本，例如 CE6.0/CE7.0/Win XP/ Windows 7/Windows10 等，例如下图：



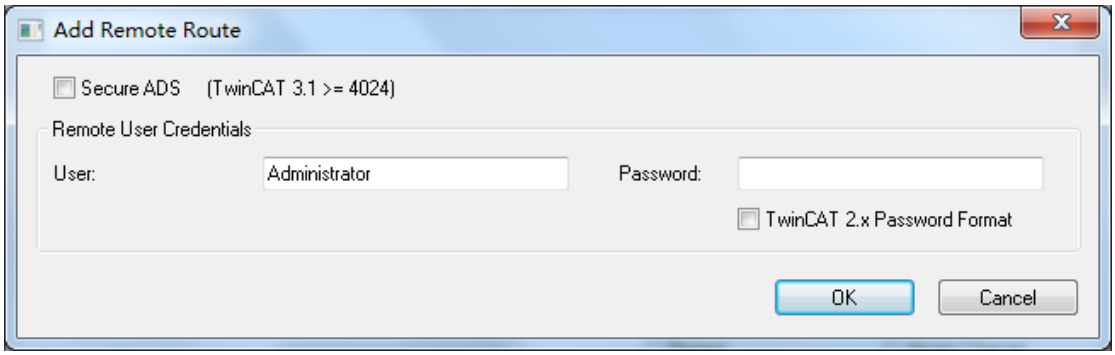
目标控制器的主机名为：CX-2578A0
IP 地址为：169.254.41.12
AMS NETID 为：169.154.14.161.1.1
内部 TwinCAT 版本为：3.1.4024
操作系统为：Windows10 1607
（8）选择搜索到目标控制器之后选择添加方式（Host Name 或 IP Address），并点击对话框右下角的 Add Route



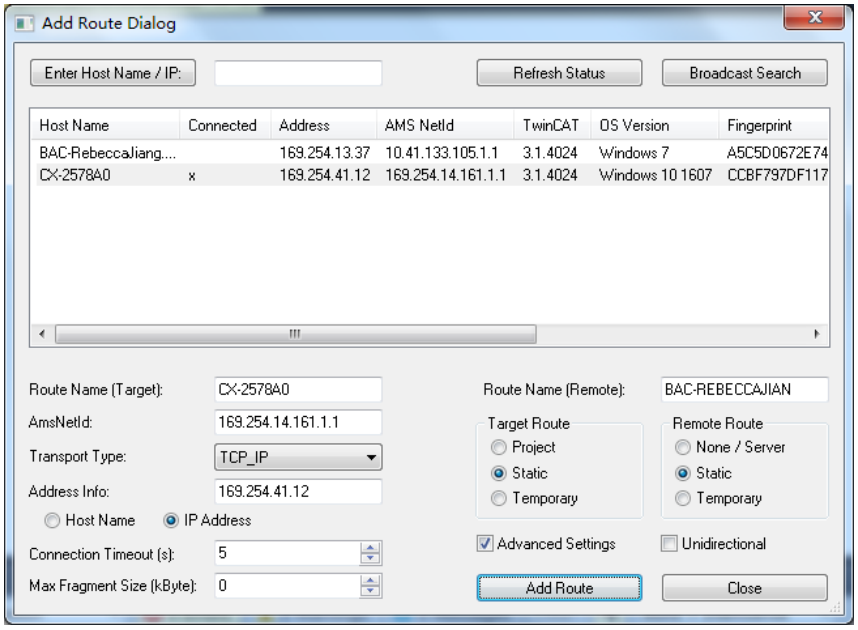
（9）输入用户名和密码

出厂设置：用户名：Administrator，密码：1；

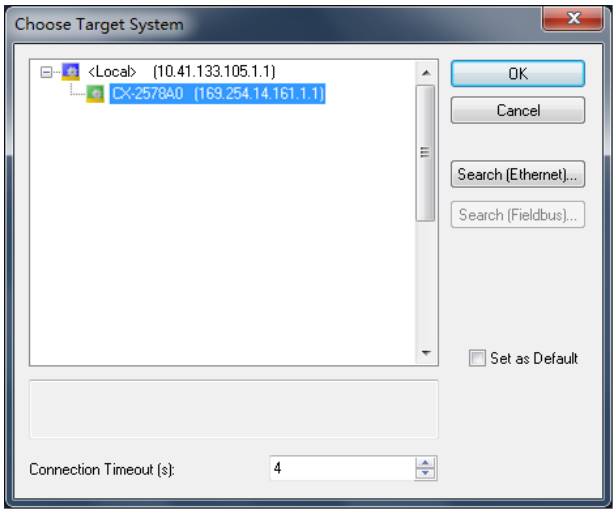
（4024 版本以后不管是 WIN7、WIN10、WES7、CE 系统，出厂默认都如上，此处与 4022 版本或更老版本不同）



（10）添加成功之后 Connected 列显示 X 标记：

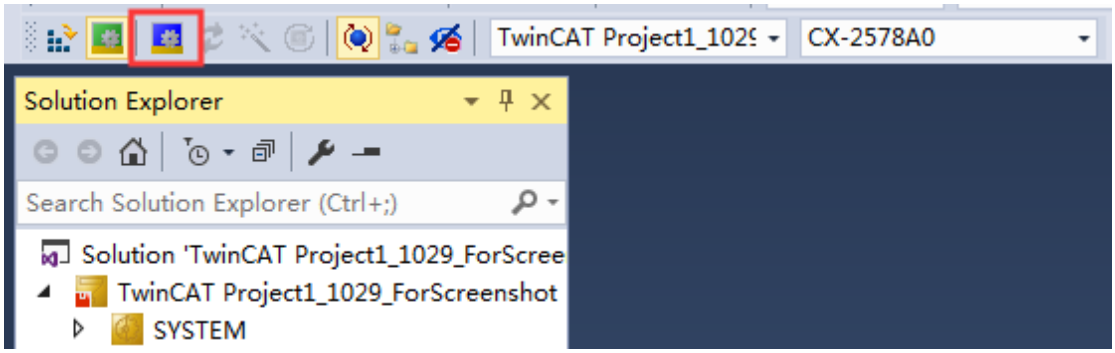


（11）点击 Close，返回前一窗体，此前添加的目标控制器就会出现在这个列表中，选中，点击 OK。



看到工具栏中是目标控制器的名称说明已经连上目标控制器

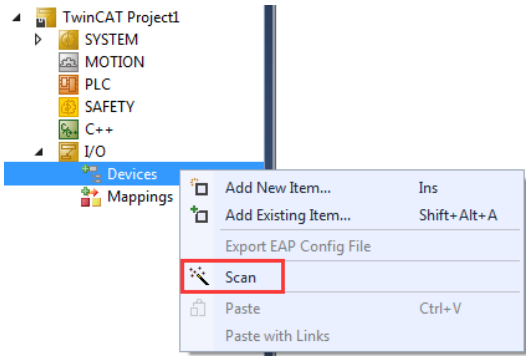
（12）点击工具栏中的蓝色图标把目标控制器切换到 Config Mode，点击确定。



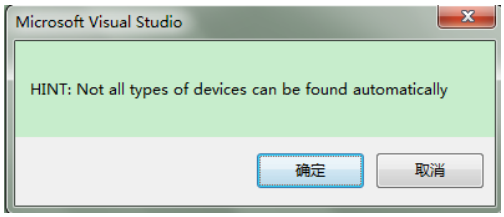
3. 扫描 IO 以及变量连接

2.1 自动扫描

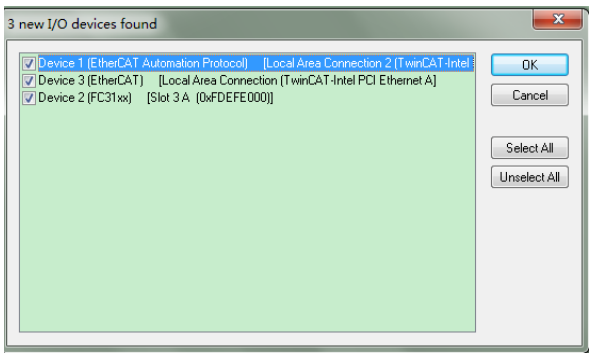
（1）键 Devices 选择 Scan 开始扫描设备和 IO，



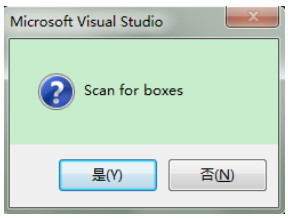
（2）弹出提示框：不是所有设备都能被自动获取，点击确定。

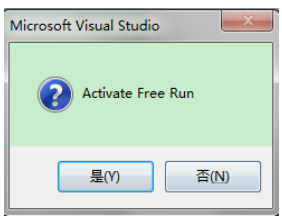


（3）扫描到 IO 设备，选择设备并点击 OK

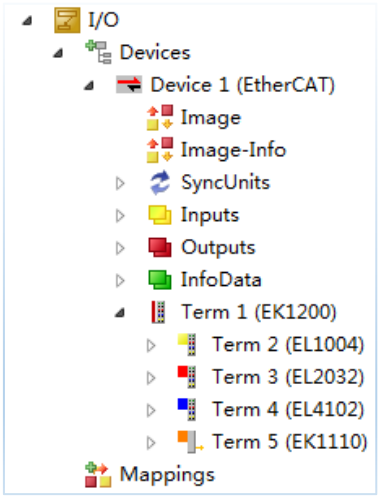


（4）弹出以下对话框，全部点击确定。





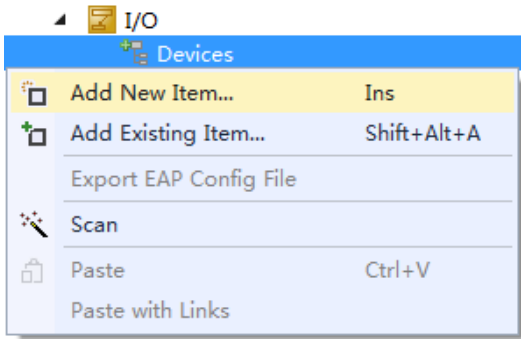
扫描完成后如下：

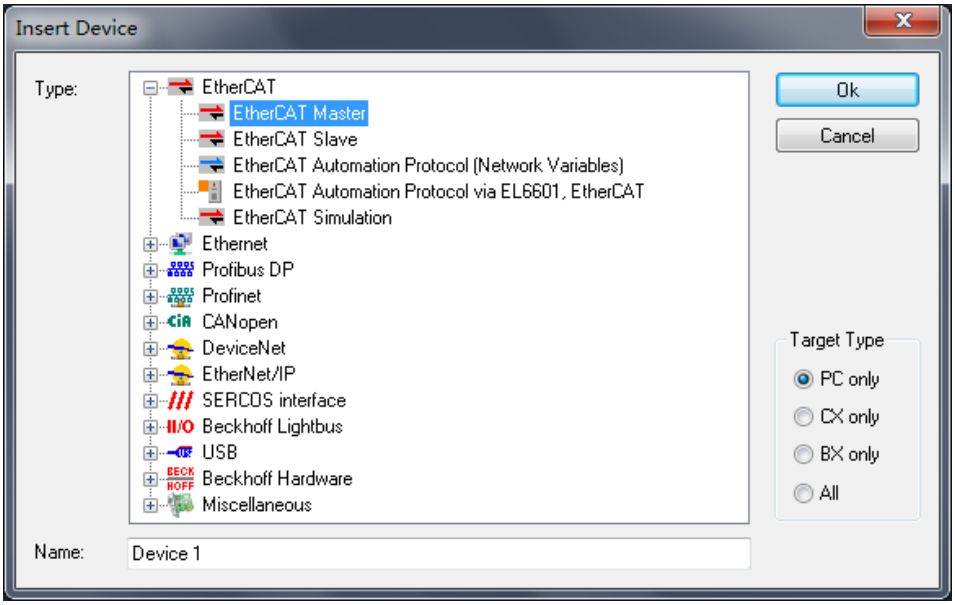


2.2 手动添加

除了自动扫描 I/O 设备外，还可以手动对 I/O 进行配置，即便在没有实际硬件的时候也可以进行前期工程配置。这里以某嵌入式控制器为例，具体步骤参考如下：

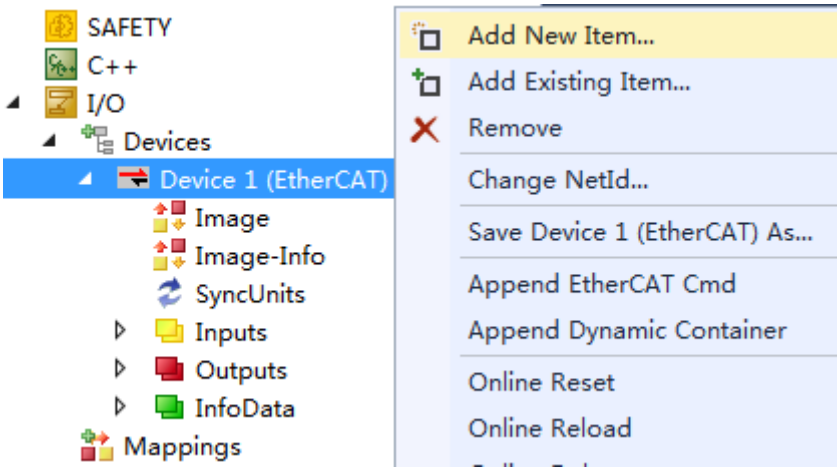
（1）右键 I/O 下的 Devices，选择 Add New Item，在弹出的对话框里选择 EtherCAT->EtherCAT Master

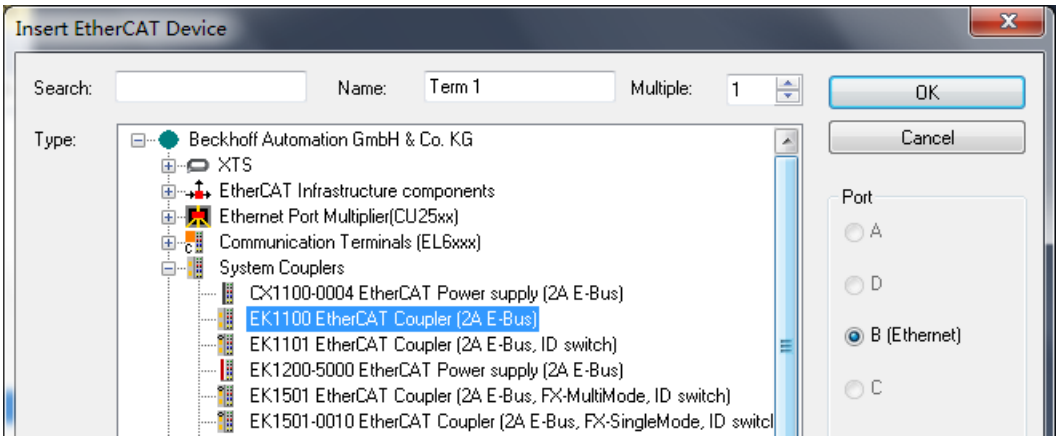




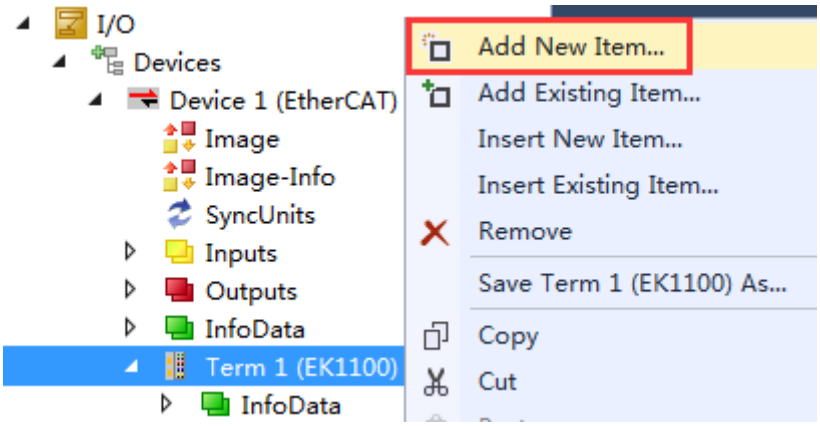
（2）在添加好的 Device1（EtherCAT）上右键，选择 Add New Item，在弹出的对

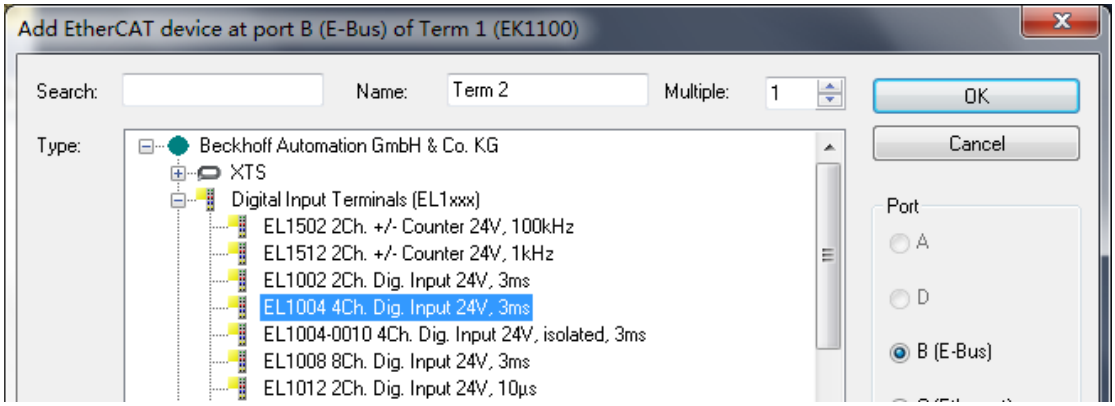
话框里选择 System Couplers->EK1100 EtherCAT Coupler(2A E-Bus)



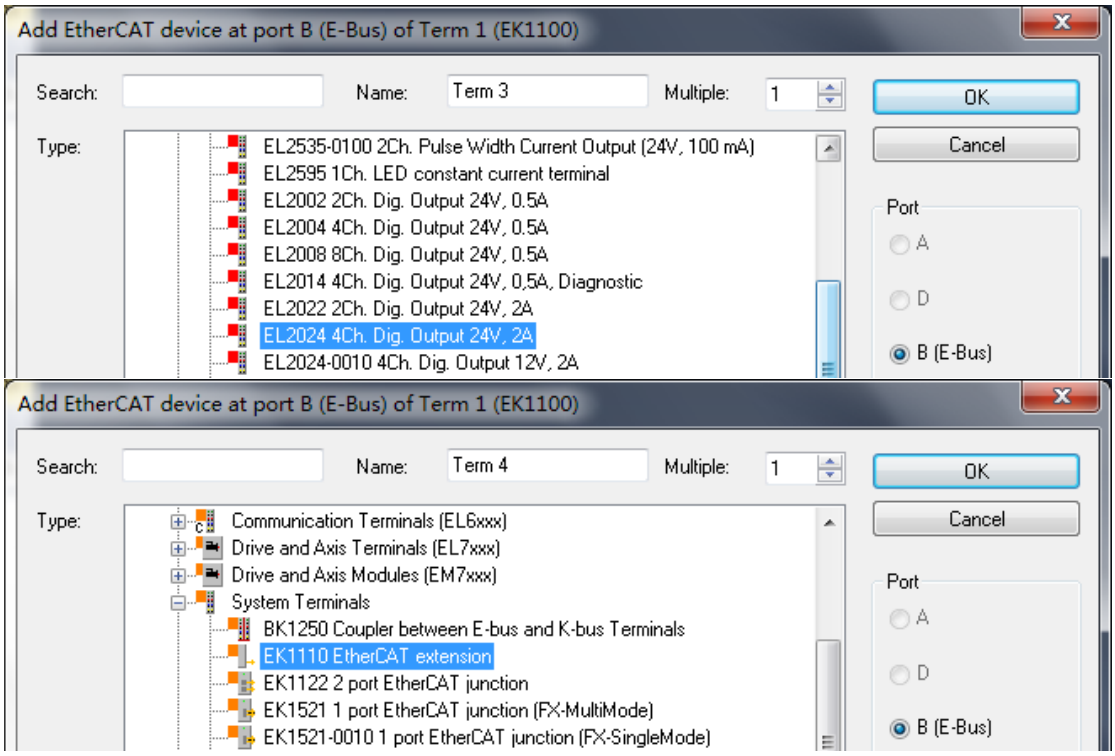


（3）这里我们添加一个输入模块、一个输出模块和一个扩展口。
在添加好的 Term1（EK1100）上右键，选择 Add New Item，在弹出的对话框里选择 Digital Input Terminals(EL1xxx)->EL1004

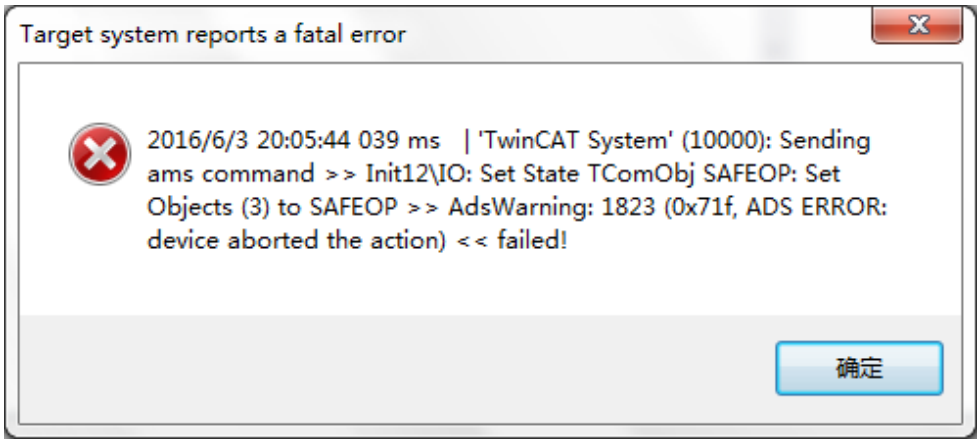




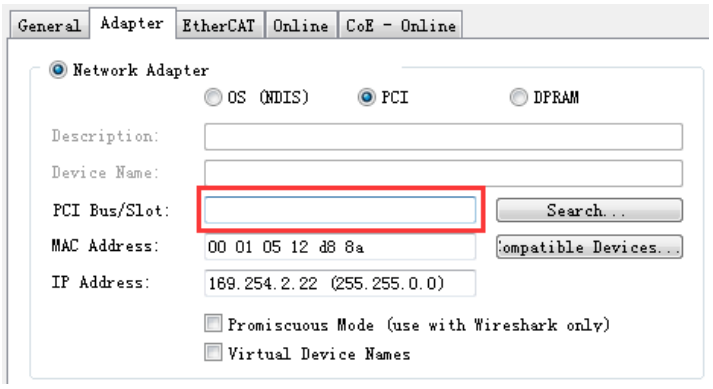
同样的方法添加输出模块 Digital Output Terminals(EL2xxx)->EL2024 和扩展 System Terminals-> EK1110 EtherCAT extension

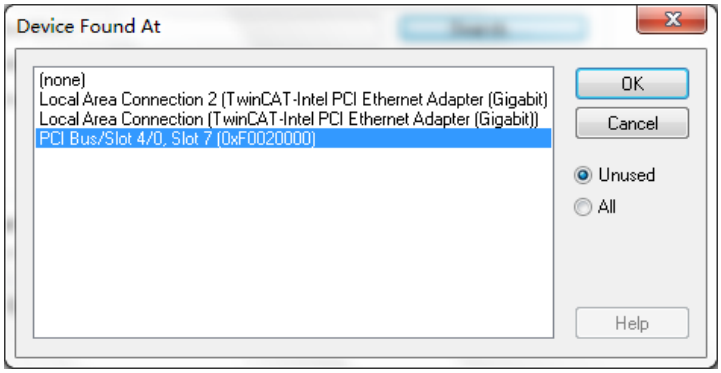


（4）此时如果我们激活配置并切换运行模式是不能成功的，会出现如下报错。这是因为没有配置网口的缘故。

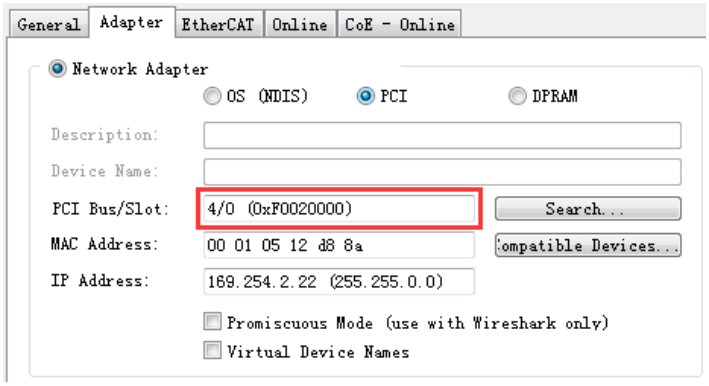


（5）单击 Device1（EtherCAT），选择第二个选项卡 Adapter，可以看到 PCI Bus/Slot 旁边一栏是空白的。点击旁边的 Search，在弹出的对话框里选择 PCI Bus/Slot 4/0,Slot 7 (0xF0020000) 。

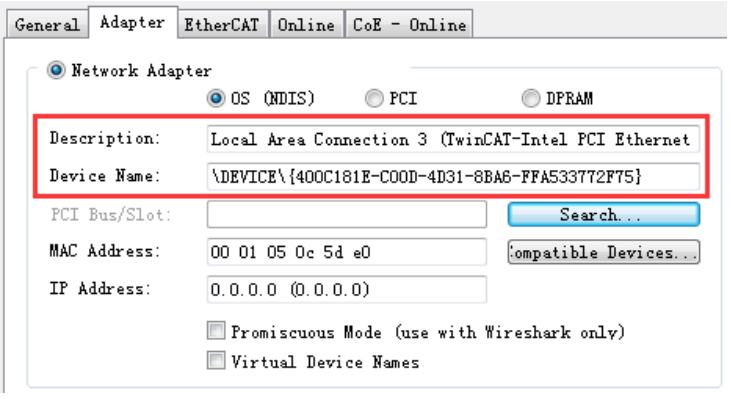




正确的配置如下，现在再激活配置切换运行模式就不会有问题了

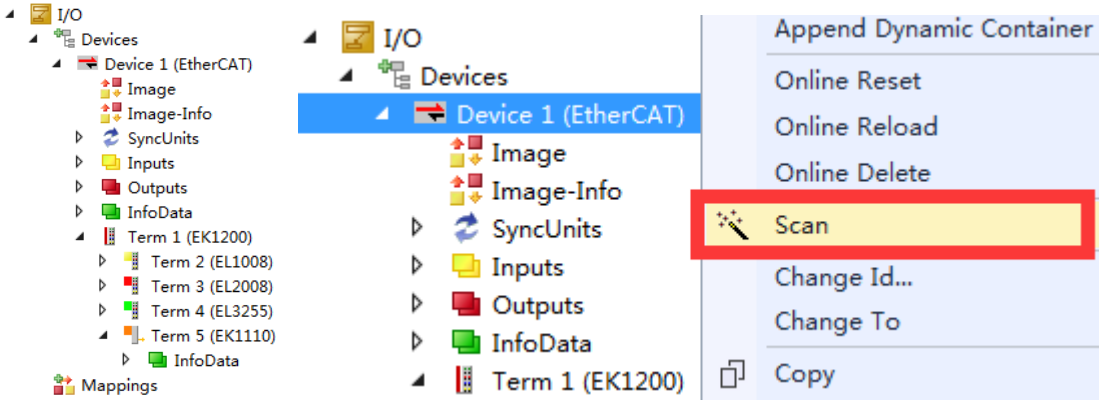


（6）上面是嵌入式控制器的网口配置，工控机的网口配置如下



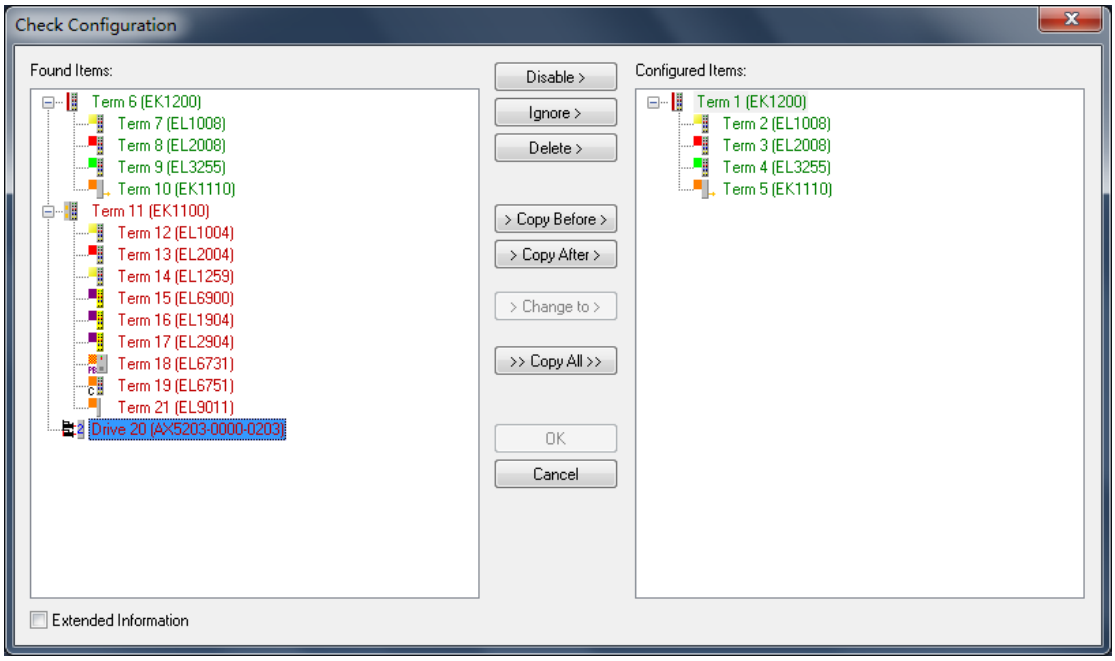
4. 在现有配置下添加新硬件

（1）如果添加了新硬件，我们可以在原有的配置基础上进行修改。现在，在 EK1110 后面再加上一部分硬件。此时，需要右键 Device 1（EtherCAT），选择 Scan，对网络下的设备进行一次新的扫描（此处的 Scan 代表 Scan Boxes）。



（2）跳出弹窗，对当前找到的设备和先前配置好的设备进行了对比。

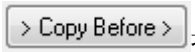
此处绿色的表示和原先 IO 配置中完全一模一样的硬件设备，而红色则是表示之前没有扫描过，或者未曾添加过的设备。



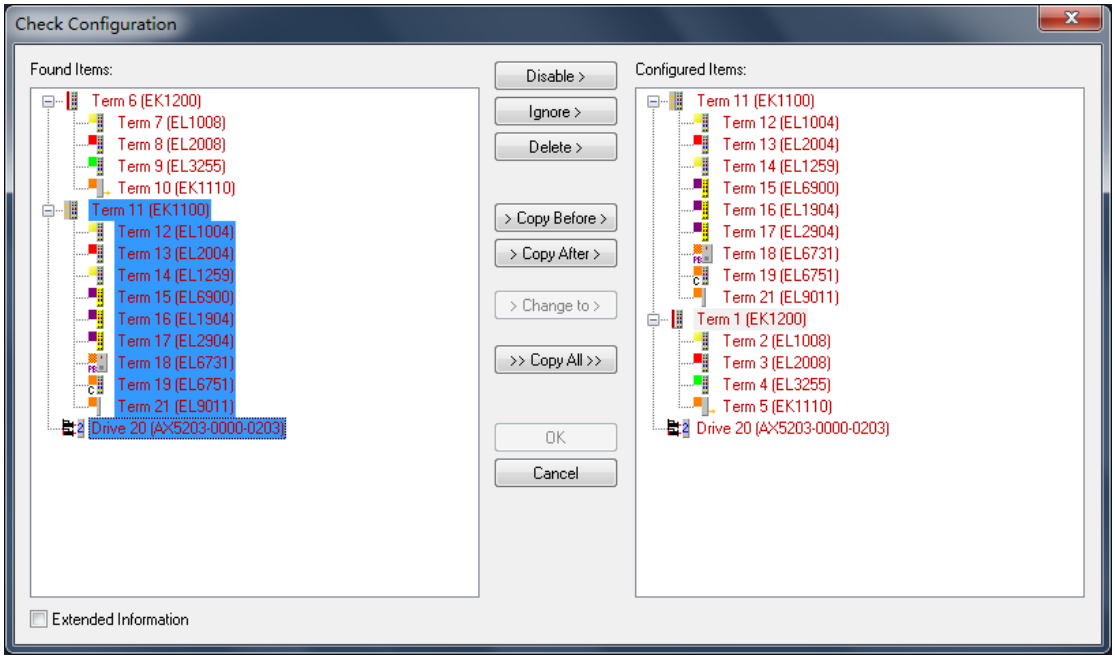
针对扫描窗口这个模块颜色的含义，在 TwinCAT2 的帮助文档中为我们进行了详尽的解释。

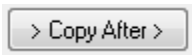
Colour	Explanation
green	This EtherCAT slave matches the entry on the other side. Both type and revision match.
blue	This EtherCAT slave is present on the other side, but in a different revision. This other revision can have other default values for the process data as well as other/additional functions. If the found revision is higher than the configured revision, the slave may be used provided compatibility issues are taken into account. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.
light blue	This EtherCAT slave is ignored ("Ignore" button)
red	• This EtherCAT slave is not present on the other side. • It is present, but in a different revision, which also differs in its properties from the one specified. The compatibility principle then also applies here: if the found revision is higher than the configured revision, use is possible provided compatibility issues are taken into account, since the successor devices should support the functions of the predecessor devices. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.

（3）其中有三个按钮单独进行一下介绍，这些按钮能在保留原有配置及完整信息的情况下把新增的硬件也添加上来。

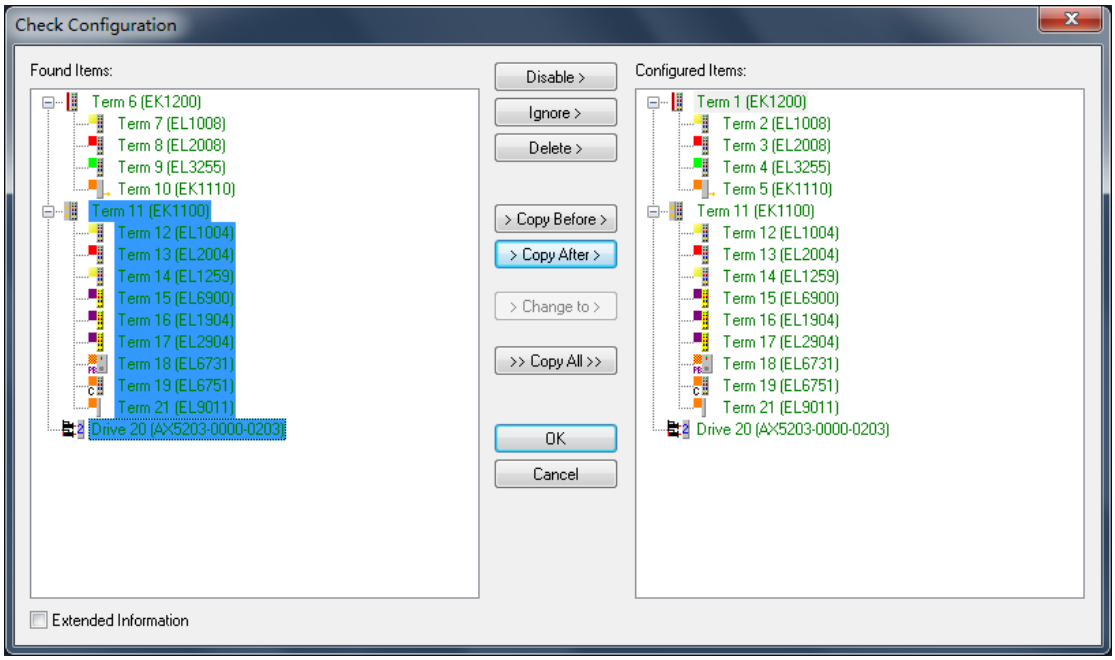


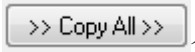
把选中的模块添加到现有项的前面，效果如下，对比发现所有模块位置都发生改变。



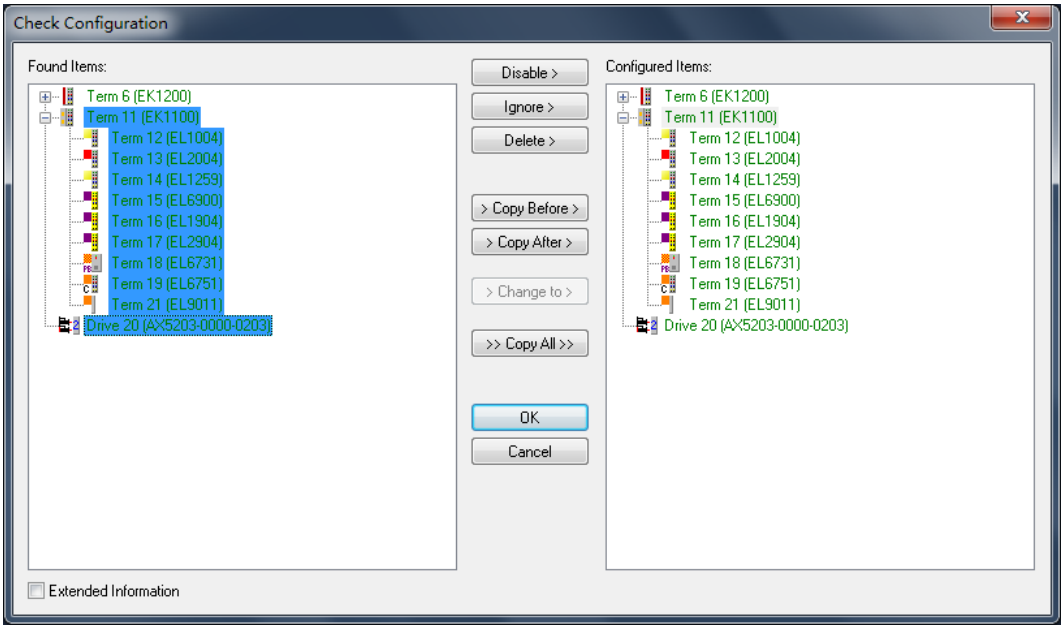


把选中的模块添加到现有项的后面，效果如下。



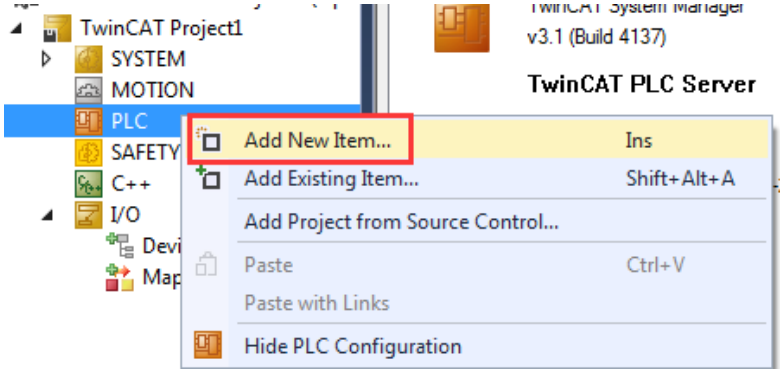


全部复制，不管两边配置是怎么样的，直接把扫到的复制过来。**但是这种方法会使得之前做过的所有变量链接全部清空。**

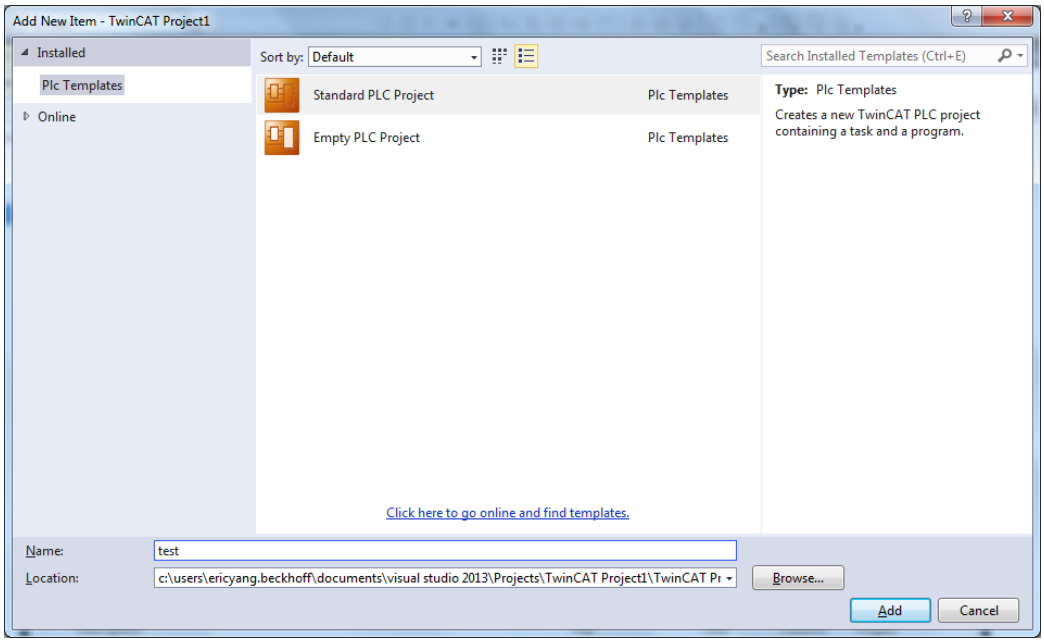


5. 创建变量并链接

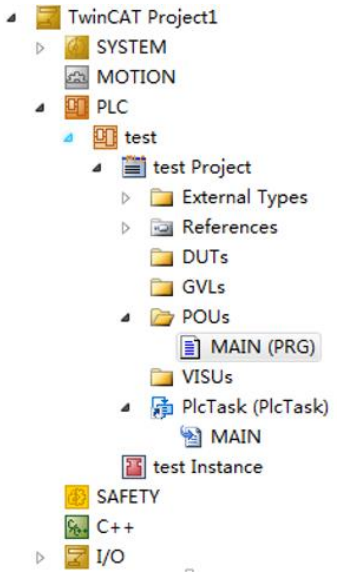
（1）右键 PLC 选择添加新项



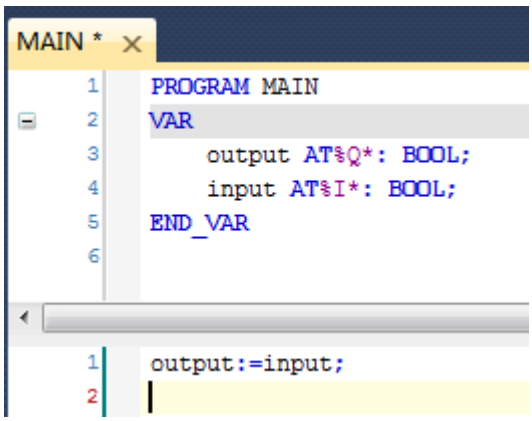
（2）选择 Standard PLC Project，并把名称改成英文，例如下图中的‘test’



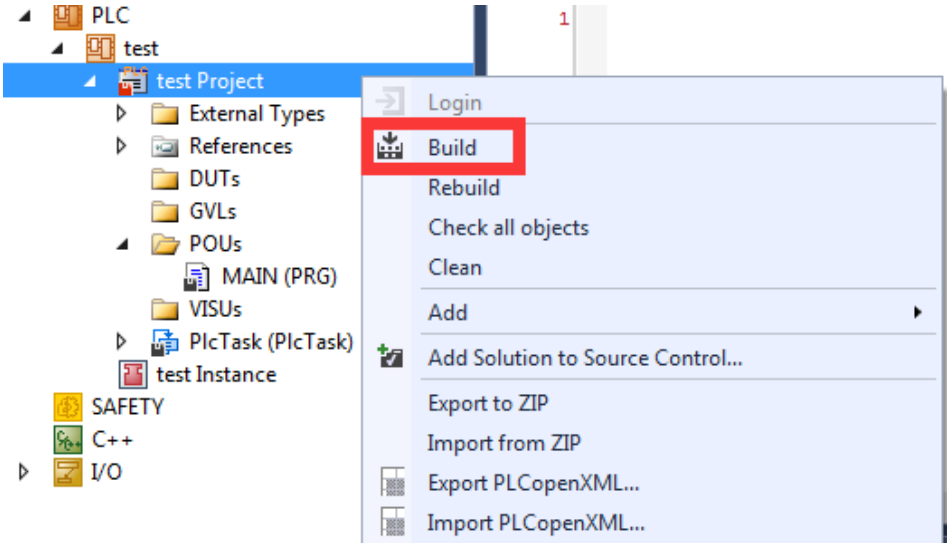
（3）双击 POUs 文件下的 MAIN 开始编程



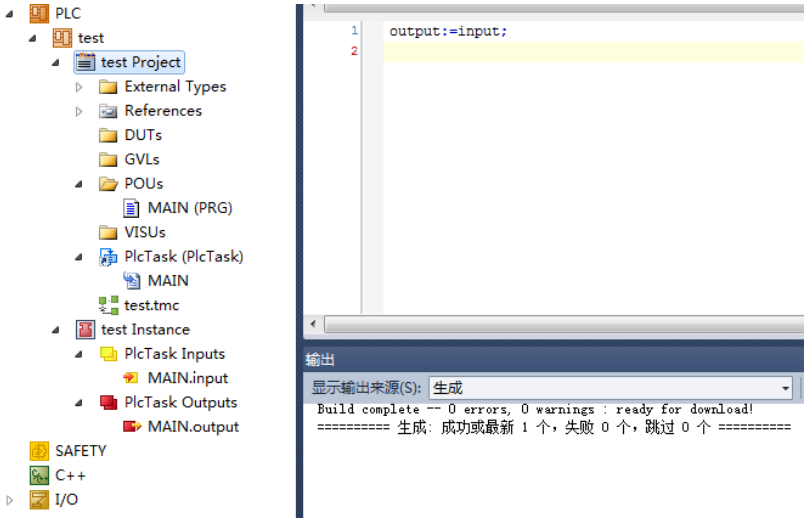
编辑一段简单的程序，输入导通输出



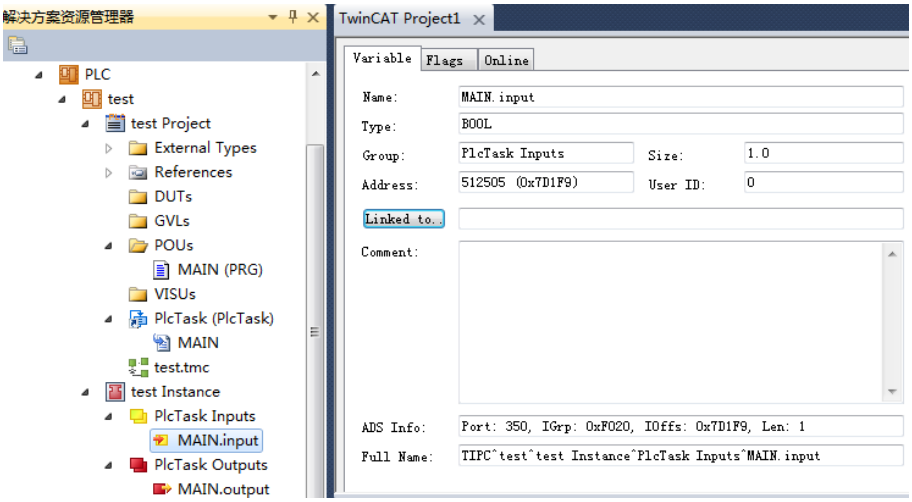
（4）程序写好后右键 test Project，选择 Build 。

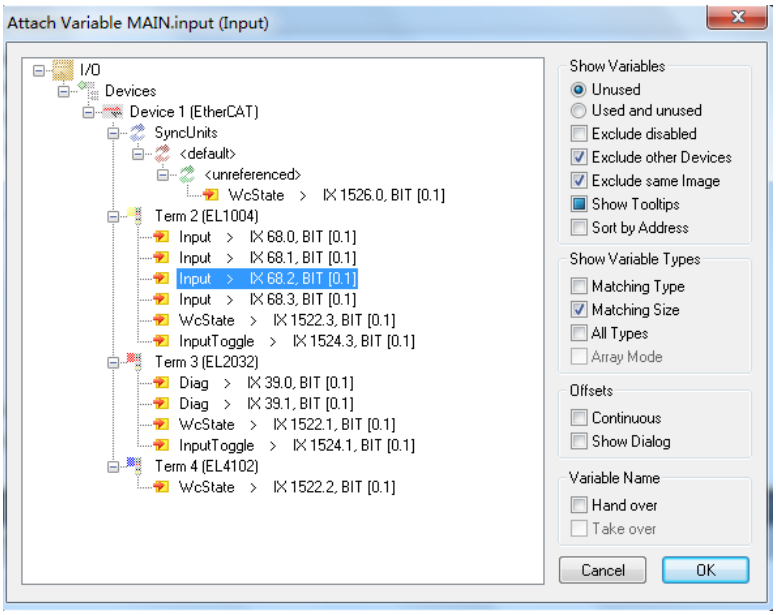


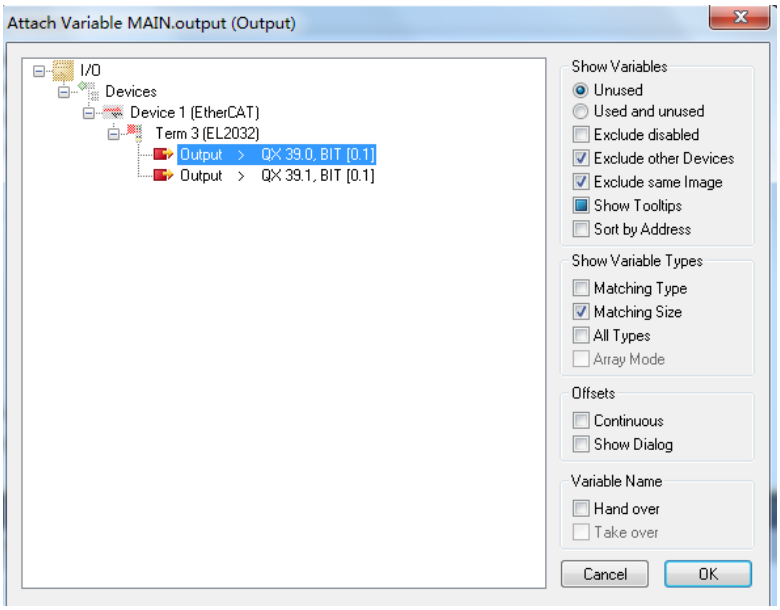
（5）系统会自动编译这段代码，如果没有错误就会在消息栏中提示成功生成，并且在 test Instance 中生成输入输出变量可供连接。



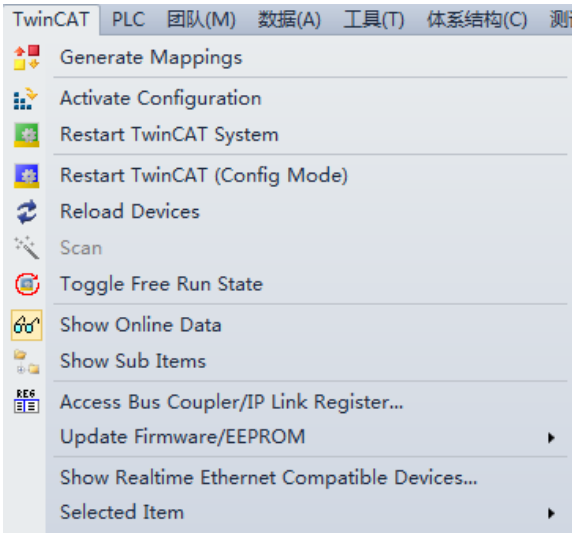
（6）分别点击 test Instance 中的输入输出变量开始进行变量连接。



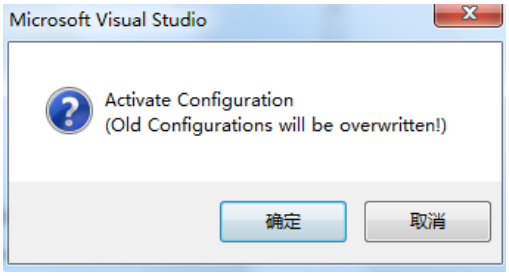




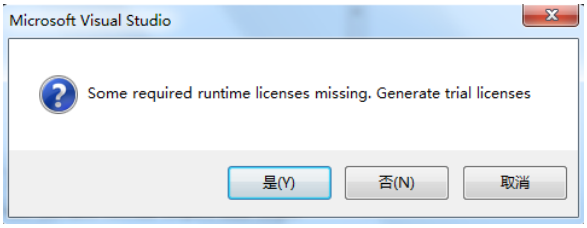
（7）变量连接做好后选择 TwinCAT，点击 Activate Conifguration。



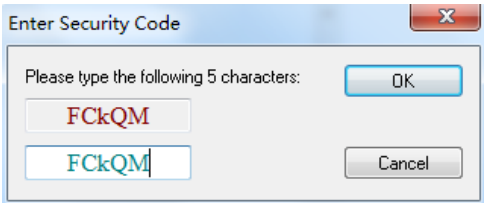
（8）弹出对话框点击确定。



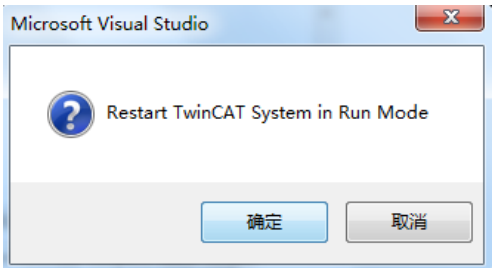
（9）如果弹出以下窗口说明你的项目中有一些 license 没有激活或者漏激活了，不过没有关系，点击是可以重新激活缺少的 license。



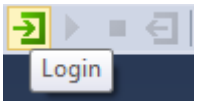
（10）输入 5 位验证码后点 OK。



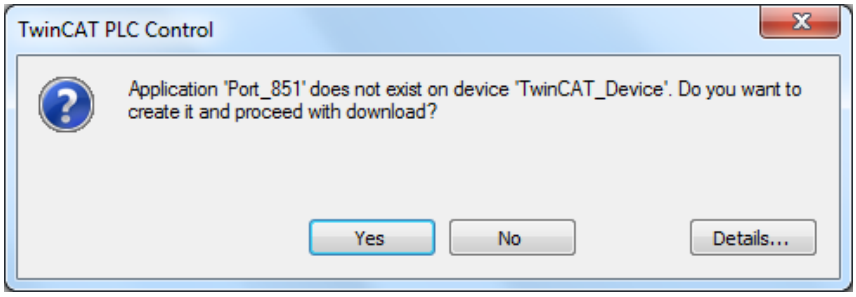
（11）点击确定切换到 RUN 模式。



（12）点击工具栏的绿色箭头 Login。

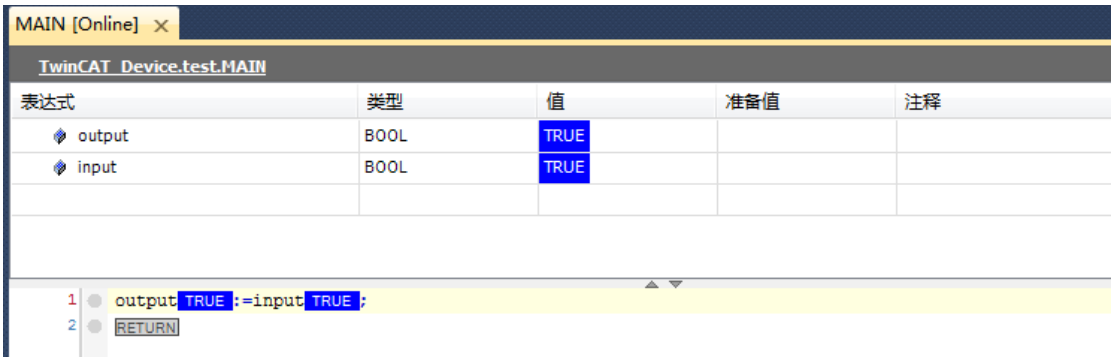


弹出窗口点击是



（13）随后点击工具栏中的绿色箭头 Start 启动程序。

程序正常运行



四、 TwinCAT3 编程语言的 IEC61131-3 标准

1. IEC61131-3 的发展和优势

在开发可编程序控制器（PLC）的早期阶段没有统一的标准。例如德国公司早期喜欢用功能块图和语句表语言，与早期他们将晶体管逻辑电路用来完成控制功能有关系。而美国公司早期喜欢用梯形图和控制鼓（Control Drum），梯形图语言则是从继电器控制逻辑延伸来的。至于法国除了用梯形图语言外还用GRAFCET语言。这一语言特别适用于完成顺序控制的功能。这种编程语言不统一的情况，给用户带来极大的不方便，近几年由于自动化系统的发展，IEC的第七个工作组制定了IEC（International Electro technical Commission）61131-3标准，IEC61131-3本身只作为PLC的编程指导，而不是强制的规则。它是IEC 61131国际标准的第三部分，是第一个为工业自动化控制系统的软件设计提供标准化编程语言的国际标准。

PLCopen国际组织是一个独立于制造商和产品的国际组织，总部位于荷兰。致力于IEC61131标准的推广并取得了很大成功。PLCopen是使PLC软件不依靠于供应商和独立于产品的世界组织。它通过发布和强化IEC 61131-3软件开发标准，给工业控制系统的用户带来很大的价值。

IEC61131-3的优势：

（1）它是国际上承认的标准，统一的结构，语言和处理方式。

（2）它节省你的时间，统一的软件模式和数据类型概念，对来自不同的PLC类型只需学习一次。减少了误解和错误。

（3）对每个问题提供了最佳编程语言规范，一致的5种编程语言规范，不同语言混合编程。

2. IEC61131-3 的内容

（1） IEC61131-3 标准的变量声明、关键字和注释。

IEC61131-3 的变量声明首字符可以是字母或下划线，后面跟数字、字母、下划线、不区分字母的大小写，不可以使用特殊字符、空格、连续的下划线。

IEC61131-3 的关键字会在 TwinCAT3 中以蓝色、大写显示。

例如：

标准的逻辑运算操作：AND，OR，NOT…

标准的数据类型关键字：BOOL，INT，REAL…

程序和功能块的关键字：FUNCTION，FUNCTION_BLOCK，PROGRAM

结构体数据相关的关键字：TYPE，STRUCT

关键字是不可以声明做变量名的。

IEC61131-3 的注释在 TwinCAT3 中是以斜体，绿色的字体显示。

注释是由“(*”开始，由“*)”结束。可以放在行首，行尾，也可以放在程序语句中间，但是不能放在字符串中间。

用于单行并且放在行尾的注释内容，可以使用“//”来进行。如下图所示：

```
(*Table Errors*) 或 //Table Errors
```

（2） IEC61131-3 标准中的数据类型。

在如下的表格中，展示了 IEC 标准中 PLC 的基本数据类型，数据范围和数据大小，声明变量数据类型时可做参考。

数据类型	最小值	最大值	数据大小
BOOL	False	True	1bit
BYTE	0	255	8bit
WORD	0	65535	16bit
DWORD	0	4294967295	32bit
SINT	-127	127	8bit
USINT	0	255	8bit
INT	-32768	32767	16bit
UINT	0	65535	16bit
DINT	-2147483648	2147483647	32bit
UDINT	0	4294967295	32bit
LINT	-2^{63}	$2^{63}-1$	64bit
ULINT	0	$2^{64}-1$	64bit
TIME_OF_DAY	TOD#00:00:00	TOD#23:59:59	32bit
DATE	D#1970-01-01	D#2106-02-07	32bit
DATE_AND_TIME	DT#1970-01-01-00:00:00	DT#2106-02-07-06:28:15	32bit
TIME	T#0s	T#49D17H2M47S295MS	32bit
REAL	$-3.4*10^{38}$	$3.4*10^{38}$	32bit
LREAL	$-1.7*10^{308}$	$1.7*10^{308}$	64bit
STRING	ASCII 码的字符串数据类型，一个字符占一个 BYTE 位。最多可以有 255 个字符。		
WSTRING	Unicode 编码的字符串，一般情况下一个字符占两个 BYTE 位。它本身对字符串长度没有限制。		

在 TwinCAT3 中上述数据显示格式如下表所示：

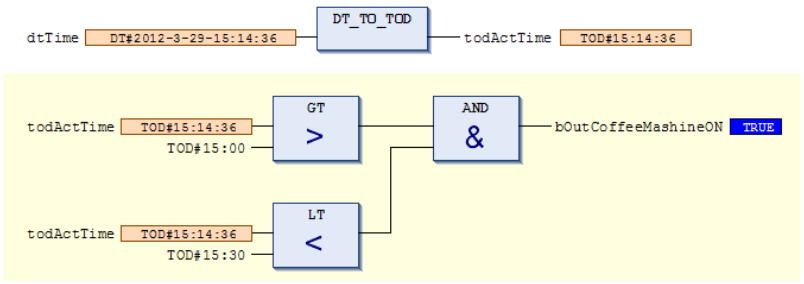
Variable Type		Examples			
BOOL		True	2#1	16#1	1
		False	2#0	16#0	0
WORD		2#1010111111111110		16#AFFE	45054
INT		2#1000000000000000		16#8000	-32768
TIME		T#1h		T#60m	T#3600000ms
D	DAY				
H	HOURS	T#0.5d		T#12h	T#43200000ms
M	MIN				
S	SEC	T#30m18s90ms		T#0.505025h	T#1818090ms
MS	MS				
REAL		0.3333		3.333e-1	

上述是 IEC 编程标准的基本数据类型和表现形式。在 TwinCAT3 中基本数据类

型变量声明方式如下：

```
VAR
    a       :   BOOL;
    b       :   INT;
    c       :   DWORD;
    d       :   DATE;
    e       :   REAL;
END_VAR
```

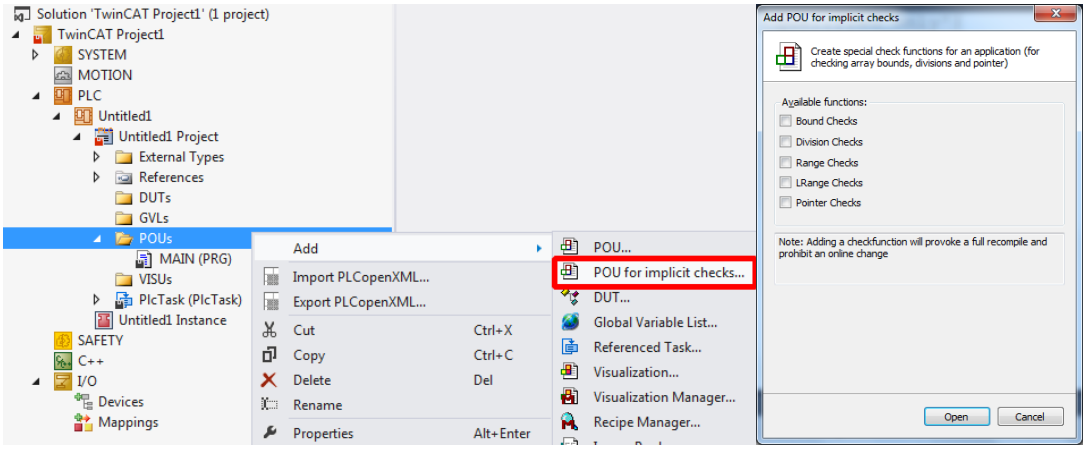
在此标准的中时间数据类型可以进行 IEC 标准操作，例如：比较、逻辑运算等等。如下图所示:



在 TwinCAT3 中通过关键字 ARRAY[0..N] OF “TYPE”对数组进行变量声明。对数组中元素数据类型通过 OF 后的“TYPE”（基本的数据类型）定义，例如:

```
VAR
    a       :   ARRAY[0..10] OF BOOL;
    b       :   ARRAY[0..90] OF INT;
    c       :   ARRAY[0..100] OF DWORD;
    d       :   ARRAY[0..1] OF DATE;
    e       :   ARRAY[0..17] OF REAL;
END_VAR
```

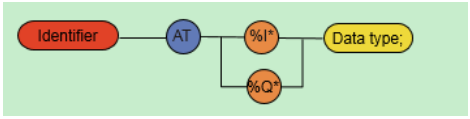
访问数组中的元素通过索引的方式，例如：a[1]：=1；b[2]：=100；等等，使用数组变量编程时，可以使用 TwinCAT3 提供的 Checkbounds 这个功能块来防止访问数组变量时，索引值超出数组长度的情况。功能块的使用非常简单，只要按下述方式添加到程序中即可：



添加完成后，就可以编程了，对数组索引的判断都是自动完成的。编程人员不需要干涉。使用起来很便捷。在 implicit checks 中还有一些 check 功能块，关于相应的功能块的功能和使用方法可以参照，TwinCAT3 帮助文档中的内容。

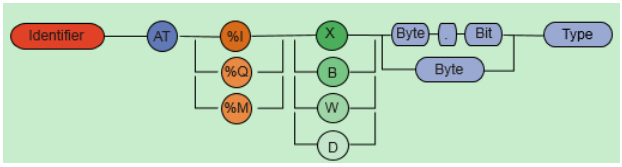
基本的变量名和类型都声明好了，实际工作中程序要求有输入输出变量，来获取，或输出信息。这种变量可以通过“AT%I（Q）*”来进行声明，AT%是关键字，I 表示输入，Q 表示输出。*表示自动分配一个内存地址给这个变量。您也可

以指定一个内存地址给这个变量。下图为输入输出变量声明的实例，




```
VAR
K12531Velocity      AT%Q* : INT;
K12531Ctrl          AT%Q* : BYTE;
K12531ExtCtrl       AT%Q* : WORD;

K12531Position      AT%I* : UINT;
K12531Status        AT%I* : BYTE;
K12531ExtStatus     AT%I* : WORD;
END_VAR
```

如果想要自己分配内存地址给输入输出变量，使用如下的方式：

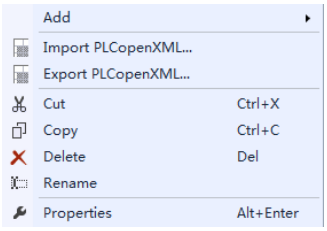


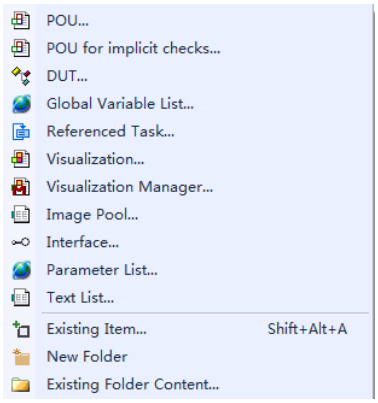
在%I（Q）后接地址存储数据的类型，X 表示 BOOL 型，B 表示 BYTE，W 表示 WORD，D 表示 DWORD。最后加上数据类型 TYPE。例如:

```
VAR
K12531Velocity      AT%QX3.2 :  BOOL;

K12531Status        AT%IB4   :  BYTE;
END_VAR
```

在变量声明区域 VAR 中声明的变量是局部变量。只能在当前程序中使用。对于功能块和函数还有输入输出变量声明区，关键字为 VAR_INPUT，VAR_OUTPUT。在 VAR_IN_OUT 中声明输入输出型变量，这里的变量即是输入也是输出。变量要声明在相应的区域中，局部变量声明在相应功能块或程序中的关键字区域中即可，全局变量在 TwinCAT3 中变量声明的方式如下：

在 GVLs 上右击，然后点击 Add，选择 Global Variable List 在这里个文件的 VAR_GLOBAL 中声明对应的全局变量。

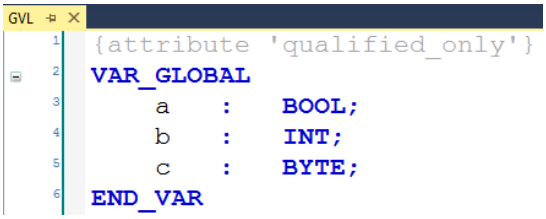




局部变量_

```
VAR
    I AT%I* : BYTE;
END_VAR
VAR_INPUT
    a : BOOL;
END_VAR
VAR_OUTPUT
    b : INT;
END_VAR
VAR_IN_OUT
    c : BYTE;
END_VAR
```

全局变量



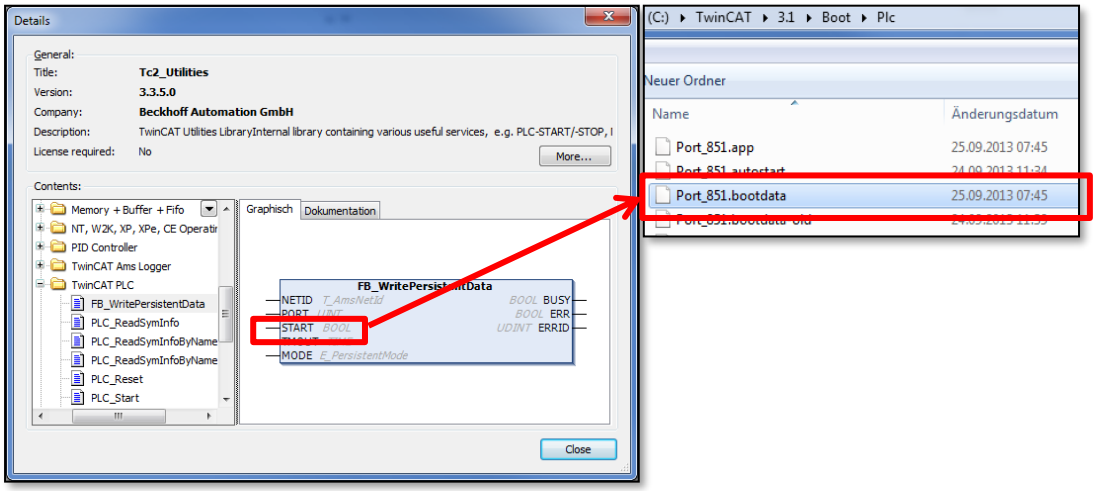
在 TC3.1.4020 版本开始，全局变量的创建都会自动加上属性编译：{attribute 'qualified_only'}
其含义是为了区分全局变量与各局部变量重名的问题，因此现在调用全局变量都需要在变量名前加上全局变量的命名空间，例如：GVL.a，GVL.b，GVL.c，但如果感觉这种方式不习惯或者麻烦，也可以把属性编译行删除，就可以回到以前的直接调用的方式了。

在 TwinCAT3 中的常量和断电保持型数据的声明方式是在变量声明区域关键字后加相应的关键字，常量是 CONSTANT，继电保持型数据是 PERSISTENT。例如:

```
VAR CONSTANT
    PI      :   REAL:=3.141592653;
END_VAR
```

常量在系统初始化完成后，便只能进行读操作，不能进行赋值。

断电保持型变量在 TwinCAT3 中的工作方式是将变量值保存到本地硬盘中，设备重新上电后自动进行读取，所以在断电前需要调用断电保持型变量的写入功能块进行操作。并且注意在调用写掉电保持数据功能块的时候，不要更改变量的值。如下图，写掉电保持数据功能块为 FB_WritePersistentData，在 Tc2_Utilities.LIB 这个库文件中。



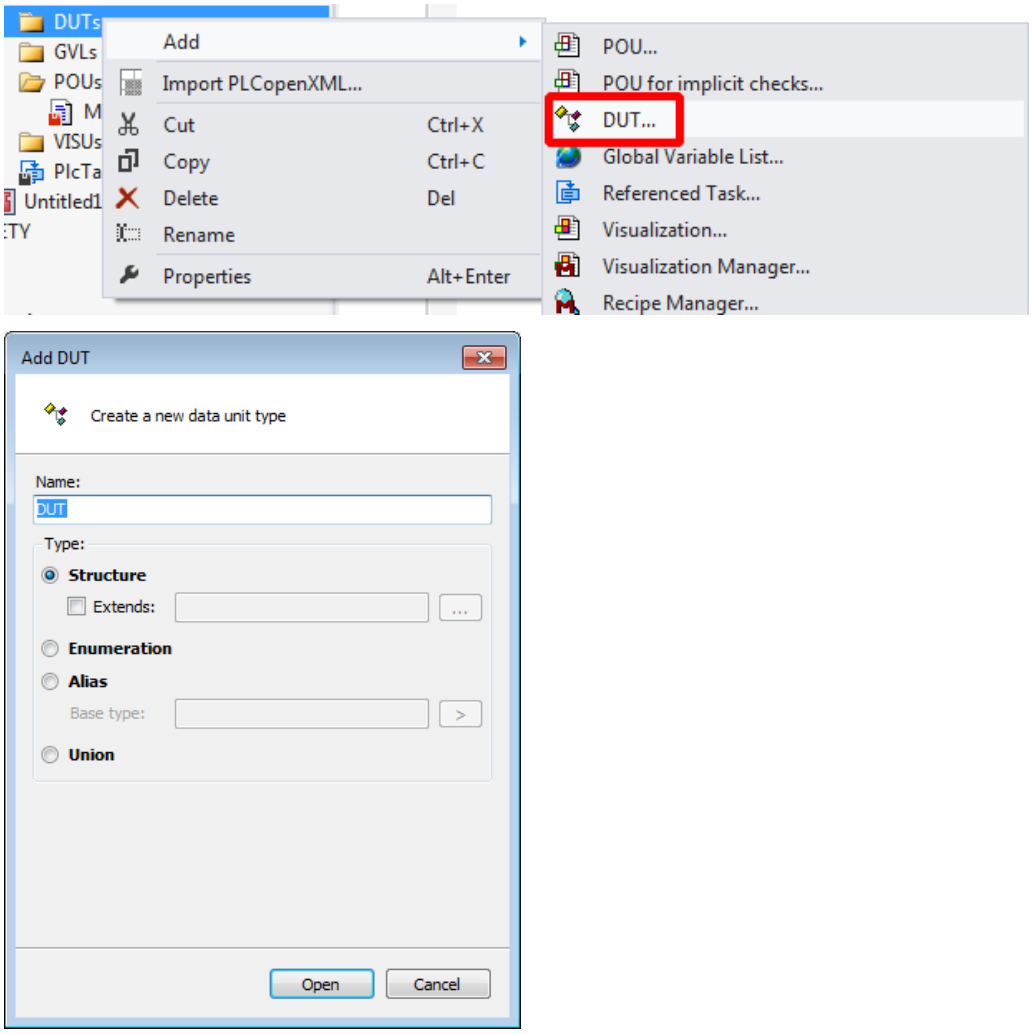
当然也可以把断电保持变量定义在 VAR_PERSISTENT 作用域中。

```
VAR PERSISTENT
    iount: INT;
END_VAR
```

不过定义 PERSISTENT 类型变量需要配合 UPS 才可以保存变量，因为此类变量只有当 PLC 停止的时候才会进行保存，不然断电重启还是会丢失。

（3） IEC61131-3 标准中的其它数据类型。

在 IEC 标准中可以通过关键字 TYPE…END_TYPE。来声明结构体数据，共用体数据和一些自定义数据。这些数据类型声明步骤为：
在 DUTs 上右击 DUTs，然后点击 Add，然后选择 DUT 出现 Creat type 对话框。选择相应的数据结构进行编程。



在 TwinCAT3 中支持共用体（Union）类型数据，共用体类型的数据表示，共用体中的多个数据共用一段内存地址。例如：

```
TYPE unionStringArray :
UNION
    sVar : STRING;
    aVar : ARRAY[0..80] OF BYTE;
END_UNION
END_TYPE
```

共同体数据在上图上声明后，sVar 和 aVar 变量共用一段内存地址。那么在使用过程中，这段地址中的字符串，可以以字符串方式访问又可以字符数组方式访问，简化了对数据的处理方式。并且节省内存。

枚举变量（Enumeration）是多个有固定数值的一类变量的集合。枚举变量的声明方式如下：

```
TYPE Color :
(
    Yellow := 0,
    Red := 3,
    Green := 5
);
END_TYPE
```

有了枚举变量，在程序中特定的地方使用，就会使程序变得清晰易读。例如在程序中有多个输入，分别是几个固定值。那么将其声明为枚举变量后，就可以

使用助记词帮助了解这个数字的用处。在程序看到助记词比看数字清楚很多。且在数据修改时，只对声明枚举变量处改动即可，对于多处使用到的数据优势明显。

同时如果有 2 个相同的枚举值在 2 个不同的枚举体中是允许被分别访问的。

```
Example:
TYPE Woche:(Mo, Di, Mi, Dn, Fr, Sa, So:=10);(*Mo = 0 Di = 1..
                                              .. Sa = 6 So = 10*)
END_TYPE

TYPE Richtung:(Up, Dn);(*Up = 0 Dn = 1*)
END_TYPE
```

就可以通过 Woche.Dn 和 Richtung.Dn 进行分别访问

结构体数据（Struct）声明方式如下：

```
TYPE STRUCT1 :
STRUCT
    input       : BOOL;
    invalue     : INT;
    outvalue    : DINT;
END_STRUCT
END_TYPE
```

对于同一组的变量，可以用结构体数据类型进行定义，每个元素的数据类型可以不同，结构体可以做成需要的数据类型，并且结构体数据可以方便进行数据的修改，以方便日后的使用。

在 TwinCAT3 中通过 alias type 在基本的数据类型上定义自己的数据类型，例如：在 STRING 类型上，可以定义固定长度的数据类型 STRING（50），方便编程和调试。

（4） IEC61131-3 标准中程序单元组成。

程序单元由程序（PROGRAM），功能块（FUNCTION BLOCK），函数（FUNCTION）组成，三种单元主要特点如下：

程序：

由任务调用（一个 Program 能够调用另一个 Program）

可以调用: Function Blocks，Functions，Programs。

局部变量: 静态，即局部变量的数据可以在下个周期使用。

输入: 一般是没有输入的变量，在 VAR_INPUT 可以定义输入变量的。

输出: 一般是没有输出的变量，在 VAR_OUTPUT 可以定义输出变量的。

输入输出型变量：可以在 VAR_IN_OUT 定义的。

监视: 在线状态下变量的值是实时可见的。

功能块：

由 Programs 或者其它的 Function Blocks 调用

可以调用: Function Blocks，Functions。

局部变量: 静态，即局部变量数据可以在下个运行周期使用，多个功能块中，每个 FB 都有自己的局部变量。

输入: 多个输入，在 VAR_INPUT 中定义。

输出: 多个输出，在 VAR_OUTPUT 中定义。

输入输出:多个输入输出变量。在 VAR_IN_OUT 中定义。

监视:在线状态下对于每个指定的功能块，局部变量是可见的。

函数：

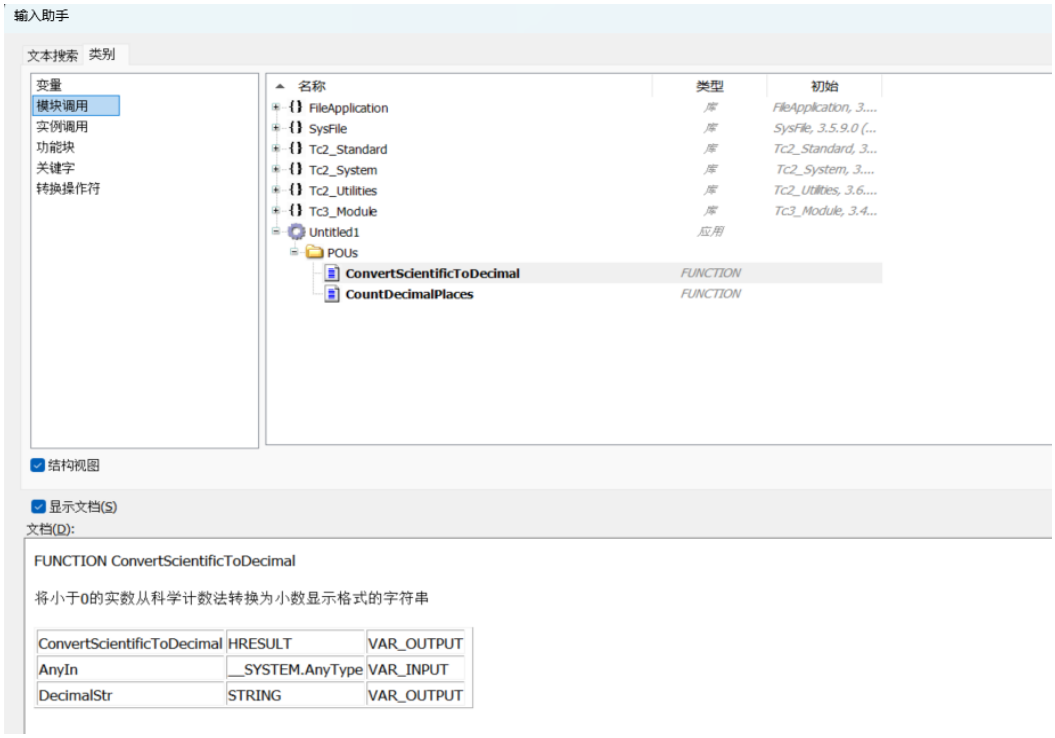
由 Programs, Function blocks 和其它的 Functions 调用

可以调用: Functions

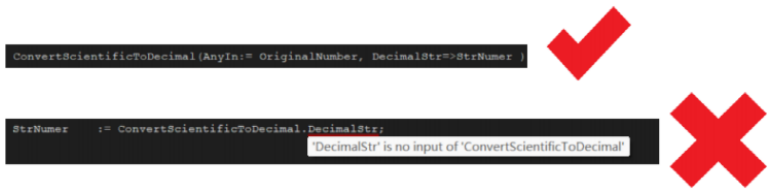
局部变量: 临时的, 即局部变量数据仅在当前 function 执行时可以使用，之后这个数据区就被用于其它 functions。
输入: 多个输入变量，在 VAR_INPUT 中定义。
输出: 多个输出变量，在 VAR_OUTPUT 中定义。
返回值: 仅有一个！
输入输出:可以有多个输入输出型变量，在 VAR_IN_OUT 中定义。
监视: 在线状态下仅能看到 “???”。使用断点可以监视其局部变量值。

函数与功能块的区别：

1、函数无需声明，可直接在程序编写区通过输入助手，进行模块调用；



2、函数和功能块不同，函数只能将模块调用后生成的输出接口关联到变量，而不能将函数的输出通过空间名索引的方式赋值给变量(详见下图)，并且函数在运行完成后就会释放内存。



3、函数多输出推荐的使用方法：可以将返回值设置为BOOL，或者系统变量HRESULT。返回值用于判断函数是否执行成功，并且可以正常关联到程序变量，如下图所示：

```
IF  ConvertScientificToDecimal(AnyIn:= OriginalNumber, DecimalStr=>StrNumer ) = S_OK THEN   // 转换格式

    DONE    := I_File.PutStrings(bExcute,StrNumer); // 写入文件
ELSE
    Error       := TRUE;
END_IF
```

（5） IEC61131-3 标准 ST 语言中的条件选择，循环和函数调用。

IF 条件语句：

```
IF （布尔表达式） THEN
    （执行语句）
ELSIF （布尔表达式） THEN
    （执行语句）
ELSE
    （执行语句）
END_IF
```

IF 条件语句，在布尔表达式的值为 TRUE 时执行（执行语句）。布尔表达式可以是布尔类型的变量（bVar..），可以是比较判断的值（a>b），功能块的值的判断（LEFT(STR:= strVar, SIZE:=7) = 'TwinCAT'）等等。

CASE 选择语句：

```
CASE 选择条件 OF
1:      执行语句 1
2, 4, 6:  执行语句 2
7..10:    执行语句 3
...
ELSE
   Default
   执行语句 4
END_CASE;
```

根据选择条件中的值，执行相应执行语句。

FOR 循环语句：

```
FOR   i:=1 TO 12 BY 2 DO
      Feld:=i*2
END_FOR
```

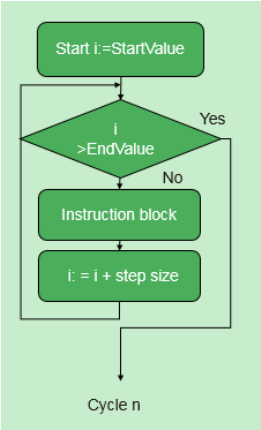
FOR 循环的工作流程如下：

1.首先赋值 i 然后

2.执行后判断 i 是否达到关键字 TO 后的状态，达到则退出，

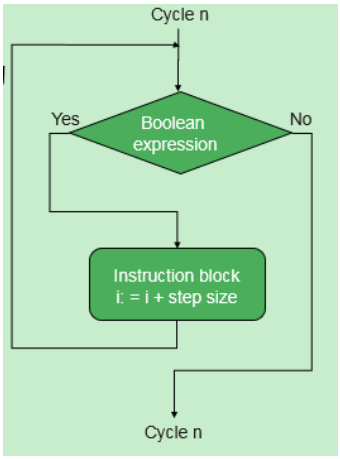
3.未达到执行循环内的语句

4. i 增加关键字 BY 后的数值并再次执行 2-4.



WHILE 循环语句：

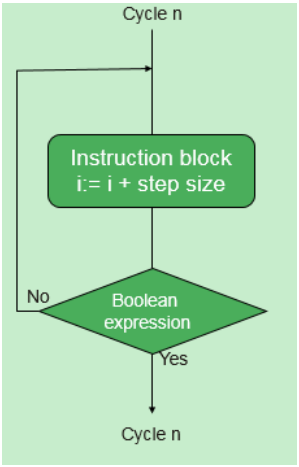
```
i:=0;
WHILE i<100 DO
   Feld:=i*2;
   i:=i+1;
END_WHILE
```



WHILE 循环的工作流程为：
若关键字
WHILE 后面的布尔表达示的值为 TRUE。则一直执行循环.

REPEAT 循环语句：

```
i:=0;
REPEAT
     Feld:=i*2;
   i:=i+1;
   UNTIL i>100
END_REPEAT
```



REPEAT 循环，在布尔表达式的值是 FALSE 时一直执行循环，在布尔表达式的值变为 TRUE 时退出循环。程序至少被执行一次。

ST 语言中功能块的调用：

首先在变量声明区域，声明一个功能块实例。然后在程序编写区域按键盘的 F2 键选择声明好的实例，会自动调出相应的功能块。将所需参数填写入相应位置，就完成了功能块的调用。还可以直接在程序编写区域，按键盘 F2 按键，找到相应的功能块，之后会弹出实例化对话框，填写实例名称，选择声明的区域然后确定完成调用。下面是调用功能块的例子:

```
TON1(IN:= , PT:= , Q=> , ET=> );
```

TON1 便是功能块 TON 的实例，（）中的内容是输入输出参数，根据自己的需要进行读写。对功能块中的单独项可以通过下面方式访问：

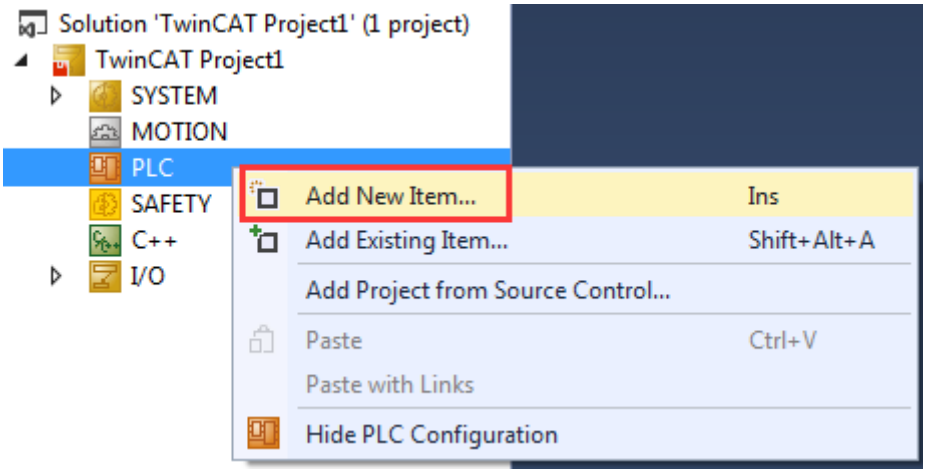
```
TON1.IN:=BON;
TON1.PT:=T#10MS;
out:=TON1.Q;
```

以上内容为 IEC61131-3 标准中基础内容介绍，对于不理解和更深入的用法可以参照 TwinCAT3 的 information system 手册。

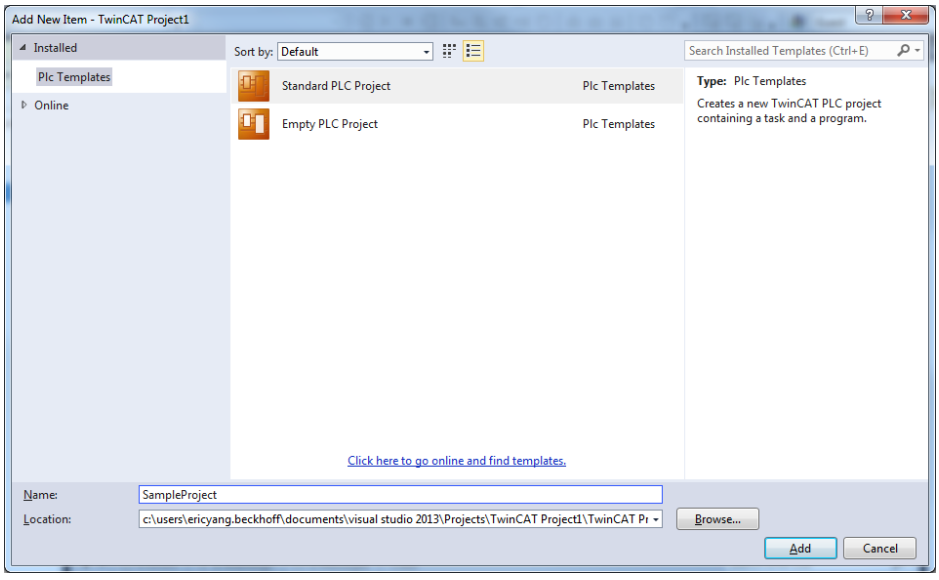
五、 TwinCAT3 PLC 简单程序编写与调试

1. PLC 简单程序编写

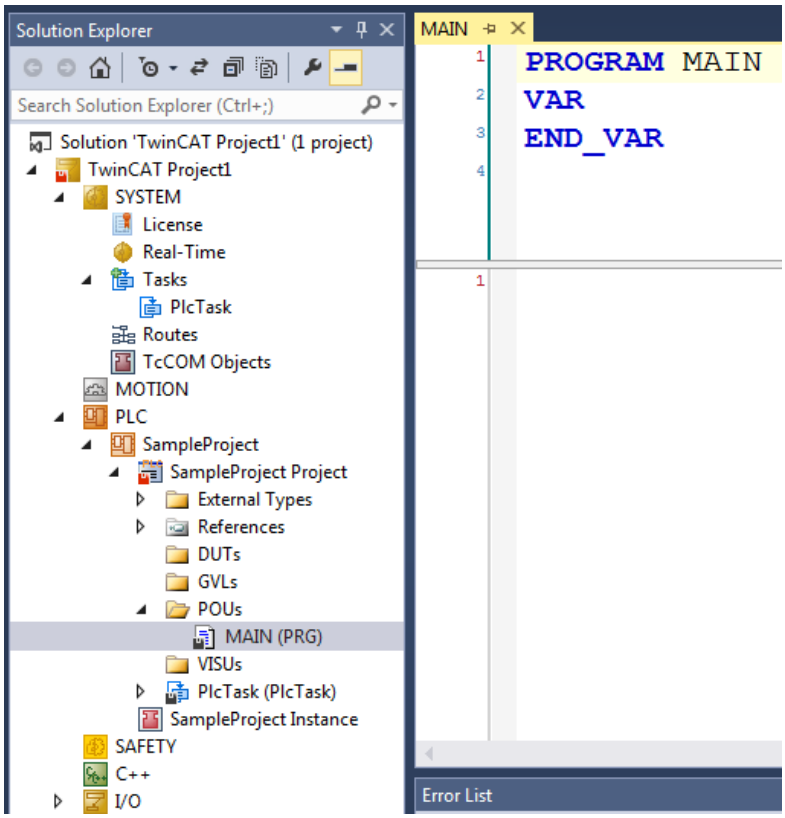
（1）新建 TwinCAT3 项目，右键 PLC 创建新 PLC 项目



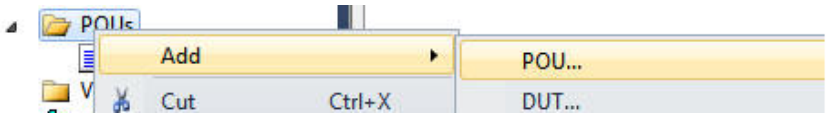
（2）弹出窗口，输入 PLC 项目名（不允许中文名），点击 Add 进行添加



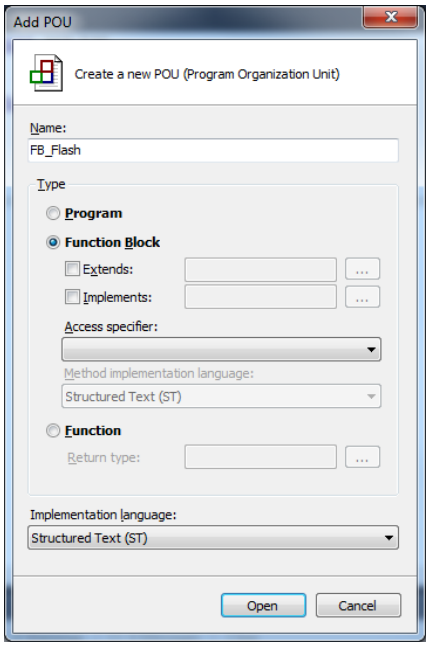
（3）创建好 PLC 项目后可以发现 standard 中已经包含了一个用 ST 语言编写的 MAIN 空程序以及一个 PlcTask



（4）右键 POUs→Add→POU



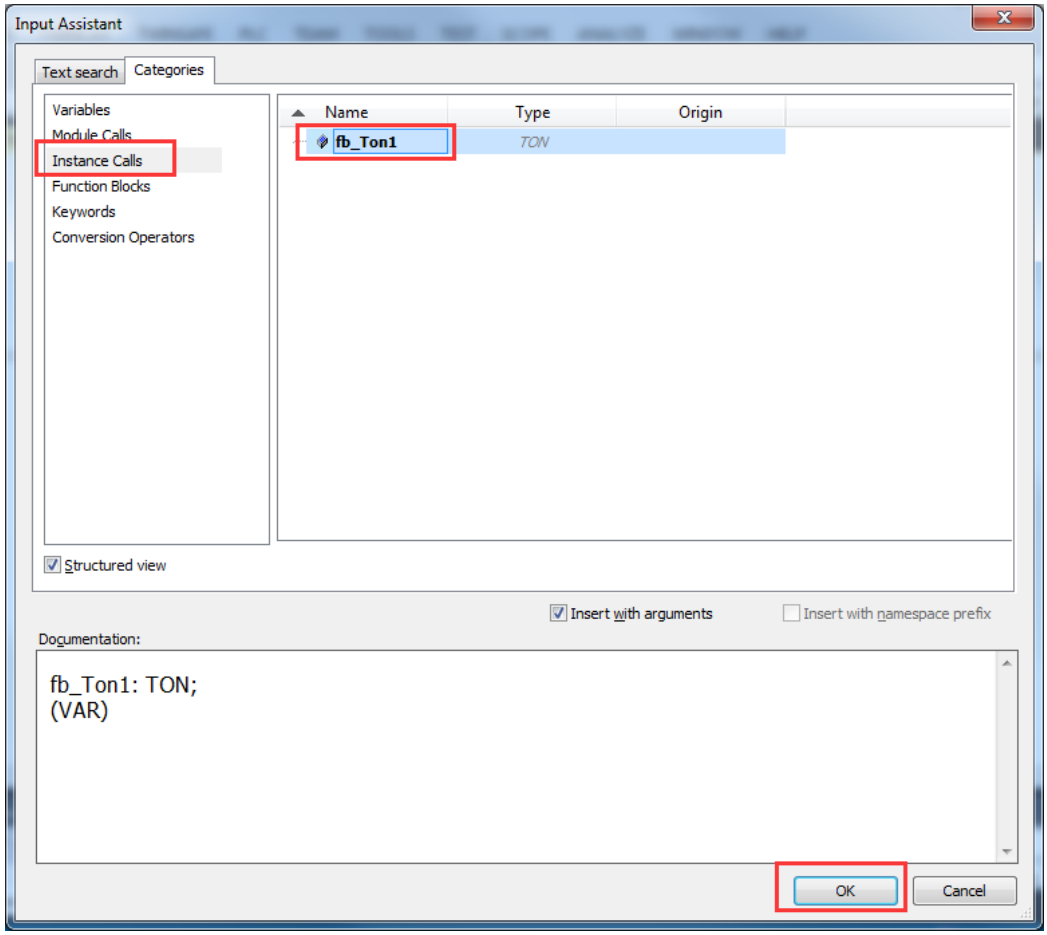
（5）把 POU 类型改成 Function Block，修改名字为 FB_Flash，语言还是用默认的 ST 后点击 Open



（6）在此功能块中创建一些输入输出变量，以及声明一个定时器功能块 TON

```
FUNCTION_BLOCK FB_Flash
VAR_INPUT
    tCycletime: TIME;
END_VAR
VAR_OUTPUT
    bQ: BOOL;
END_VAR
VAR
    fb_Ton1: TON;
END_VAR
```

（7）随后在此功能块中实现一个简单的闪灯功能，首先调用声明好的定时器功能块 fb_Ton1，在功能块主程序窗口按 F2 快捷键，选中 Instance Calls 中的 fb_Ton1 后点击右下角的 OK 进行调用



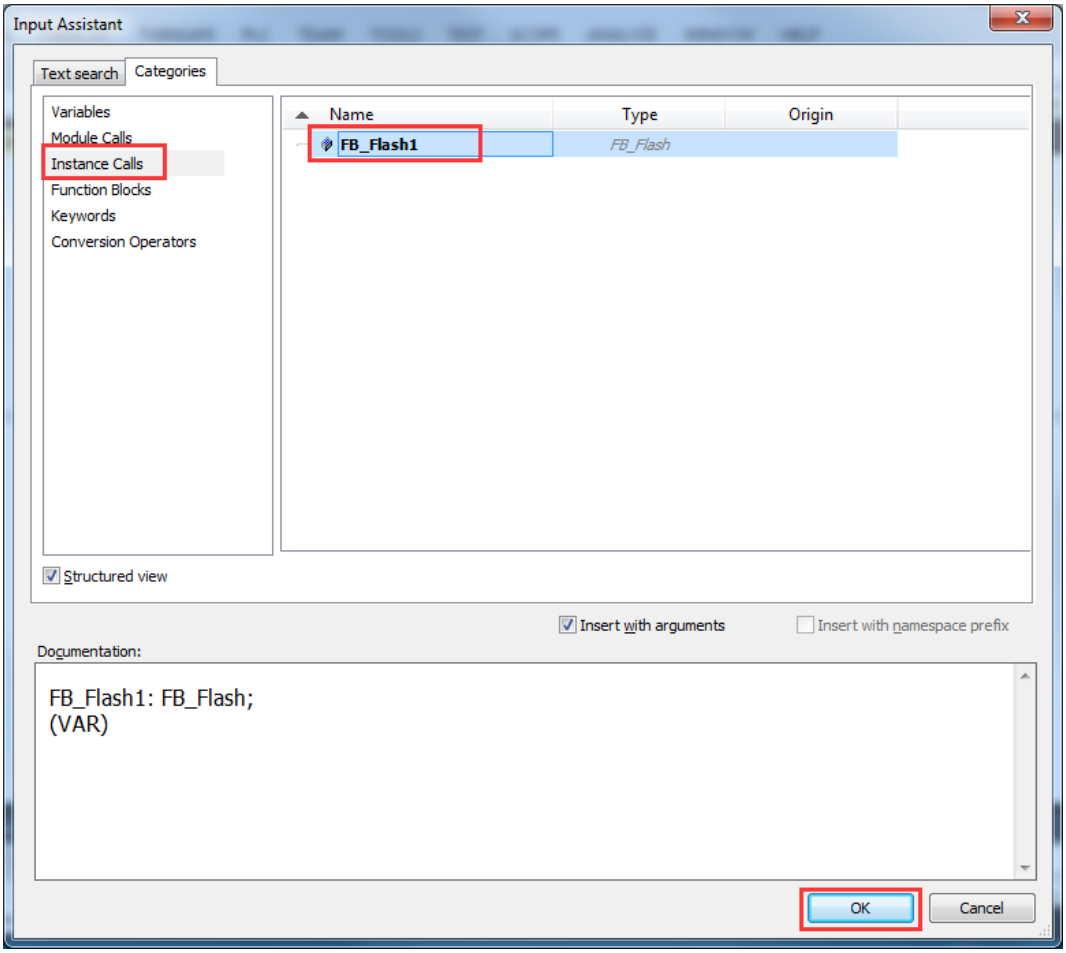
（8）最后在程序中编写一下这段程序，这样就完成了简单闪灯功能块的创建

```
fb_Ton1(IN:= NOT fb_Ton1.Q  , PT:=tCycletime , Q=> , ET=> );
IF fb_Ton1.Q = TRUE THEN
    bQ := NOT bQ;
END_IF
```

（9）回到主程序 MAIN，编写程序调用所创建的功能块 FB_Flash，在声明窗口先对功能块进行变量声明，并且创建若干变量以便后面程序需要用到

```
PROGRAM MAIN
VAR
    FB_Flash1: FB_Flash;
    t1: TIME:=T#3S;
    bDout1 :BOOL;
    A: INT;
    B: INT;
END_VAR
```

（10）随后在主程序窗口按 F2，选中 Instance Calls 中的 FB_Flash1 后点击右下角的 OK 进行调用



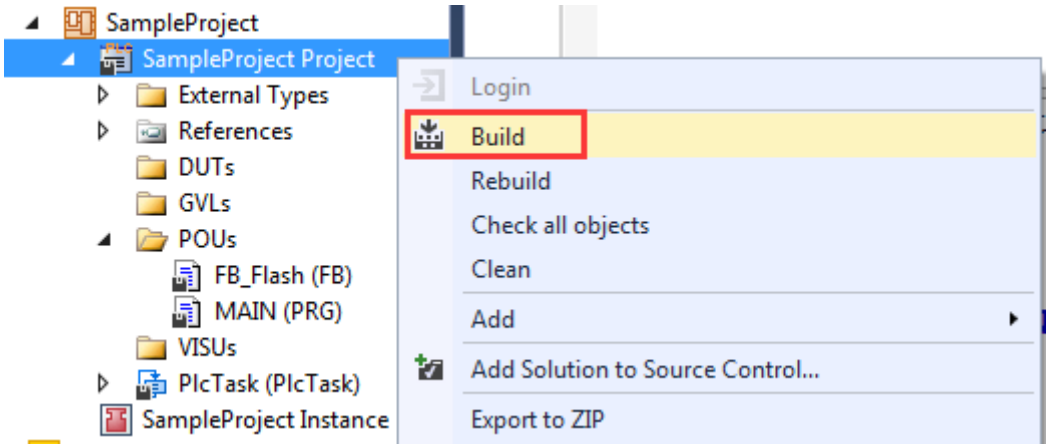
（11）分别把 t1 和 bDout1 与功能块输入输出对应

```
FB_Flash1(tCycletime:=t1 , bQ=>bDout1 );
```

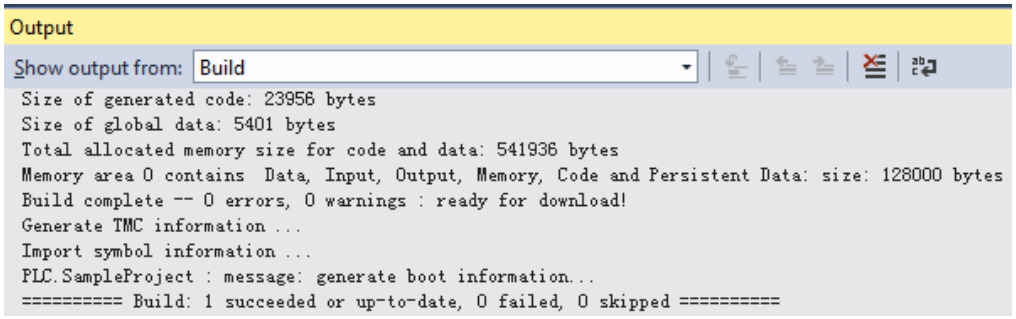
（12）紧接着在 MAIN 中编写一段简单的 IF 语句

```
IF A=0 THEN
    B:=1;
ELSIF A=1000 THEN
    B:=-1;
END_IF
A:=A+B;
```

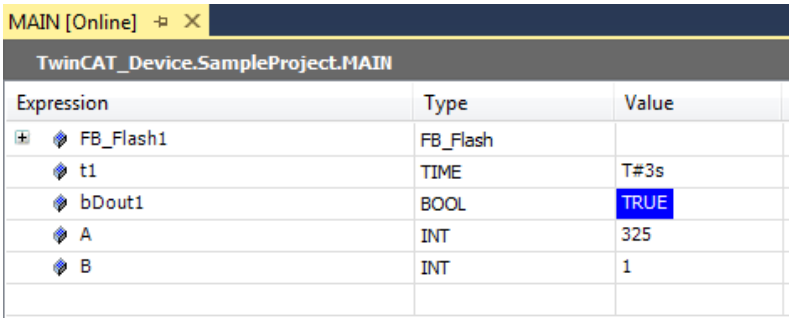
（13）程序写完后右键 PLC 项目 build 进行编译



（14）输出窗口没有报错后就可以把项目激活运行了



（15）active configuration 后选择 login，并且把 PLC 运行起来后可以发现 bDout1 基于给定时间 3S 来回闪烁，同时 A 的值 0-1000 递增，再从 1000-0 递减，不断循环

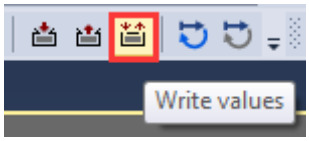


```
FB_Flash1(tCycletime T#3s :=t1 T#3s , bQ TRUE =>bDout1 TRUE )

IF A 302 =0 THEN
    B 1 :=1;
ELSIF A 302 =1000 THEN
    B 1 :=-1;
END_IF
A 302 :=A 302 +B 1 ;
RETURN
```

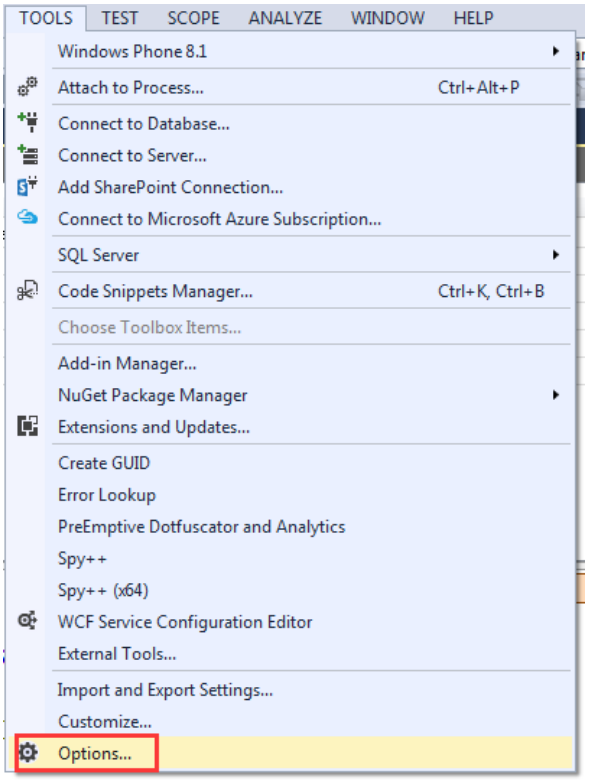
（16）如果希望对闪烁时间进行修改，只需要在 Prepared value 中输入对应时间，比如希望修改 1S 闪烁，因此只需要输入 T#1S

Expression	Type	Value	Prepared value
FB_Flash1	FB_Flash		
t1	TIME	T#3s	T#1s
bDout1	BOOL	TRUE	
A	INT	746	
B	INT	1	

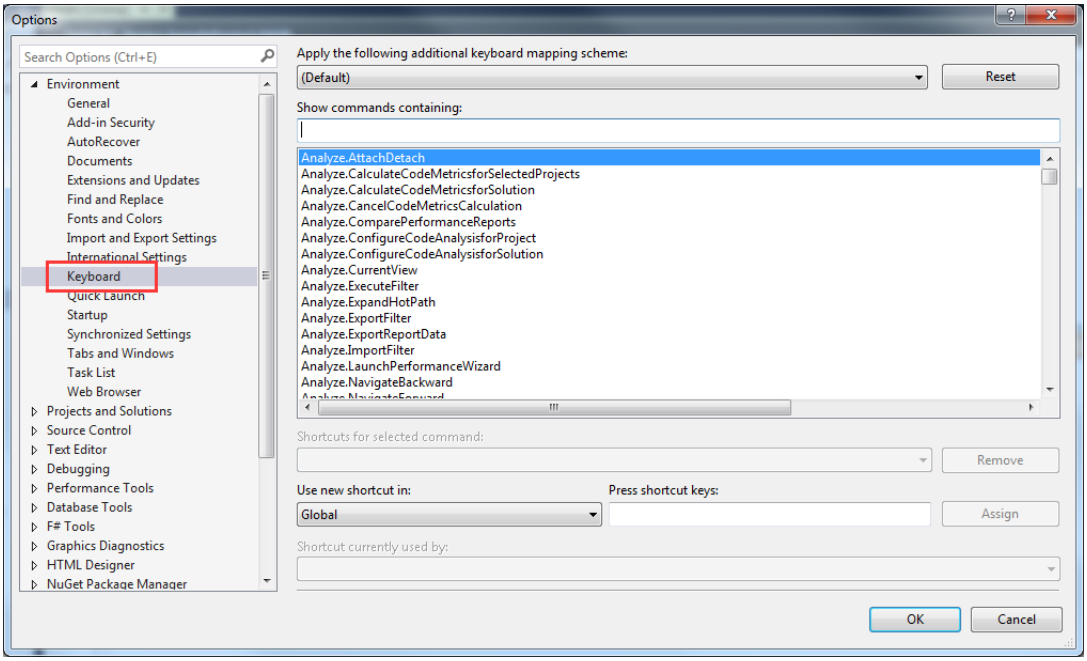
（17）在工具栏中找到 Write values，单击此按钮就可以把所需要修改的值输入到程序中



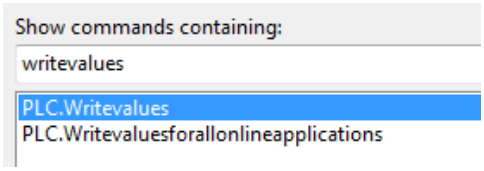
（18）如果希望这个按钮用快捷键来操作，那我们需要在 VS 中对此按钮进行快捷键分配，打开菜单栏 TOOLS，找到 Options



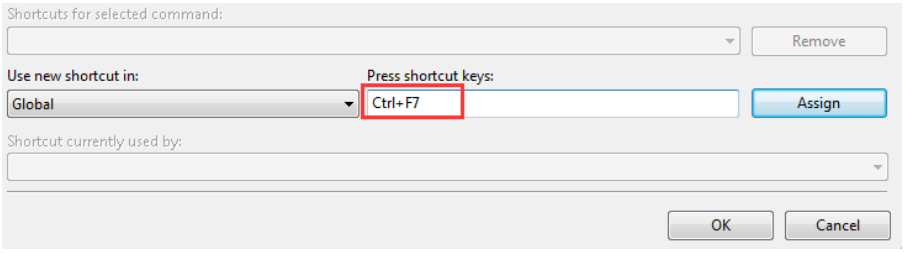
（19）找到 Environment 中的 Keyboard



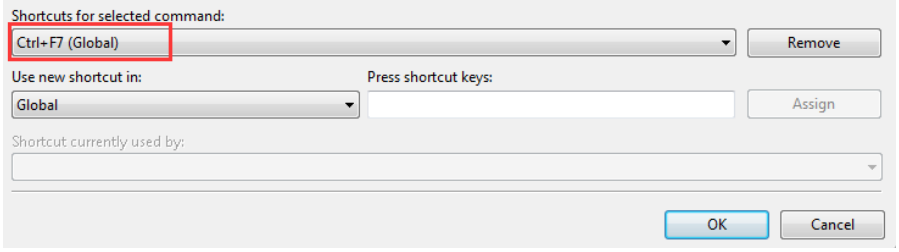
（20）在 Show commands containing 中输入希望分配快捷键的选项，比如 PLC 中的 writevalues，你可以直接输入 writevalues 进行查找



（21）随后在窗口下面 Press shortcut keys 中自定义快捷键，比如希望和 TC2 一样 ctrl+F7，那只需在这个窗口按下 ctrl+F7 即可



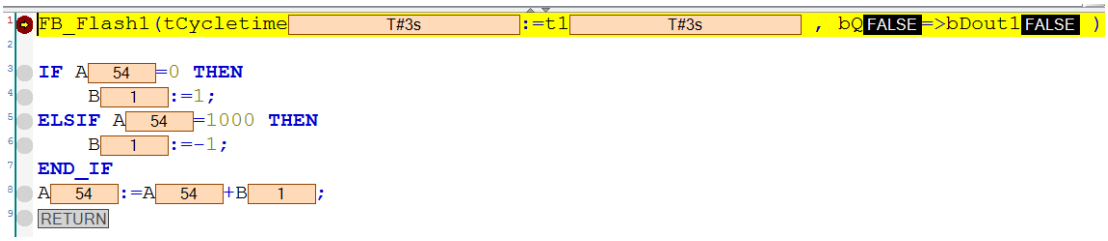
（22）点击 Assign 进行分配，就完成了快捷键的创建，当然如果希望移除此快捷键也只需要点击 Remove 即可



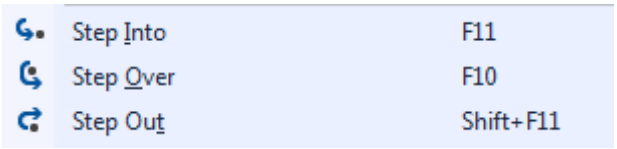
2. PLC 程序调试

2.1 断点（Breakpoint）调试程序

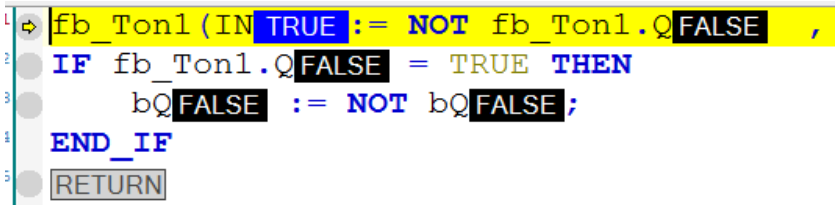
（1）创建断点很简单，只需要鼠标点击所需要添加断点的那一行，随后按快捷键 F9 即可，断电创建好后可以发现原本灰色的圆点变成了红色，并且断点当前位置用醒目的黄色标注



（2）通过工具栏中的快捷键就可以对程序进行一步一步调试

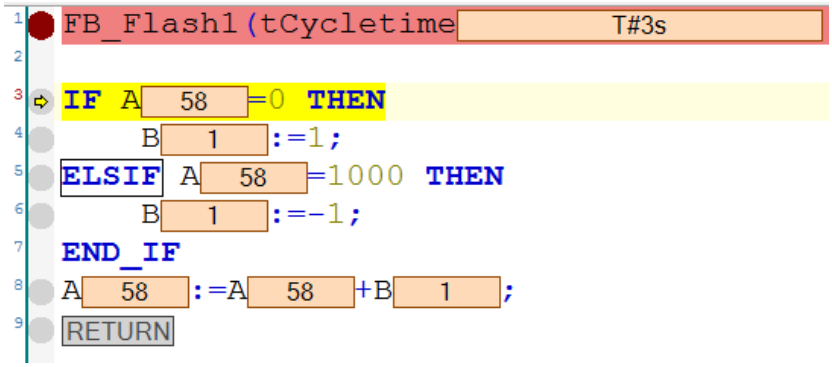


（3）Step Into 快捷键是 F11，主要对程序中所有程序步进行调试，即使是功能块，因此可以发现 Step Into 会进入功能块里面调试

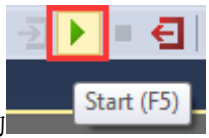


不过如果当此功能块是 BECKHOFF 自带的或者自定义的不开源库中的功能块，就无法进入功能块内部查看到调试结果，只能多按几下 F11 就会跳出。

（3）Step Over 快捷键是 F10，主要对主程序每一行进行调试，一行为一个单位，一个功能块算一步，因此可以发现按 F10 进行调试的结果，始终会在 MAIN 程序中依次执行



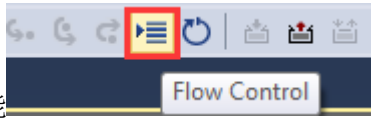
（4）Step Out 快捷键是 Shift+F11，当程序调试到某一段，可以通过 Step Out 把剩下的代码执行完，回到断点初始位置重新开始调试

（5）最后如果调试完，希望取消此断点，可以在添加断点的位置再按一次 F9 即可，此时可以发现红色圆点变成了灰色，这个时候只需要点击工具栏中的 

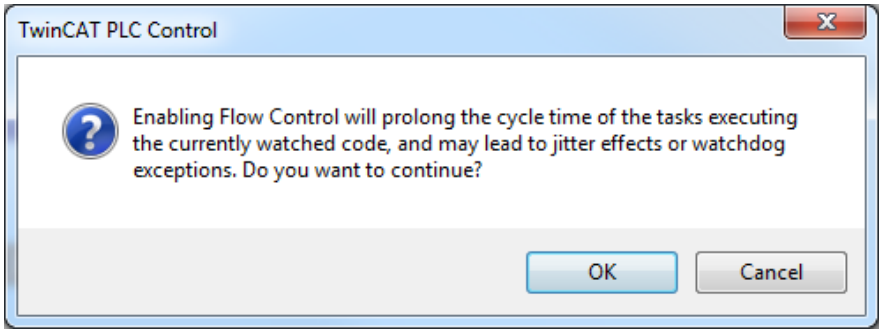
就可以继续执行代码了

```
FB_Flash1(tCycletime:=T#3s);

IF A:=72=0 THEN
    B:=1:=1;
ELSIF A:=72=1000 THEN
    B:=1:=-1;
END_IF
A:=72:=A:=72+B:=1;
RETURN
```

2.2 Flow Control 和单步循环

（1）Flow Control 是流控功能，在工具栏中可以激活此功能 

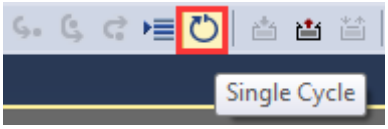
点击 Flow Control 弹出窗口点击 OK



（2）激活后会发现程序中所执行的代码中变量都变成了绿色，没有执行的都变成了白色，通过这个功能可以很清晰观察到那些代码执行，那些代码没有执行

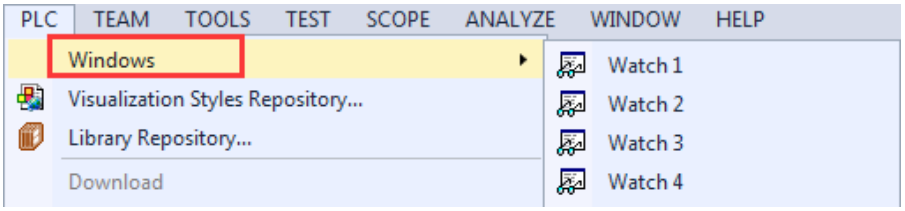
```
FB_Flash1(T#3s,tCycletime:=t1:=T#3s, bQ=>bDout1:=TRUE);

IF A:=119=0 THEN
    B:=1:=1;
ELSIF A:=119=1000 THEN
    B:=1:=-1;
END_IF
A:=120:=A:=119+B:=1;
RETURN
```

（3）在工具栏中还有一个单步循环的功能，单击此功能，程序会停止，每次点击一次 Single Cycle，程序会执行循环一次，当然这个按钮也可以设置快捷键，方便做单步循环的调试

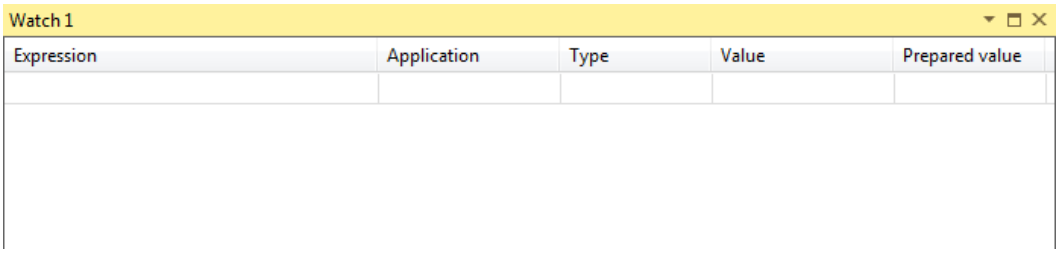


2.3 多变量同时在线监控

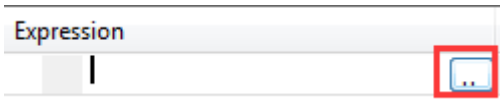
（1）在菜单栏 PLC→Windows 中找到有 Watch1-4



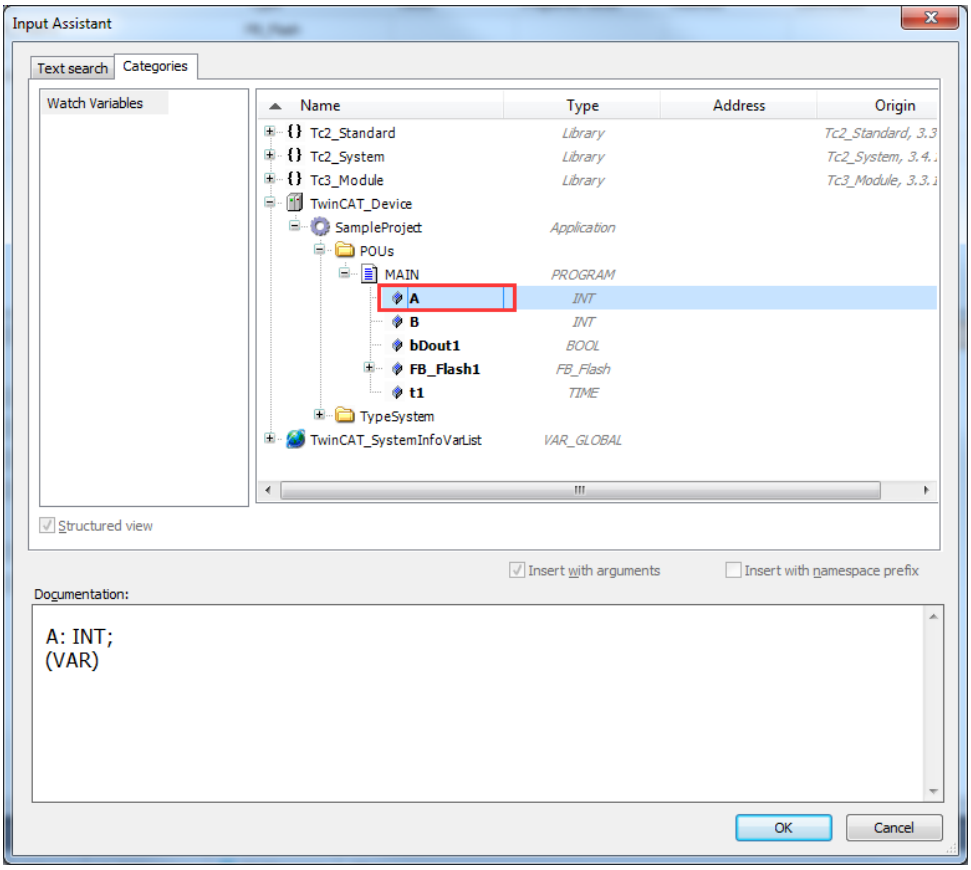
（2）选择任意一个，会弹出一个窗口，在里面可以把复杂程序中的变量集中监控



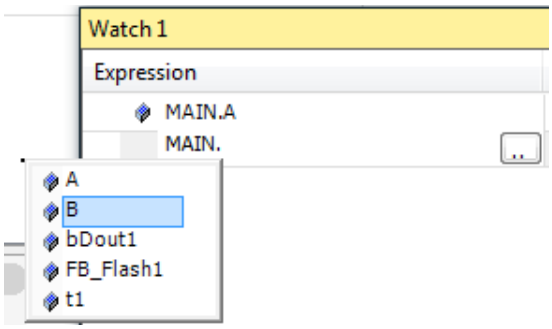
（3）双击 Expression，手动输入所需要监控的变量，或者在右边的按钮选择也可以



（4）比如选择 MAIN 程序中 A 变量，选中后点击 OK



（5）用手动输入的方法把 B 变量也加入进来



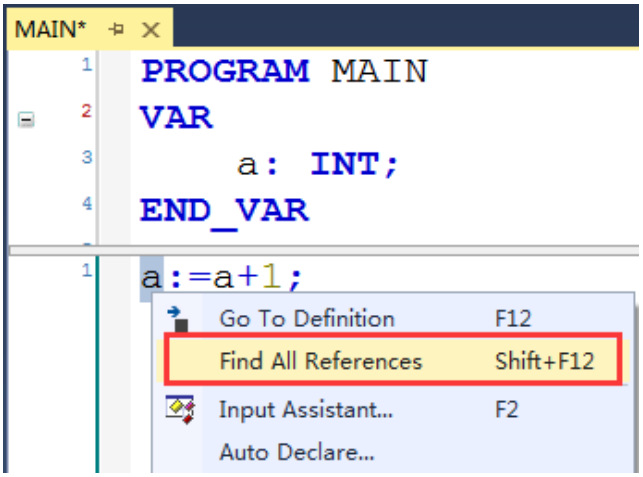
（6）此时就可以在这个窗口中同时监控这 2 个变量的变化情况，也可以对他们进行附值的操作

Watch 1

Expression	Application	Type	Value	Prepared value	Execution point
MAIN.A	TwinCAT_Device.Sa...	INT	987	1	Cyclic Monitoring
MAIN.B	TwinCAT_Device.Sa...	INT	-1		Cyclic Monitoring

2.4 交叉索引功能

（1）对于程序中的某个变量或者某个功能块，可以使用交叉索引的功能将用到该变量的地方都罗列出来。只要选中该变量，右键选择 Find All References



（2）列表所示就是所有该变量的使用情况，双击即可跳转。

Cross Reference List

Name: MAIN.a

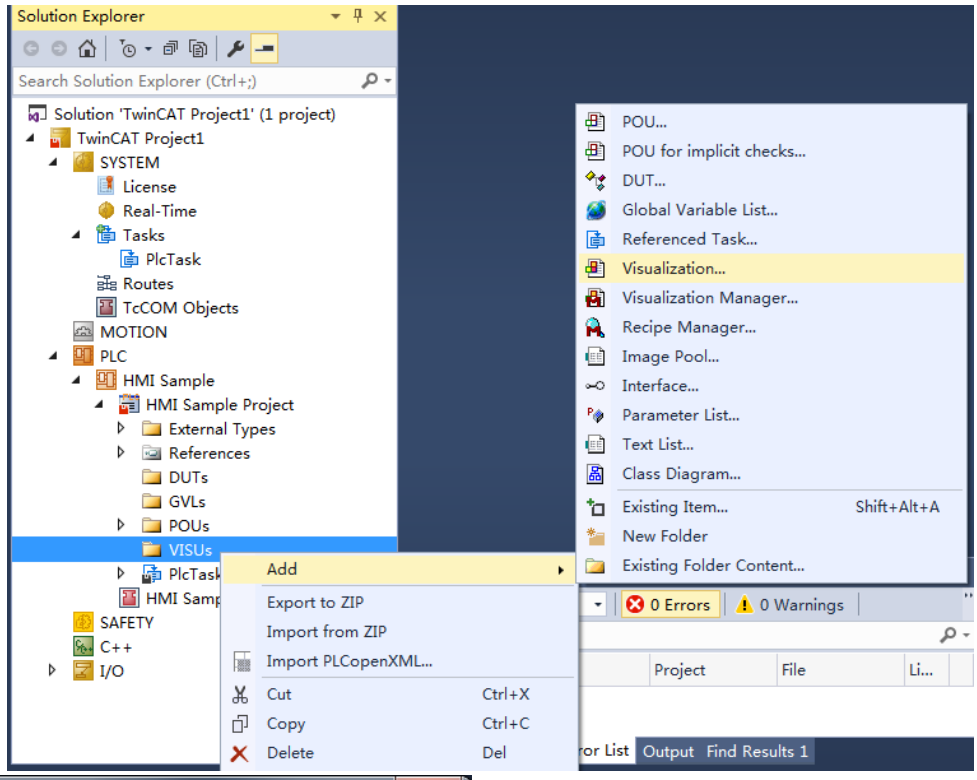
Symbol	POU	Variable	Access	Type	Addres
a	MAIN	a	Declaration	INT	
a	MAIN	a	Write	INT	
a	MAIN	a	Read	INT	

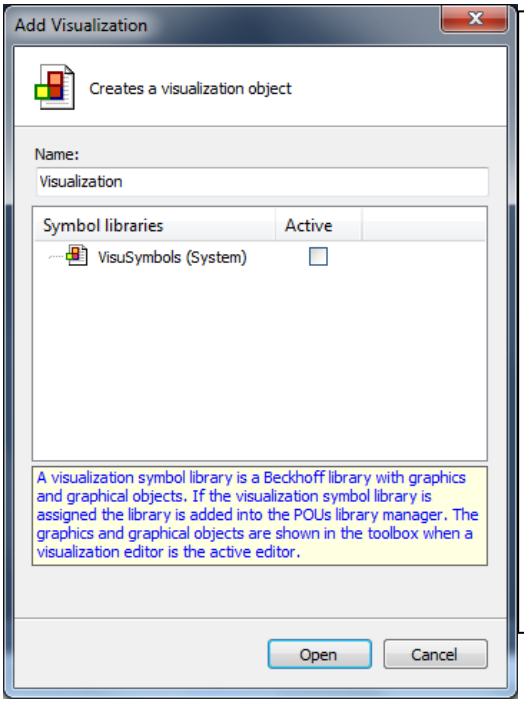
六、 TwinCAT3 PLC HMI 可视化编程

1. 可视化项目简介

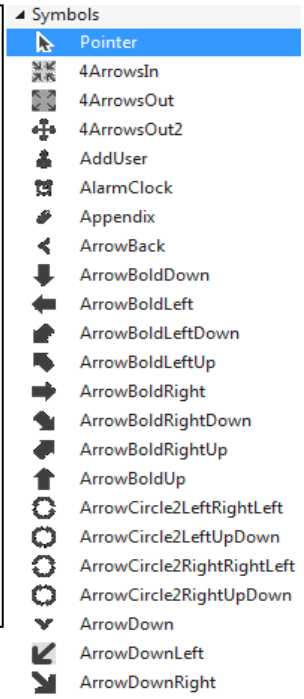
1.1 新建可视化项目

（1） 新建一个 PLC 项目，把这个新项目展开找到 VISUs 并右击新建一个新的可视化项目取名叫“visu1”，完成后点击 OPEN。过程如下

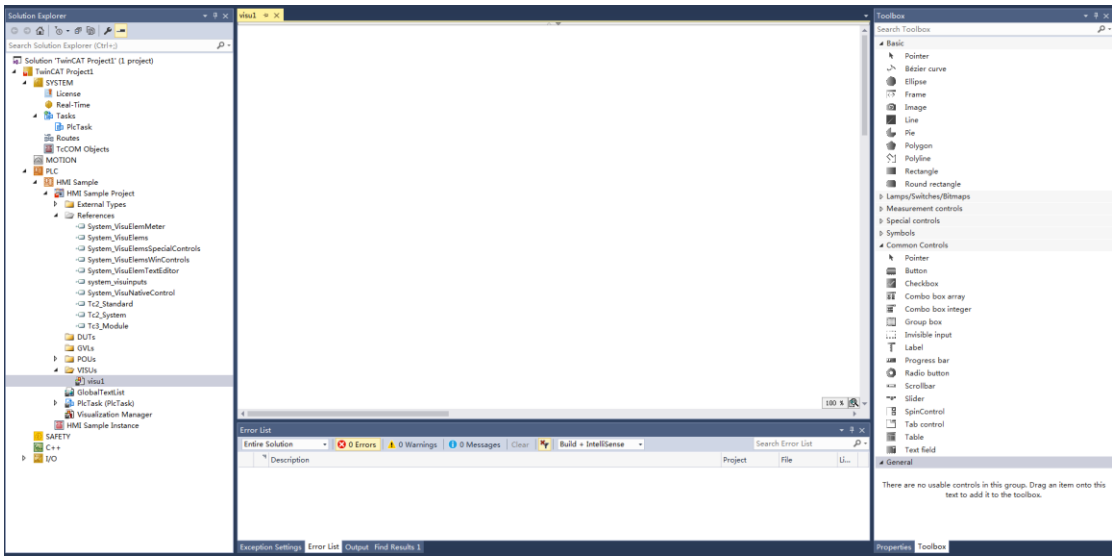




在点 Open 之前可以发现有 Active 选项可以勾选，这个选项的意思是在当前项目是否需要更多的图形控件，如果需要就勾选 Active，那么就会自动加载 Symbol libraries 并且在 toolbox 中多出许多图形控件，比如：各种箭头，电池，闹钟图标等等。

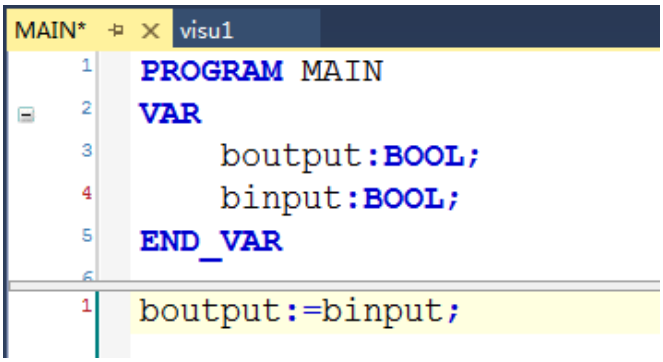


（2） 新建完后 TwinCAT 会直接进入 visu1 的界面，画面最右侧会出现工具栏，其中包含两个选项卡 toolbox 和 properties，可以进行控件选择和控件属性的调整。

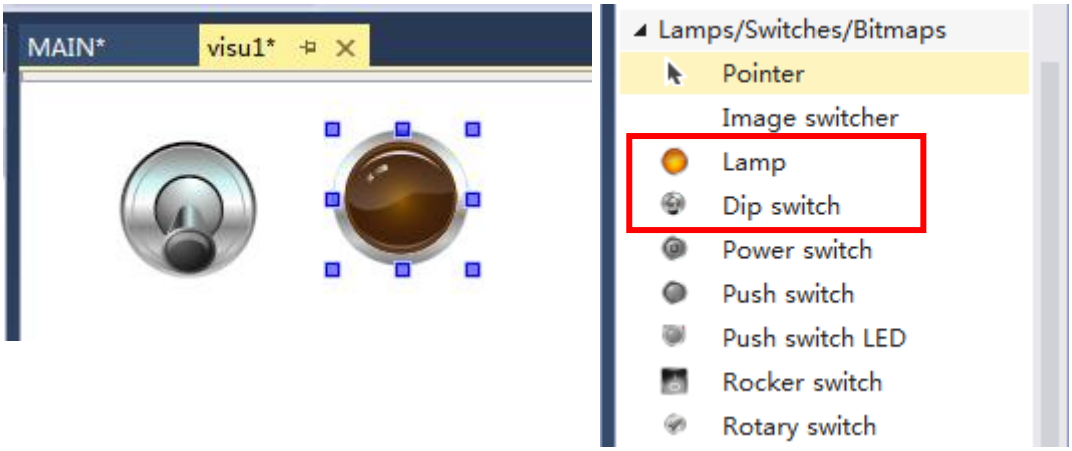


1.2 按钮控制小灯

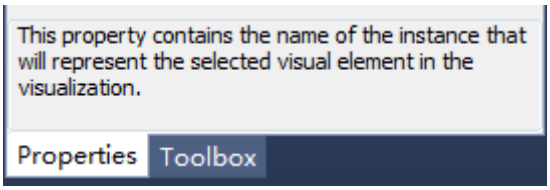
（1） 首先编写一个示例程序。bool 型变量 boutput 作为输出，binput 作为输入。程序如下：



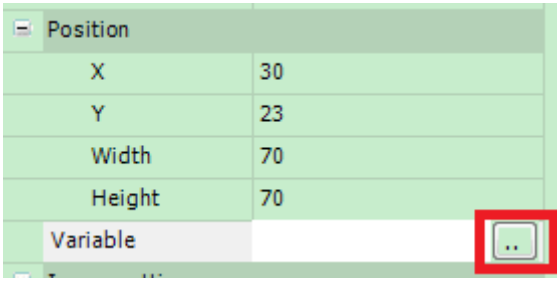
（2） 通过在 Toolbox 中找到 Lamps/ Switches/Bitmaps 选择控件 Dip switch（拨码开关）和 Lamp（位灯） 拖拉到 visu1 的画面当中，控件的大小可以直接调整。

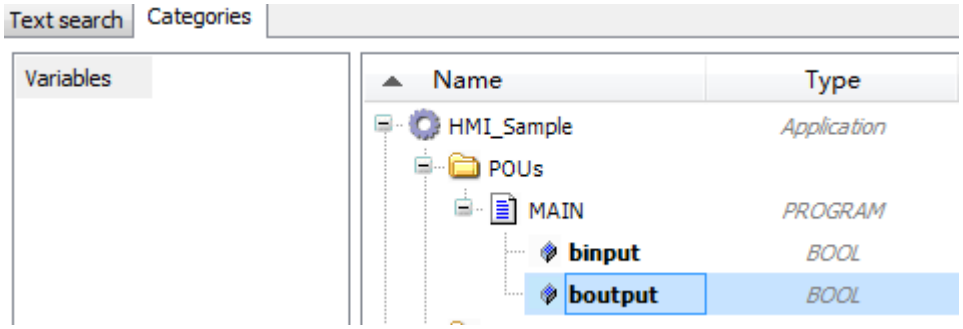


（3） 单击画面上的任一控件，会在右侧显示对应控件的属性栏。

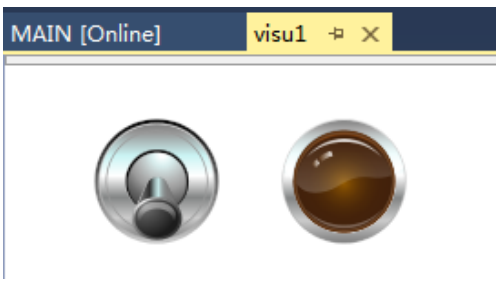


（4） 接下来需要将程序里的变量链接到 HMI 的控件当中。选中需要进行变量链接的控件，在属性栏里找到 Variable 标签，可以通过程序名称.变量名称的方式，直接进行输入例如：Main.boutput 或者找到空白栏右侧 ..，单击后在弹出的 Input Assistant 里找到程序中变量并进行变量链接。





（5） 上述操作完成后，点击工具栏激活图标 将当前项目切换成运行模式，完成后点击登入 再运行 ，效果如下：

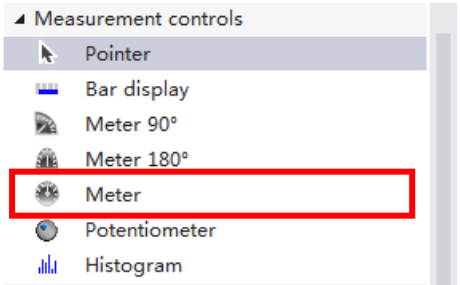


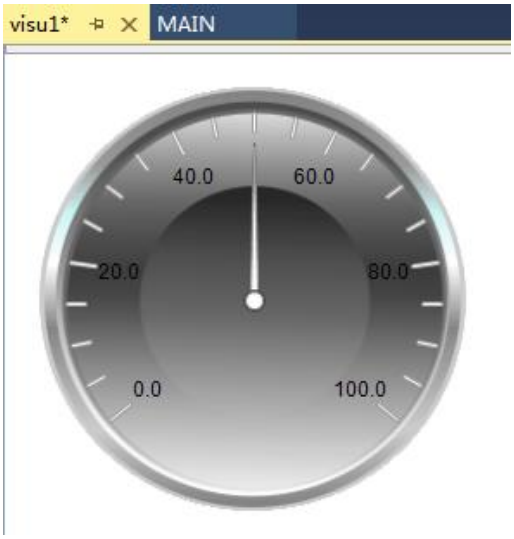
初始状态

运行后的状态

1.3 计量仪的添加和显示

（1）接下去介绍计量仪的添加，在右边 toolbox 里找到 Measurement Controlsr，可以选择不同度数的计量仪进行画面的编辑，以 Meter 为例。

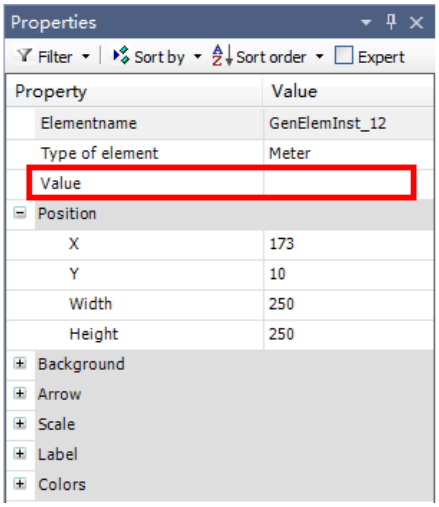


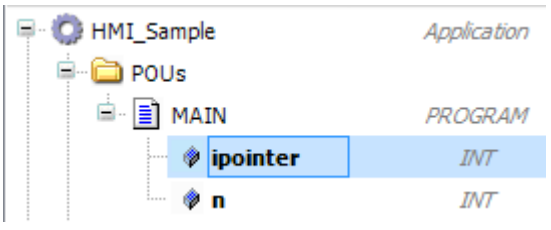


编写示例程序如下：

```
PROGRAM MAIN
VAR
    ipointer:INT;
    n:INT;
END_VAR

IF ipointer=0 THEN n:=1;
    ELSIF ipointer=1000 THEN n:=-1;
END_IF
ipointer:=ipointer+n;
```

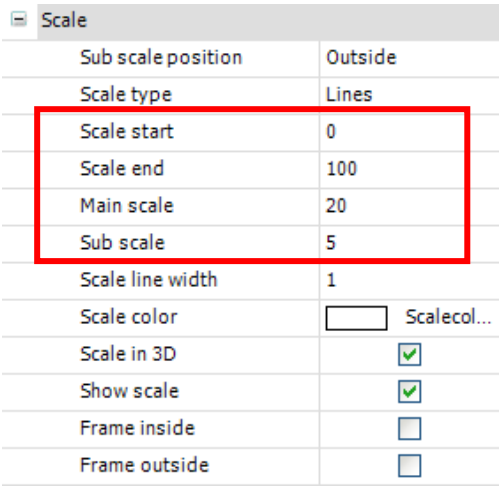
（2）选中计量仪，在属性栏里找到 Value 标签，在空白栏右侧点击[..]进行变量链接。



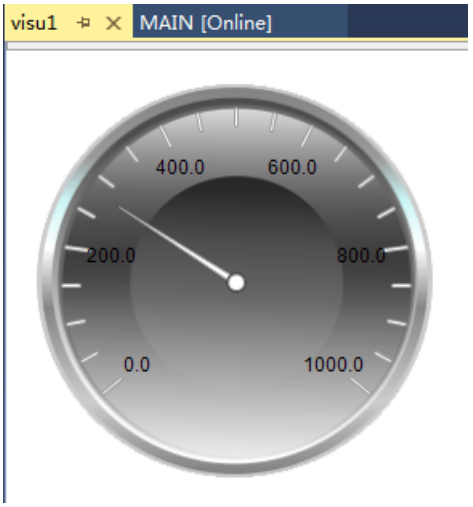


Property	Value
Elementname	GenElemInst_3
Type of element	Meter
Value	MAIN.ipointer

（3）计量仪刻度值的修改在属性菜单下的 Scale 中，展开 Scale，红框中是重点需要修改的参数，四个参数分别对应开始刻度、终止刻度、主刻度、副刻度。这里双击 Scale end 旁的栏目，将值改成 1000；把 Main scale 改成 200；把 Sub scale 改成 50



（4） 修改完成后效果如下：

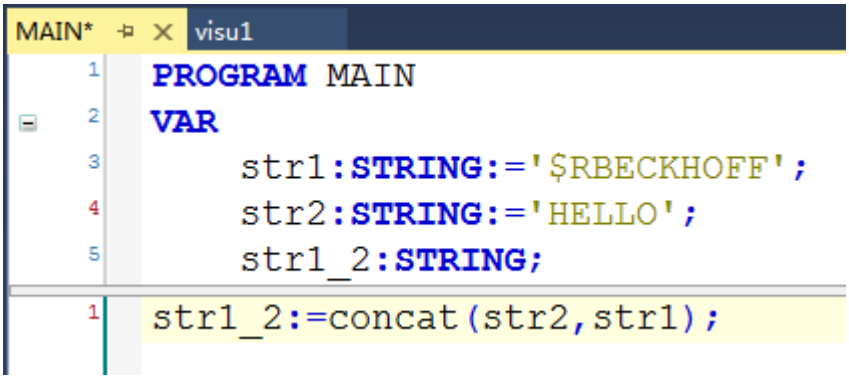


运行后的状态

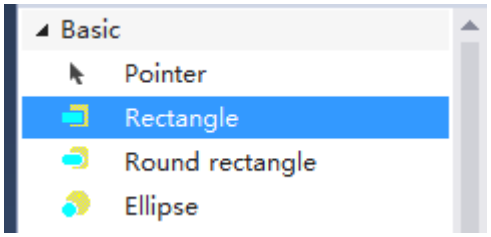
1.4 非固定常量的输入与显示

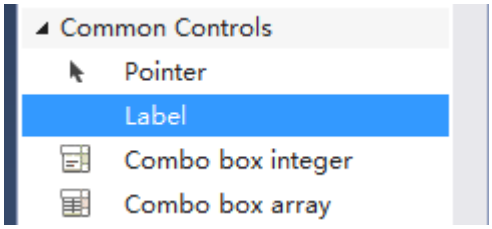
在 TwinCAT3 中数字、英文和中文字符（Wstring）都是可以在画面中进行显示和输入。以 string 为例：

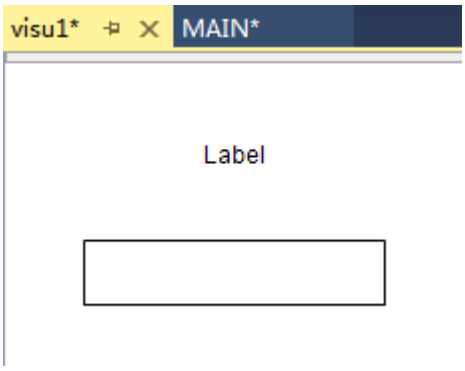
（1） 示例程序编写如下：



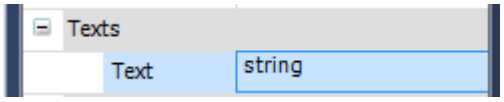
（2） 选用矩形框作为字符串输入框，在 Toolboxs 中找到 Basic→Rectangle，为了显示矩形框内输入内容添加一个标签，在 Toolboxs 中找到 Common Controls→Label，分别拖到 visu1 中



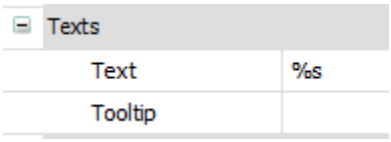


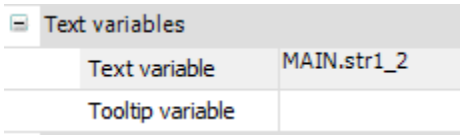


（3） 选中标签，右侧 properties 选项卡展开 Texts 标签，双击 Text 右侧的空白处，在框体中输入 string。或者也可以直接在控件上单击，输入字符。



（4） 选中矩形框，展开右侧 properties 选项卡中 Texts 标签，对输入内容作显示，对于非固定常量的显示，TwinCAT3 中有规定的格式，此例中为字符串数据类型，所以格式为：%s（字母区分大小写）；对于更多常用数据类型请参考下图表格。接下去在 Text Variable 中进行变量链接。

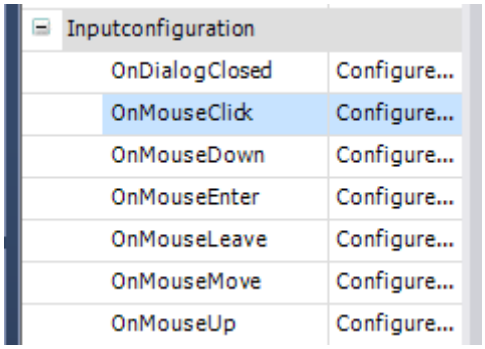


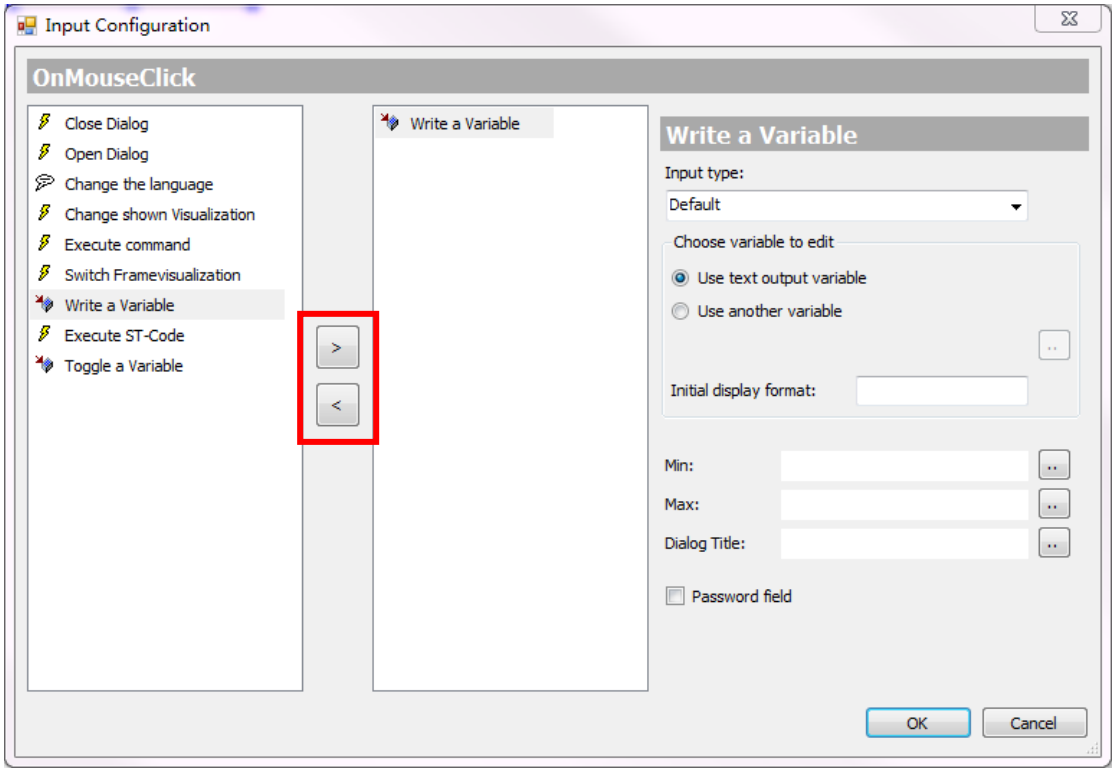


%b	Binary number
%c	Individual character
%d, %i	Decimal number
%f	REAL value
%o	Unsigned octal number (without prefixed zero)
%s	Character string
%u	Unsigned decimal number
%x	Unsigned hexadecimal number (without prefixed '0x')

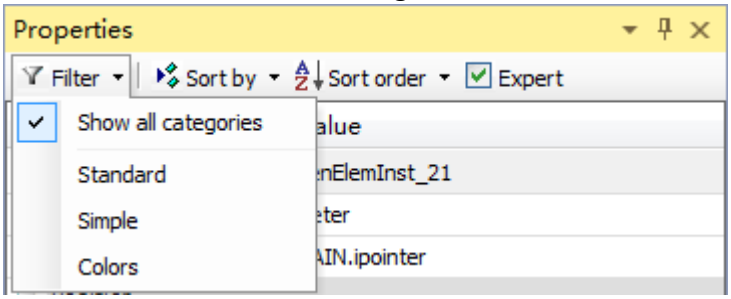
更多的数据类型显示格式，可以参考 TwinCAT3 帮助系统中更详细的说明：
https://infosys.beckhoff.com/content/1033/tc3_plc_intro/18014398645619339.html?id=185184623111977455

（5） 完成字符串的显示后，需要对其做输入。对画面中的内容做输入需要找到属性菜单中的 Inputconfiguration 标签，在该标签下可以看到鼠标的不同动作触发不同的输入方式。找到 OnMouseClick 点击 Configure…，在 Input Configuration 里双击或者选中后单击向右的引导箭头添加 Write a Variable 的操作，向左即为删除该操作。

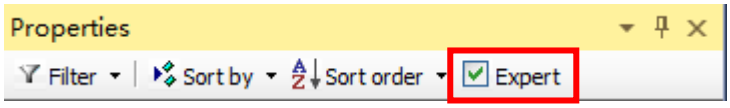




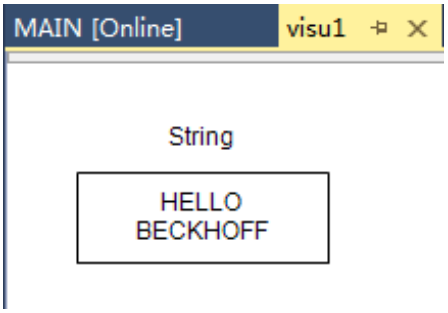
添加 Inputconfiguration 下鼠标操作动作时，如果需要显示所有的鼠标动作，需要在属性菜单的 Filter 下拉选择 Show all categories



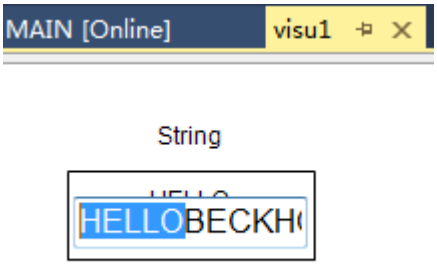
如果希望看到更多的参数列表请勾选属性菜单 Expert



在 Input type 中选择输入方式，如果输入内容即显示内容选择 Use text output variable，激活并运行程序后效果如下：

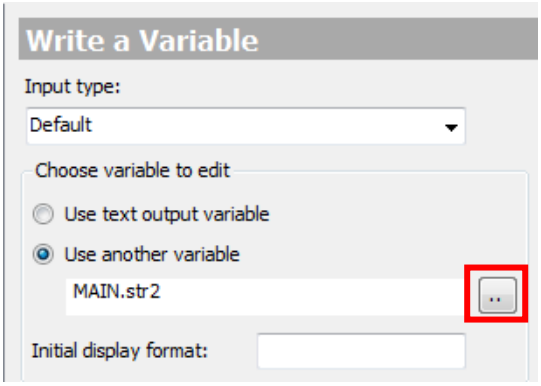


运行后状态

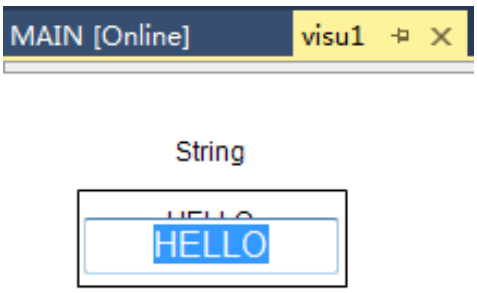


修改变量值

如果输入内容不为显示内容则选择 Use another variable 选项，并进行变量链接。以 str2 为例，运行后修改变量值效果如下：

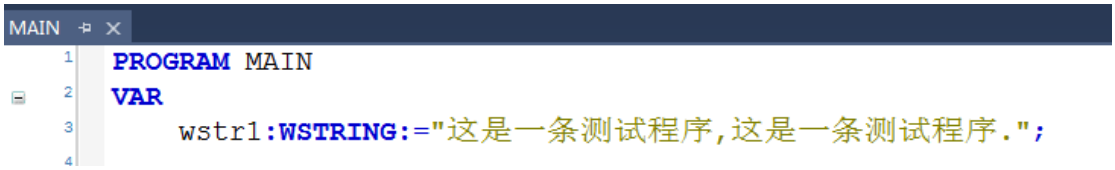


进行变量链接

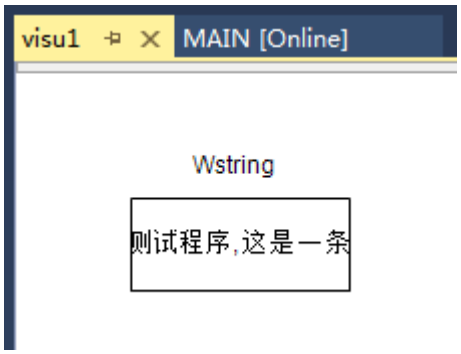


修改变量值

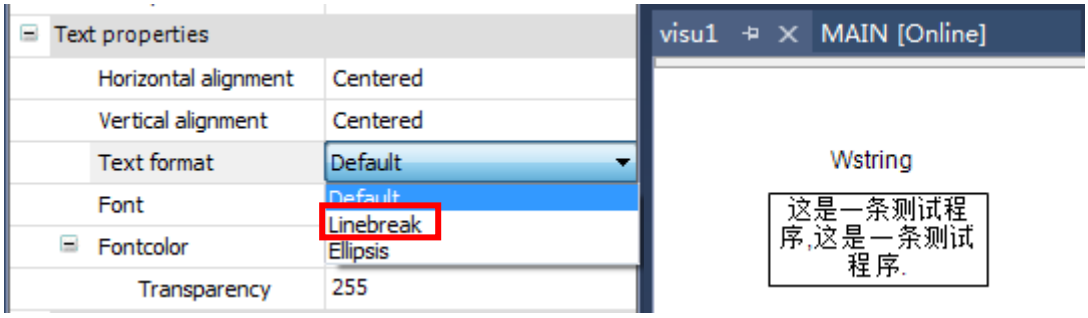
（6） 在 TwinCAT3 中，当字符长度超出单行显示范围的时候，除了在程序中定义‘$N’进行换行之外，也可以直接通过属性页面中的 Line break 来实现，以 Wstring（字符串类型支持中文字符）编写示例程序如下:



在 HMI 中通过添加矩形框进行字符串的显示，配置过程请参考(1)-(5)，在没有进行换行操作之前，激活并运行程序，在线显示效果如下：



可以看到矩形框内无法显示全部的变量内容，选中矩形框，在右边属性菜单中找到 Text properties→Text format，在 Text format 选项卡下面选择 Linebreak 进行自动换行，Elipsis 代表省略无法显示部分内容，效果如下：

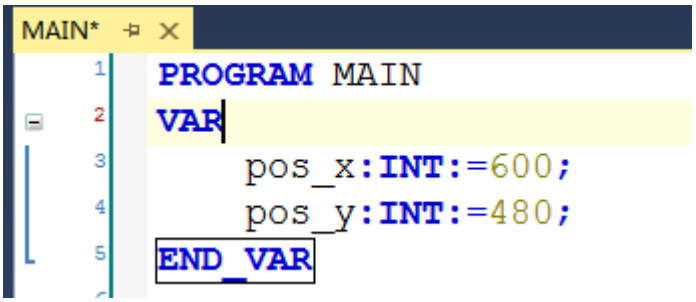


Label 作为标签想要实现换行，只需要在输入显示内容时，选择 Ctrl+Enter 进行换行操作。

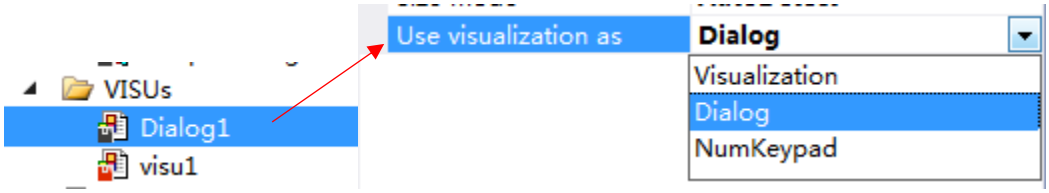
1.5 Dialog 会话框的使用

在 TwinCAT3 中同样可以实现会话窗口的调用。

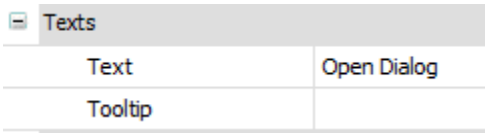
（1） 首先编写一段示例程序，其中 pos_x，pos_y 用于定义会话窗口的位置，数据类型为 INT，初值需要根据当前画布大小确定。



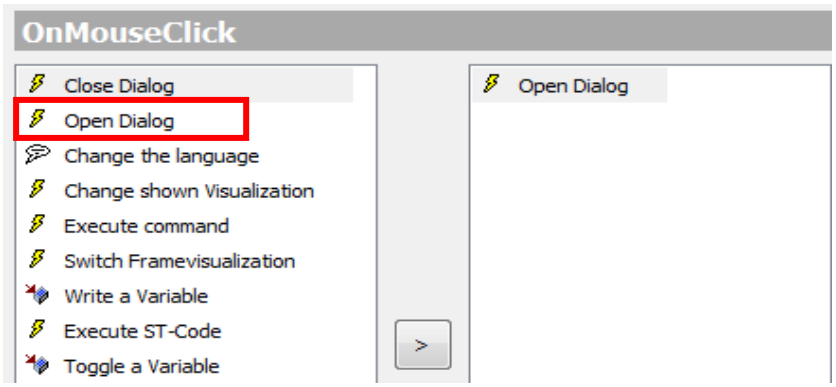
（2） 在 VISUs 文件夹右击添加新的可视化界面命名为 Dialog1。添加完成后，选中 Dialog1，找到右边属性选项卡下的 Use visualization as 选择 Dialog



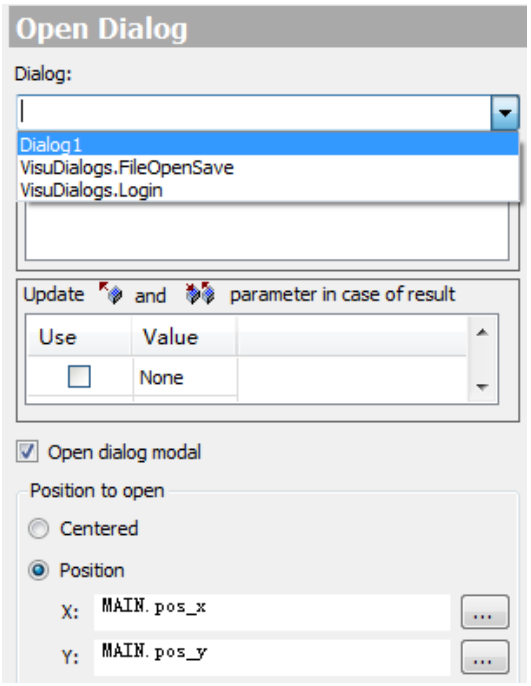
（3） 进入画面 visu1 中添加按钮，在 Toolbox 中找到 Common Controls→Button，拖到画面中，在按钮中直接输入显示内容或者在右边属性选项卡的 Texts 标签下找到 Text 进行输入



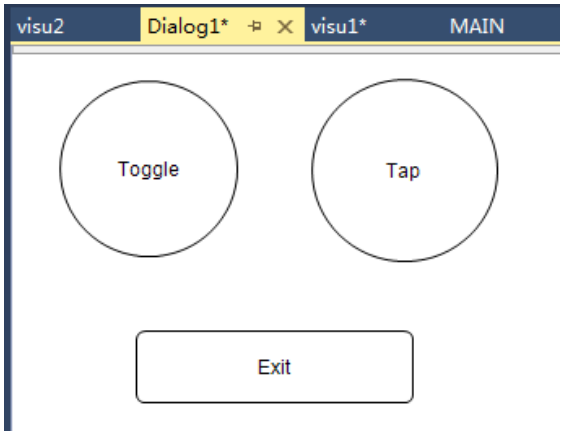
（4） 在属性选项卡中找到 Inputconfiguration→OnMouseClick，单击 Configure…，在弹出的 Input Configuration 中双击 Open Dialog 进行添加



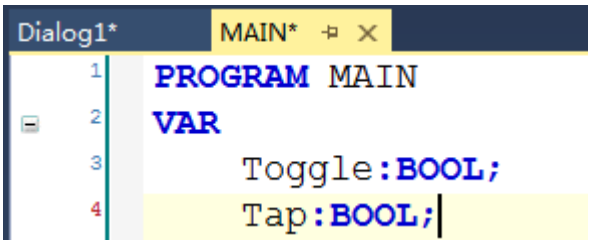
（5） 单击 Open Dialog 在 Input Configuration 的右边找到 Dialog 选择之前添加的 Dialog1，并在下方 Position to open 中通过[..]链接示例程序中的变量，用以改变 Dialog 在调用窗口中的位置。



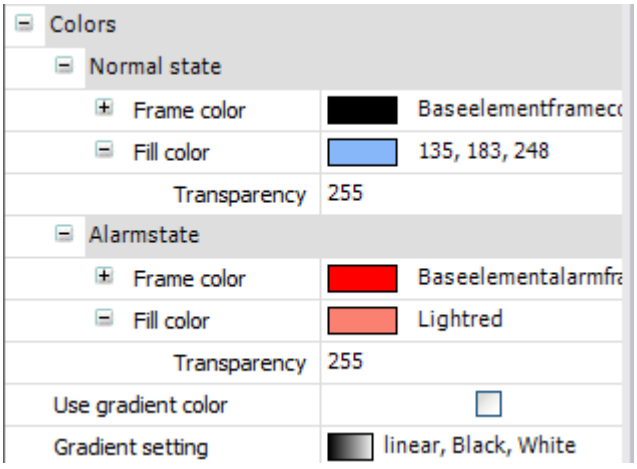
（6） 在新建的 Dialog1 添加两个开关，分别实现 Toggle 和 Tap，在 Toolbox 中找到 Basic→Elipse、Rectangle，在椭圆形和矩形框中直接输入或在右侧 Text 标签中分别输入 Toggle、Tap、Exit，效果如下：



（7） Toggle 是自锁按钮，Tap 为自复位按钮，以 Toggle 为例，首先在程序中添加示例程序

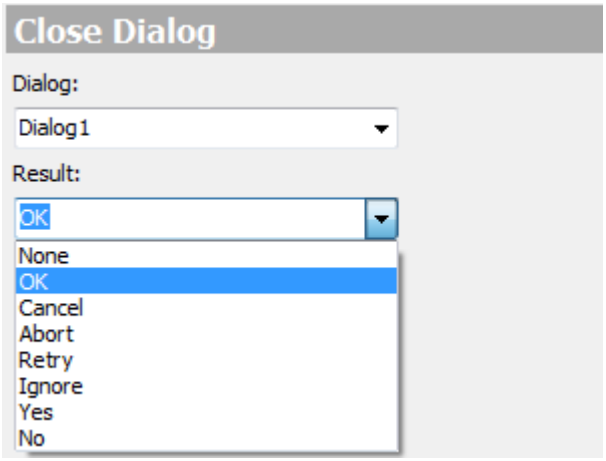


（8） 回到 Dialog1，在画面中选中 Toggle 按钮，通过属性菜单下 Inputconfiguration→Toggle→Variable 链接到程序中变量，可以通过 Colors 下的 Normal state 和 Alarm state 进行 Toggle 变量不同状态下的颜色设置



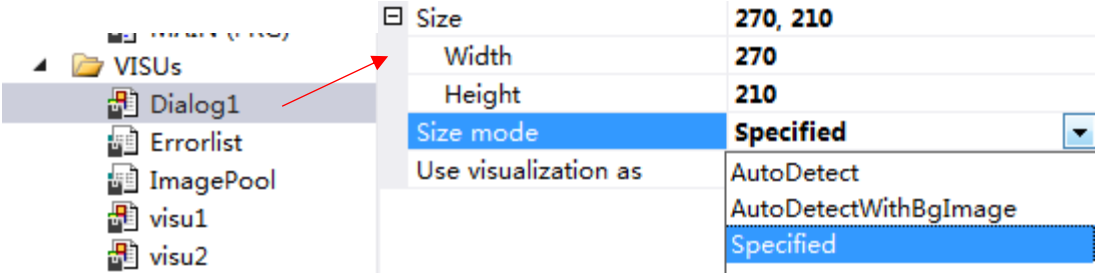
（9） 对 Tap 按钮进行相同的配置后，选择 Exit，找到右侧属性菜单中的 Inputconfiguration→OnMouseClick，单击 Configure…，在弹出的 Input Configuration 中双击 Close Dialog 进行添加

（10）添加完成后，在右边 Close Dialog 中选择 Dialog1，并在 Result 中选择 OK 作为用户反馈值

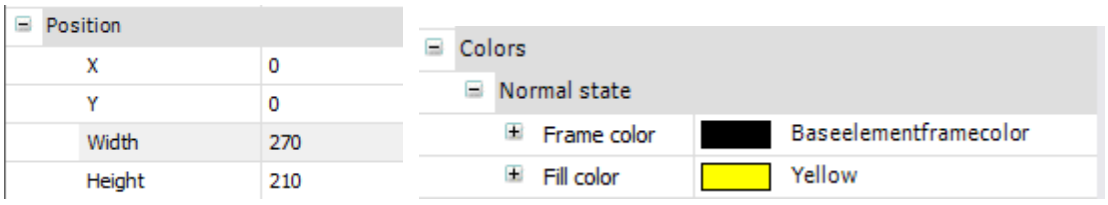


（11）需要注意的一点是，在 Dialog1 界面中配置的 Background 无法显示在弹出的对话框中，如果需要进行背景的显示，可以通过 Rectangle 来实现，实现步骤如下：

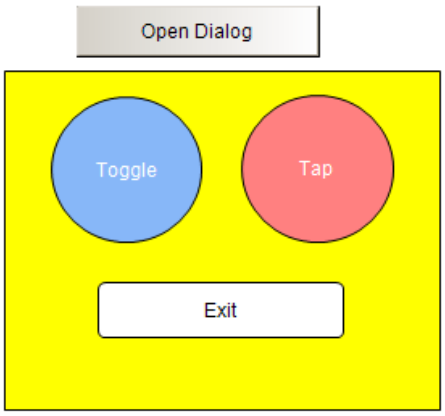
在 Dialog1 界面中添加矩形框，可以在 Dialog1 的属性中选择 Dialog1 的大小为自定义 Specified



按照 Dialog 的大小手动配置矩形框的大小进行匹配，将匹配好大小的矩形框设置需要的背景色后右击选择 Order→Sent to back



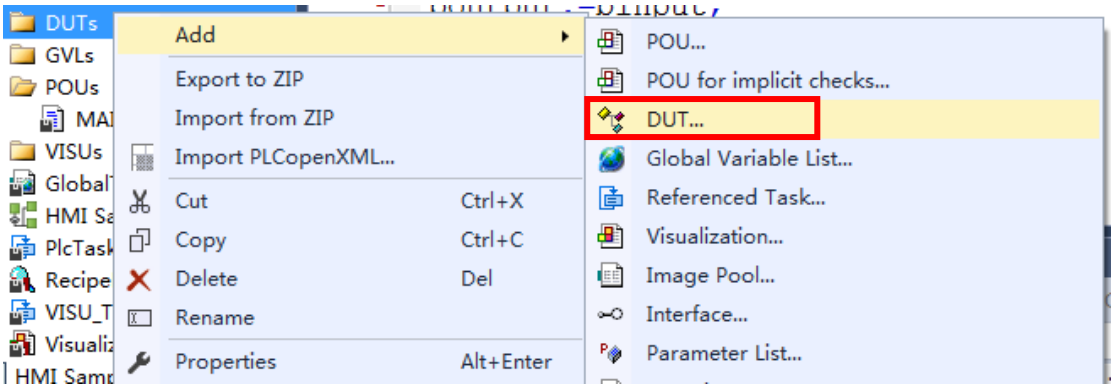
（12）激活并运行程序，点击界面中的 Open Dialog 按钮，Dialog1 就会弹出，如果需要退出 Dialog1，点击 Dialog1 画面中的 Exit 按钮即可，运行效果如下:



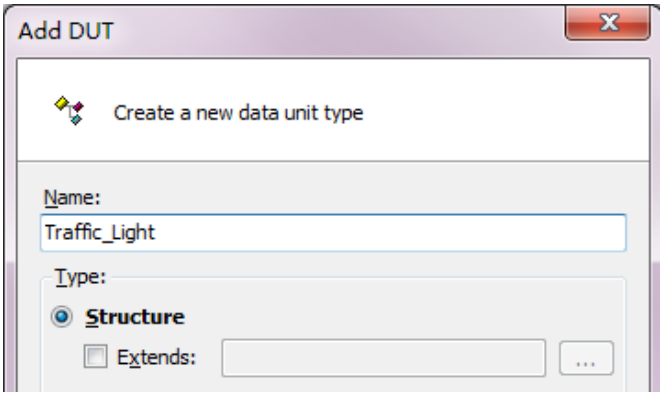
1.6 自定义控件

在编写程序时，用户可以通过自定义数据类型声明特殊的数据类型实现特殊功能，在编写 HMI 画面时，同样可以通过声明自定义控件创建画面中需要使用到的特殊的控件类型。

（1） 创建自定义数据类型，作为例子程序 ，在 PLC 下 DUTs 文件夹右击，选择 Add 添加 DUT



选择添加结构体，命名成 Traffic_Light,点击 Open 创建



结构体编写程序如下：

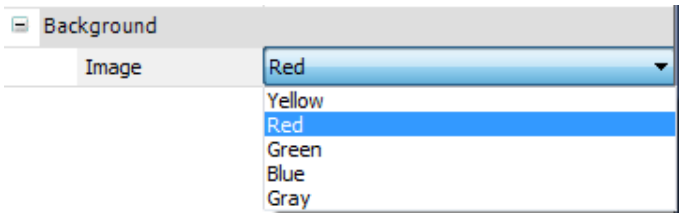
```
TYPE Traffic_Light :
STRUCT
    Red:BOOL;
    Yellow:BOOL;
    Green:BOOL;
END_STRUCT
END_TYPE
```

自定义数据类型需要在程序中进行变量声明，实例化后在主程序窗口编写简单示例程序：

```
PROGRAM MAIN
VAR
    Traffic1:Traffic_Light;
    Traffic2:Traffic_Light;
```

```
Traffic1.Red:=TRUE;
Traffic2.Green:=TRUE;
```

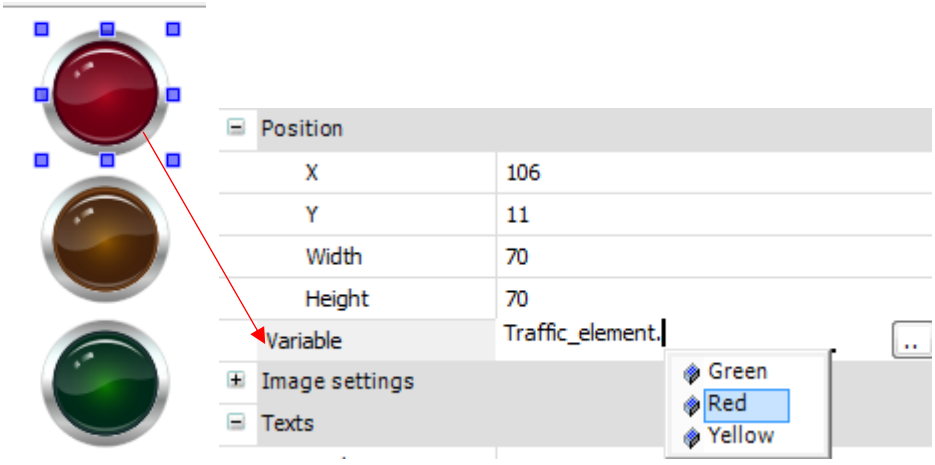
（2） 创建新的可视化界面，命名成 Traffic_element 后，在 Interface Editor 中编辑输入输出变量，并编辑基础控件来创建自定义控件，采用 ToolBox→Lamps/Switches/Bitmaps→Lamp 编辑交通灯，通过属性菜单下的 Background→Image 可以修改 Lamp 的颜色



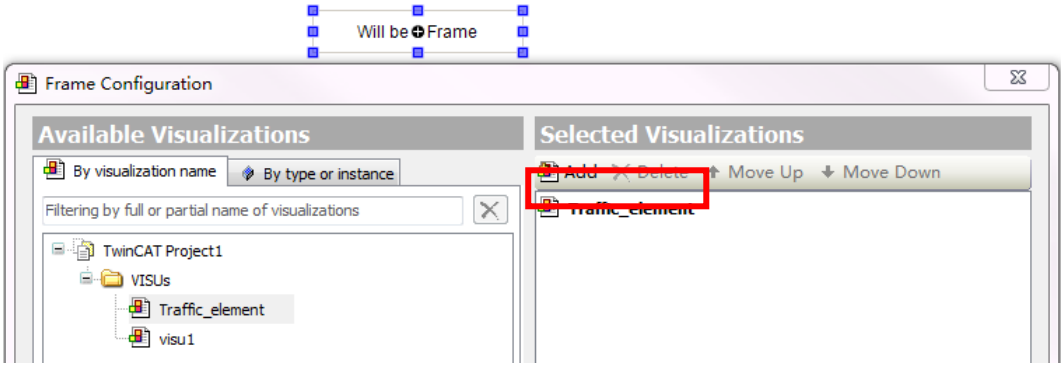
重复添加三个 Lamp，并按照 Red,Yellow，Green 分别修改 Lamp 的颜色，效果如下：

```
VAR_IN_OUT
    Traffic_element:Traffic_Light;
END_VAR
```

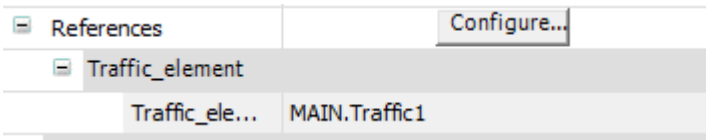
（3） 在属性菜单中找到 Position 下的 Variable，双击后在对话框中输入在 Interface Editor 中实例化的结构体 Traffic_element，通过点索引的方式调用结构体中元素，按照顺序分别调用 Red,Yellow,Green



（4） 双击进入 visu2 的画面中，在右边的 Toolbox→Basic 下找到 Frame，添加 Frame（框架）到画面中，在自动弹出的 Frame Configuration 左边找到 Traffic_element，选中，双击或单击对话框中 Add 添加作为框架的画面，Delete 可以删除已添加画面，点击 OK

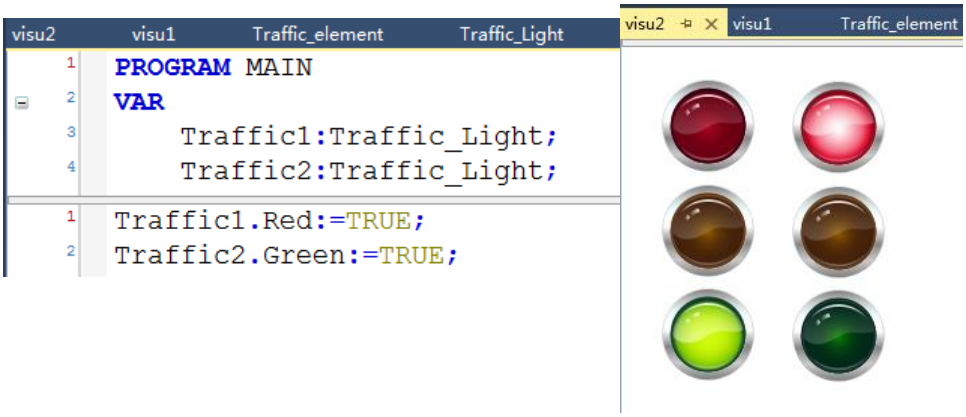


（5） 将 Traffic_element 作为 Frame 添加到 visu2 中后，需要为新添加的控件做变量链接，在右边属性菜单中找到 Reference 展开 Traffic_element 链接程序中实例化的结构体



如果需要添加多个交通灯只需要添加 Frame 并链接程序中实例化后的结构体即可，此时 Traffic_element 相当于用户自定义的控件，可以通过类似于基础控件的添加方式进行添加，变量链接等操作。

（6） 按照以上步骤在画面中添加两个 Traffic_element 控件，分别连接到程序中实例化的的 Traffic1 和 Traffic2 上，激活并运行程序，效果如下：

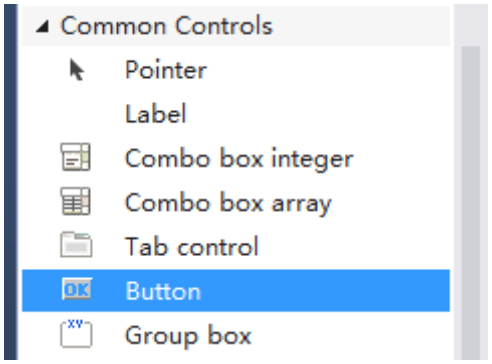


1.7 画面切换

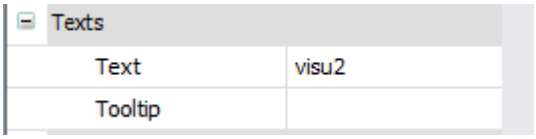
很多时候工程需要不止一个画面，所以这里介绍一下画面切换的方法。

（1） 新建一个画面，找到 VISUs 右击新建一个新的可视化项目取名叫“visu2”，完成后点击 OPEN

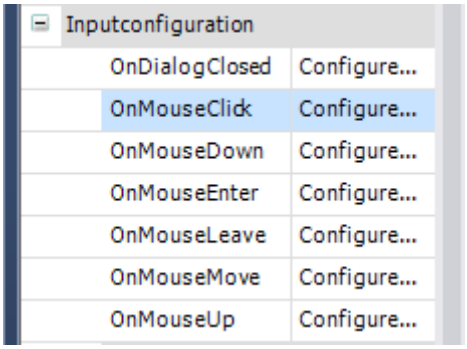
（2） 这里选用按钮，在 Toolboxs 中找到 Common Controls→Button，拖到 visu1 画面中



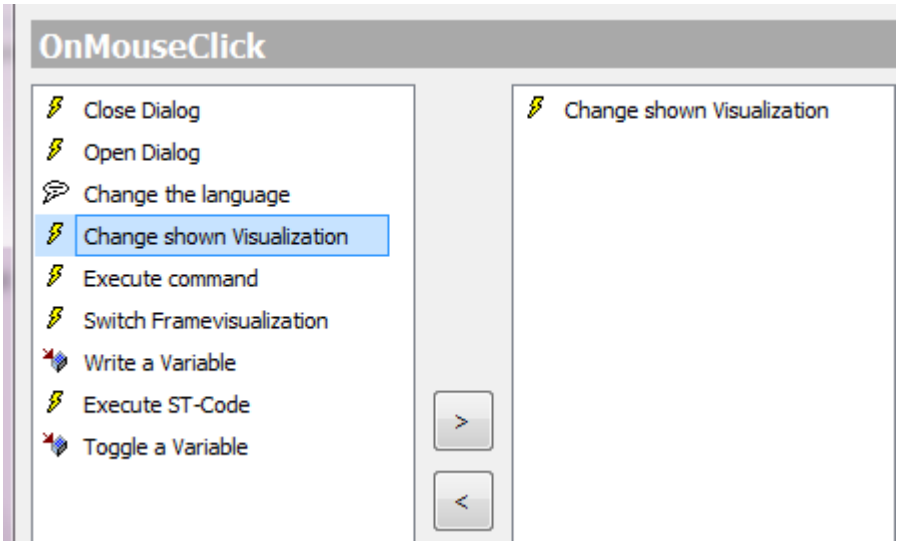
（3） 选中按钮，在右侧 properties 选项卡中展开 Texts 标签，双击 Text 右侧的空白处，在框体中输入 visu2



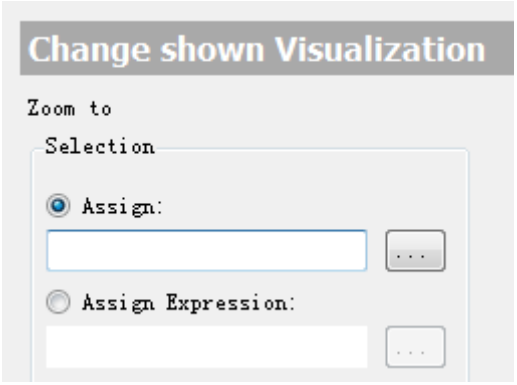
（4） 填写完成后，在 properties 选项卡展开 Inputconfiguration 标签，找到 OnMouseClick，单击其右侧的 Configure…



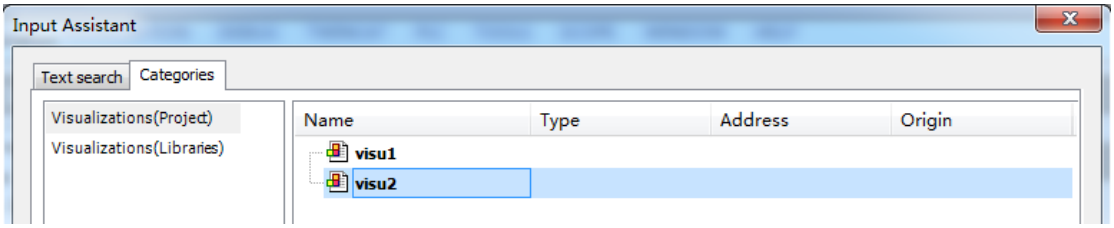
（5） 在弹出的对话框中，找到左侧的 Change shown Visualization 功能，双击该功能，对话框右侧会显示其属性



（6） 在右侧属性栏点击 Assign 下空白栏右边的小按钮

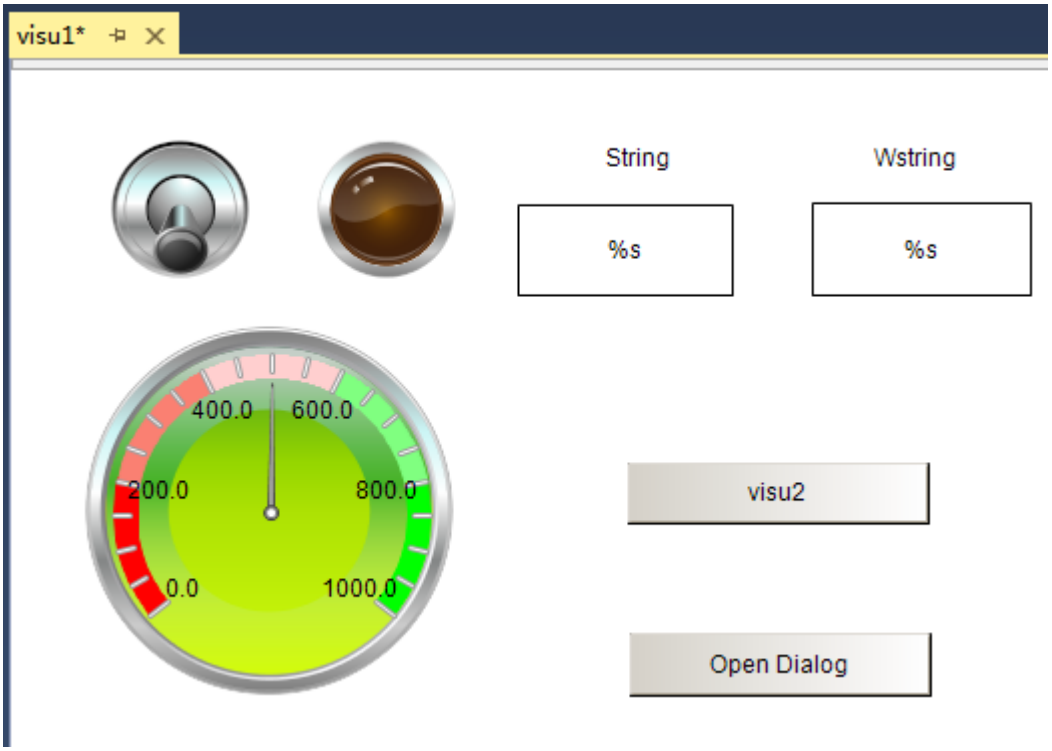


在弹出的对话框中会显示所有已有的画面，选中 visu2，点击 OK。



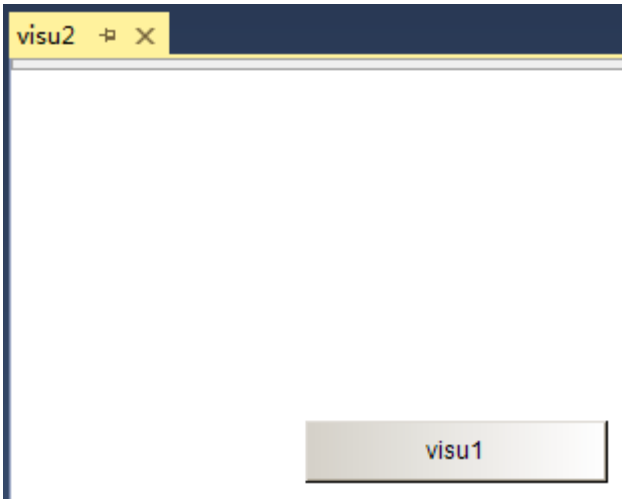
（7） 同理，双击 visu2，来到画面 2。添加按钮，命名为“visu1”，添加 Change shown Visualization 功能 ，在右侧属性栏点击 Assign 下空白栏右边的小按钮，在弹出的对话框中选中 visu1，点击 OK。

（8） 上述操作完成后点击 激活配置，点击登入 再运行 效果如下：



画面一

单击 visu2 按钮，来到画面二。



画面二

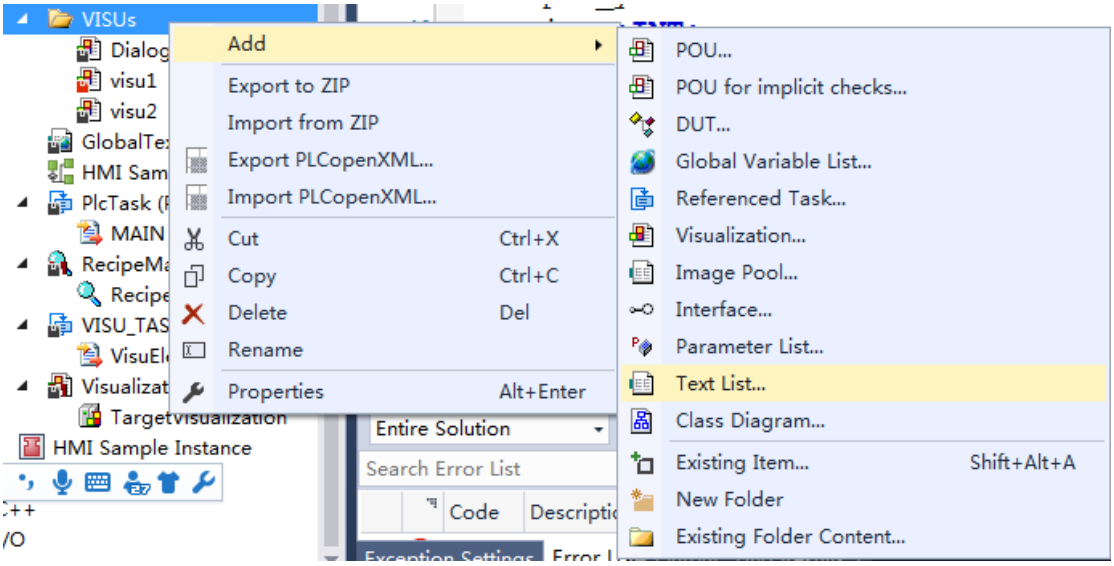
1.8 动态文本显示

动态文本的显示体现在 Errorlist 的应用。因为不存在实际的报错，在程序中声明 INT 型变量，用改变量的值来显示不同的报错信息。

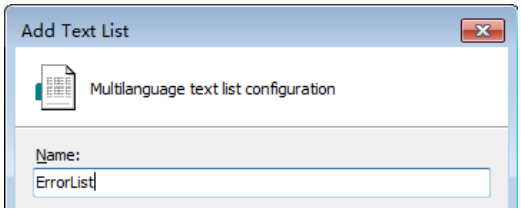
（1） 编写示例程序如下，添加 iVar 变量

```
PROGRAM MAIN
VAR
    str1:STRING:='$RBECKHOFF';
    str2:STRING:='HELLO';
    str1_2:STRING;
    binput:BOOL;
    boutput:BOOL;
    ipointer:INT;
    n:INT;
    pos_x:INT:=100;
    pos_y:INT:=80;
    iVar:INT;
END_VAR
```

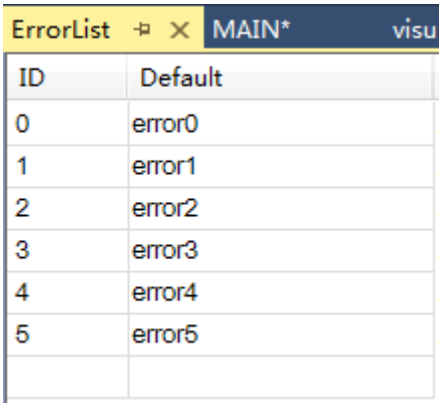
（2） 接下去添加报错列表，在左侧配置管理窗口，找到 VISUs 文件夹，右键 Add，在菜单中选择 Text List



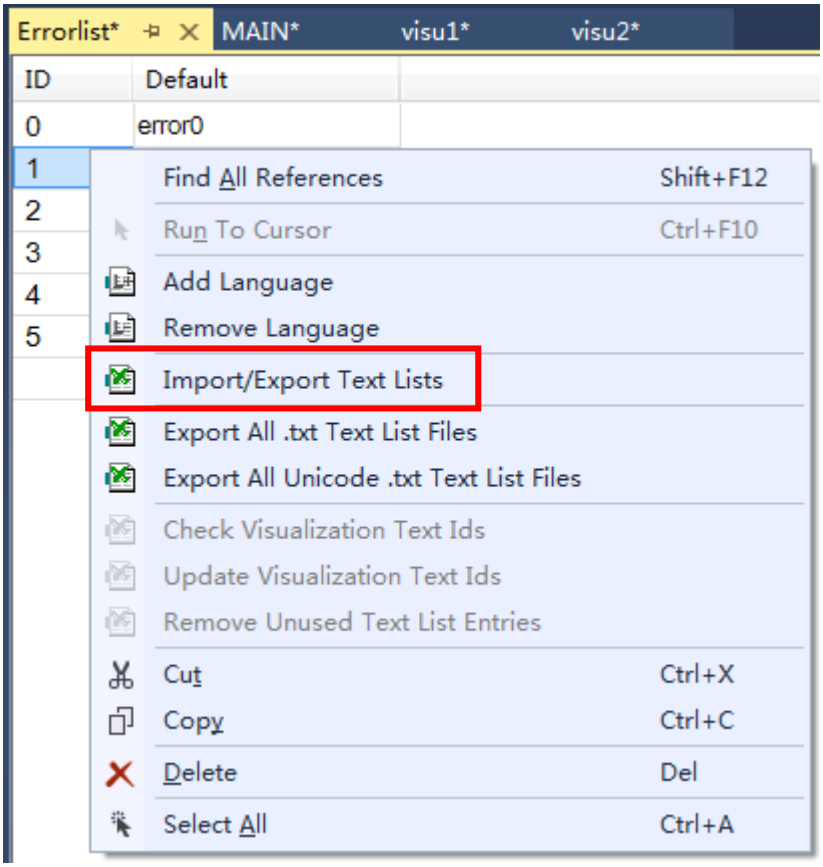
在弹出的对话框中将其命名为 ErrorList



（3） 在新建的 ErrorList 中编辑报错列表，在 ID 一栏输入数值，在 Default 一栏输入需要做显示的报错信息

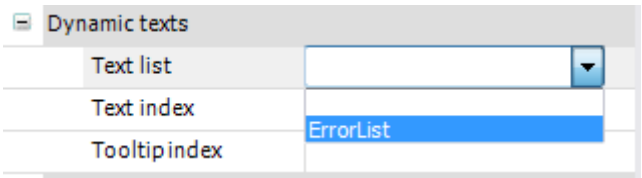


也可以通过导入编辑好的报错信息添加 Errorlist 中的内容，在新建 Errorlist 空白处右击进行添加

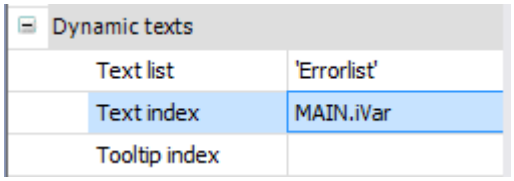


（4） 在画面中添加矩形框来作为报错信息的显示窗口，在 Toolbox 中选择 Basic→Rectangle；继续添加 Common Control→Label 作为动态文本的标签

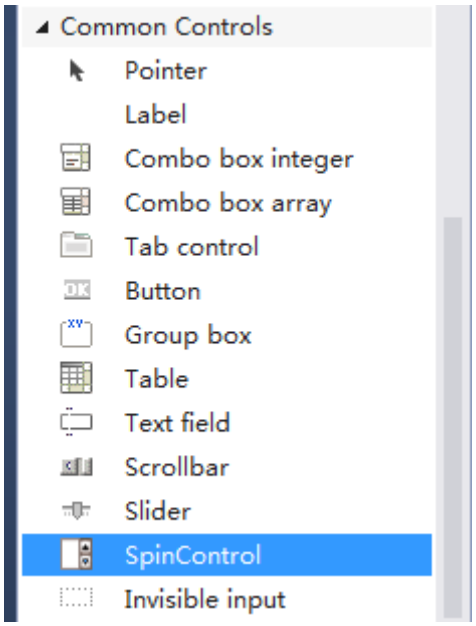
（5） 选中矩形框，在右侧 properties 选项卡下找到 Dynamic texts，在 Text List 右边下拉选项中找到 Errorlist 进行添加



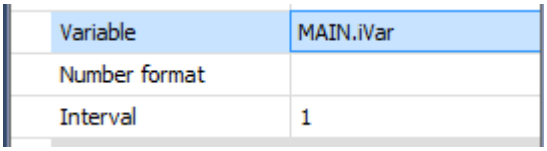
在 Text index 中添加程序中变量 iVar



（6） 选择 Common Controls→SpinControl 进行 iVar 值的切换，iVar 即为 Error ID

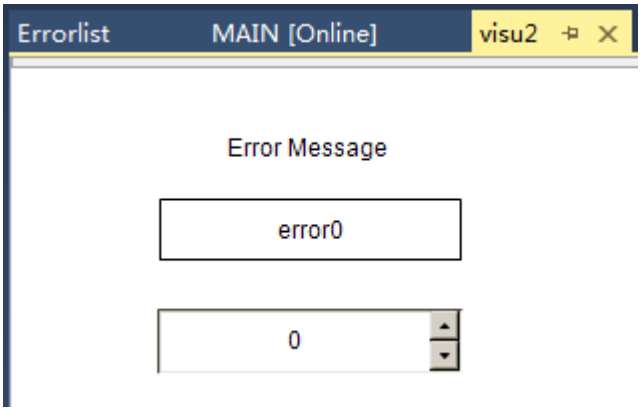


选中这个控件，右侧 properties 选项卡找到 Variable 进行变量链接



（9） 上述操作完成后点击  激活配置，点击登入  再运行  效果如下：

初始状态：

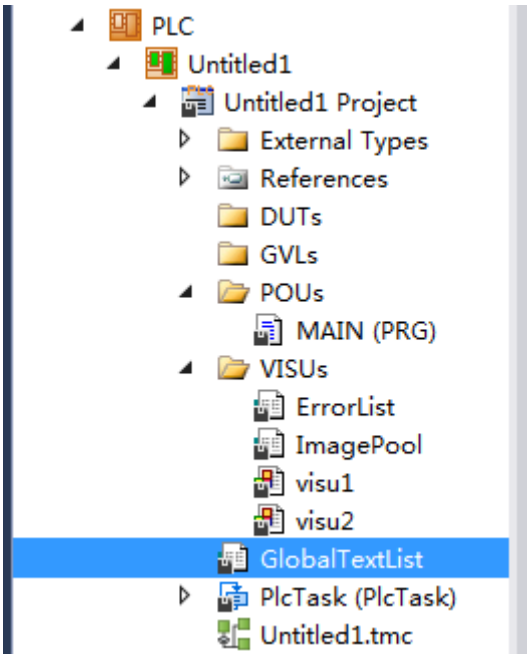


如果对 SpinControl 中的数值进行调节（不超过 ErrorList 中 ID 栏的最大值，不低于最小值），就可以查看到相应的报错信息。

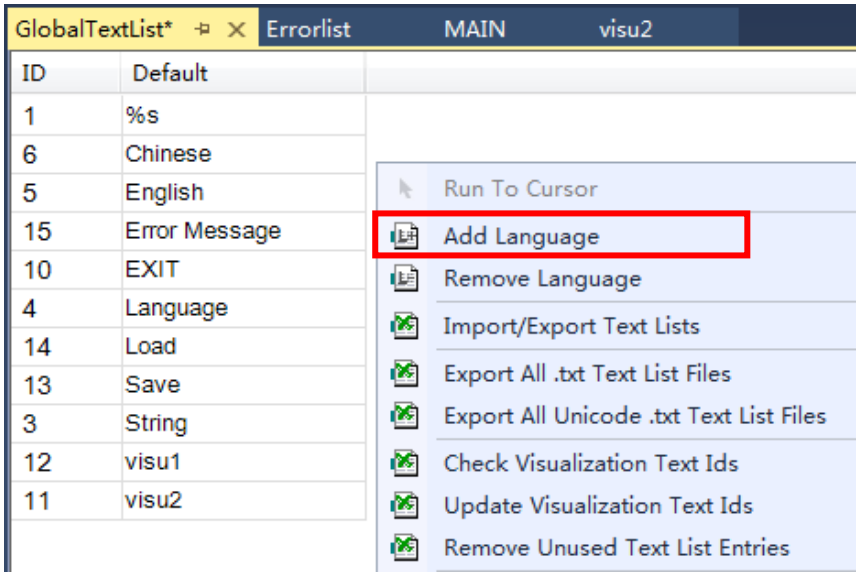
1.9 语言转换

多语言切换同样是通过 InputConfiguration 来实现，在 TwinCAT3 中提供 GlobalTextList，包含在界面中出现的所有的语言文字，并且可以在 GlobalTextList 手动添加语言，对界面中使用的文本进行翻译。

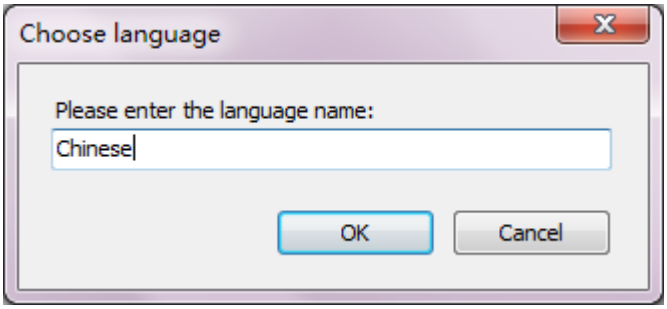
（1） 在左侧管理窗口 VISUs 文件夹下找到 GlobalTextList 这个文件



（2） 双击该文件后会出现之前在 Visualization 中已经编辑过的文字，在空白处右键，在菜单中找到 Add Language，单击后在弹出的对话框中添加语言，点击 OK



输入添加语言名称



在新建语言 Chinese 下对界面中文本进行翻译

GlobalTextList* | Errorlist | MAIN

ID	Default	Chinese
1	%s	
6	Chinese	中文
5	English	英语
15	Error Message	报错信息
10	EXIT	退出
4	Language	语言
14	Load	上载
13	Save	保存
3	String	字符串
12	visu1	界面1
11	visu2	界面2

再新建一个语言 English，输入对应的文本

GlobalTextList* | Errorlist | MAIN | visu2

ID	Default	Chinese	English
1	%s		
6	Chinese	中文	Chinese
5	English	英语	English
15	Error Message	报错信息	Error Message
10	EXIT	退出	EXIT
4	Language	语言	Language
14	Load	上载	Load
13	Save	保存	Save
3	String	字符串	String
12	visu1	界面1	visu1
11	visu2	界面2	visu2

在 ErrorList 中重复以上操作

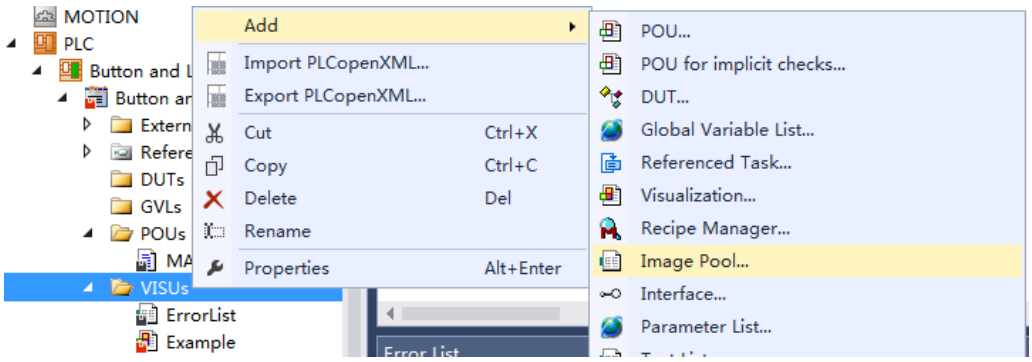
GlobalTextList visu2 | ErrorList | MAIN

ID	Default	Chinese	English
0	error0	错误0	error0
1	error1	错误1	error1
2	error2	错误2	error2
3	error3	错误3	error3
4	error4	错误4	error4
5	error5	错误5	error5

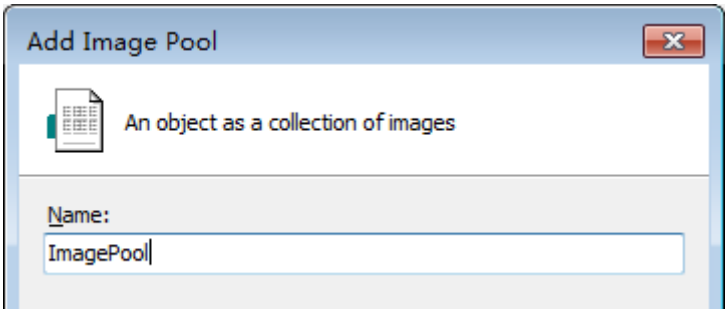
（3） 完成以上操作后需要在界面中通过按钮或者图片进行语言切换，以图片作为语言切换的提示，需要使用 TwinCAT3 HMI 中的 Image Pool

（4） TwinCAT3 中 Image Pool 的使用

在左侧配置管理窗口，找到 VISUs 文件夹，右键 Add，在菜单中选择 Image Pool



在弹出的对话框中命名其为 Image Pool，点击 Open

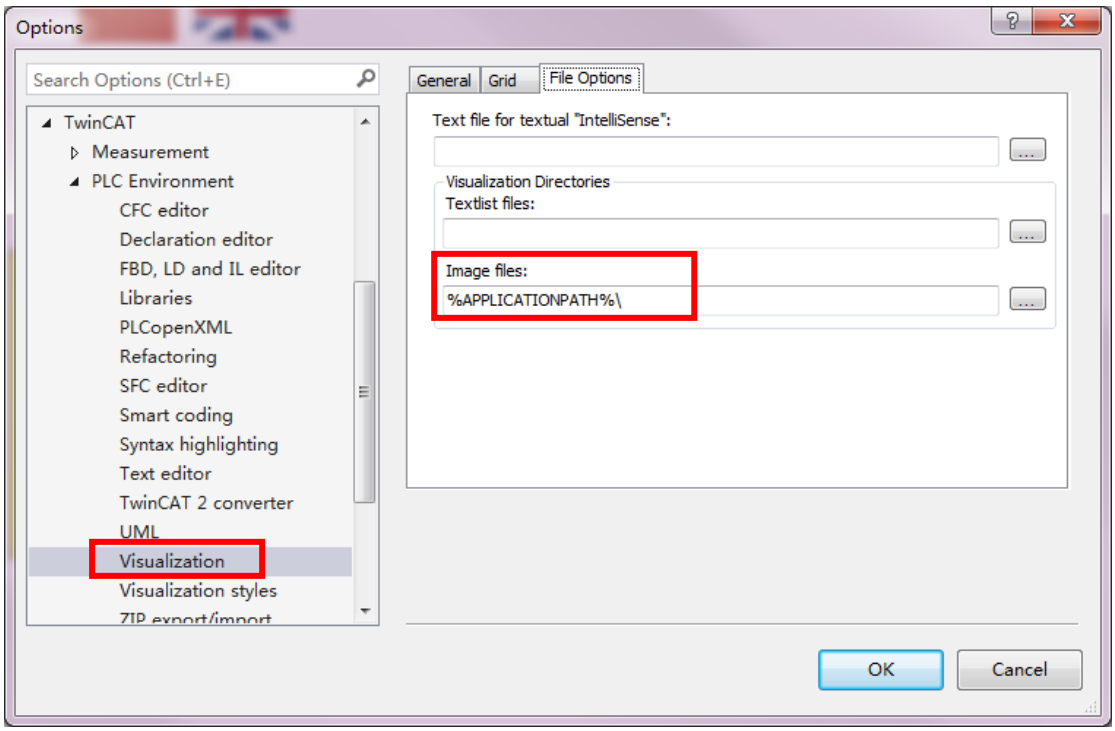


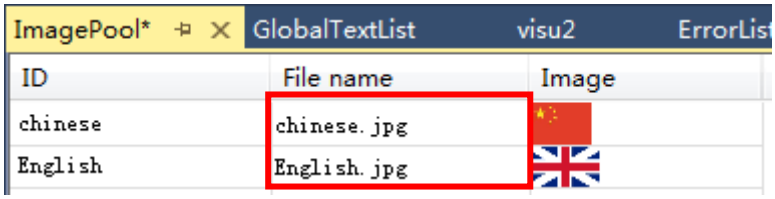
（5） 双击添加好的 ImagePool，在右边的配置窗口中的 File name 一列通过右边的 [...] 添加在界面中会使用到的图标，支持大部分的图片格式

MAIN | ImagePool | Example*

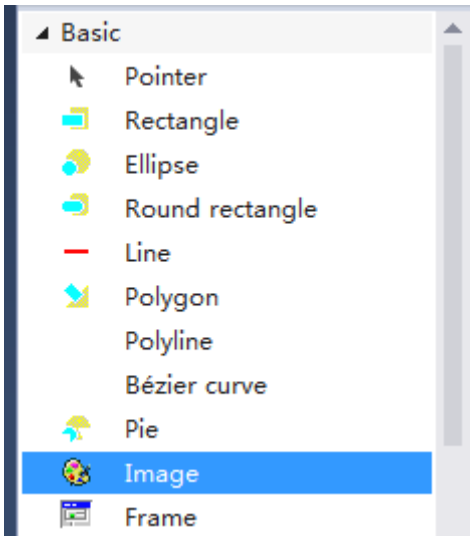
ID	File name	Image	Link type
	[...]		

如果将图片添加到 TwinCAT3 自动创建的项目文件夹中，并且确保菜单栏 Tools→options 中的 Visualization 右边 File Options 下的 Image files 路径为%APPLICATION%，就可以看到添加的图片名称不再带有路径，否则如果改变导入的图片路径会导致画面中的图片无法显示

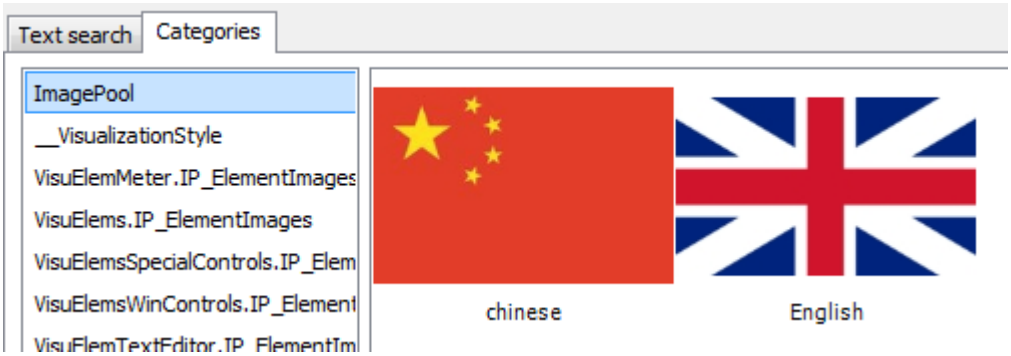




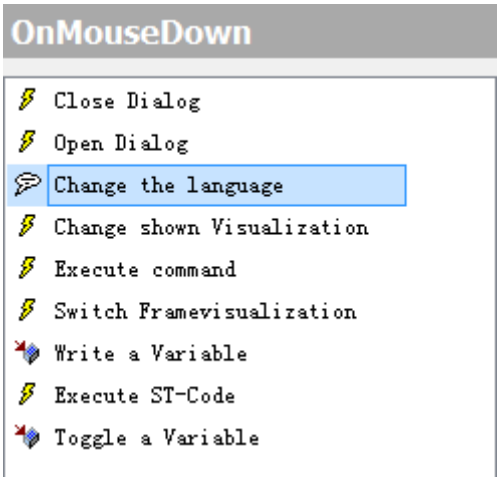
（6） 在画面中添加 Toolbox→Basic→Image 显示图片



将 Image 拖到画面中会自动弹出对话框，在 Categories 选项卡下选择刚刚生成的图库 ImagePool，右侧小窗口会显示图库中的图片，选择需要的图片，点击 OK，在画面上将其调节到合适的大小。

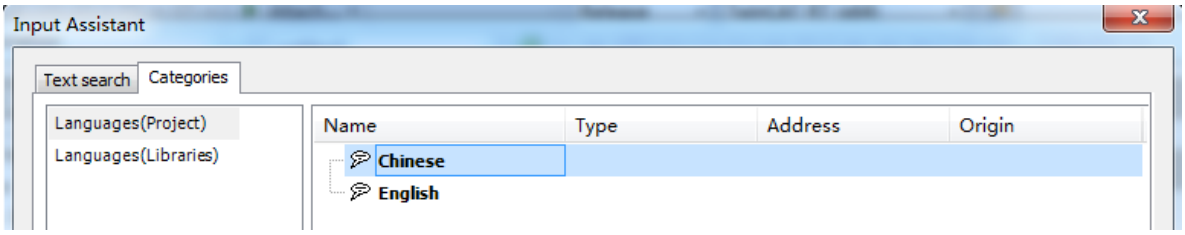


（7） 选中显示中国国旗，在右侧属性栏里将 Inputconfiguration 展开，选择 On MouseClick，单击 Configure…后在弹出的对话框左侧选项中双击 Change the language



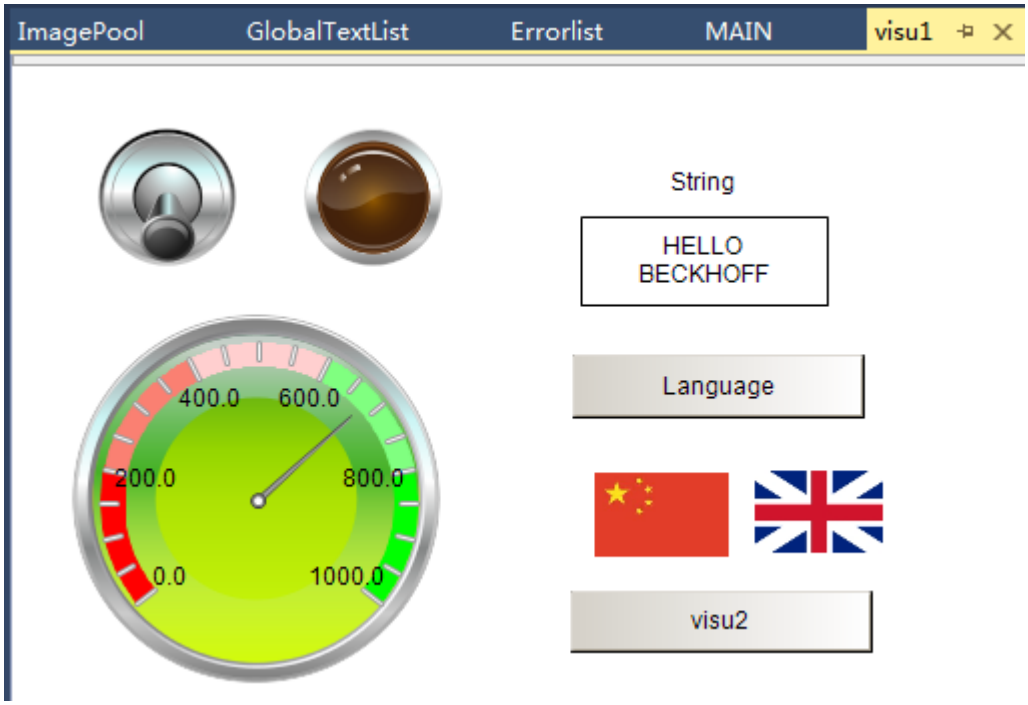
在右侧属性栏点击 Language 下空白处右边的小按钮

在弹出的对话框中选择之前创建的语言，选中 Chinese，点击确定



（8） 同理，选中显示英国国旗，在右侧属性栏里将 Inputconfiguration 展开，单击 Configure 后在弹出的对话框左侧选项中双击 Change the language，在右侧属性栏点击 Language 下空白处右边的小按钮，在弹出的对话框里选中 English 点击确定

（9） 上述操作完成后，激活并运行程序，效果如下：

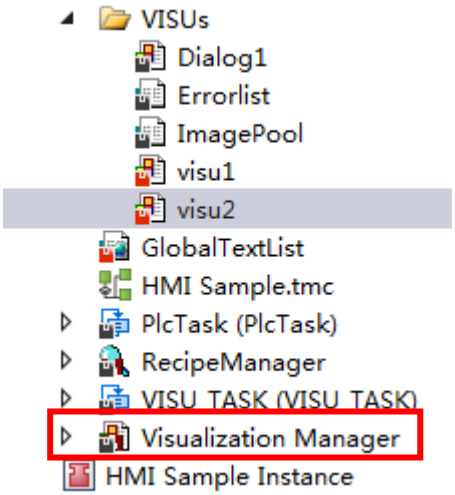


点击中国国旗，切换到中文

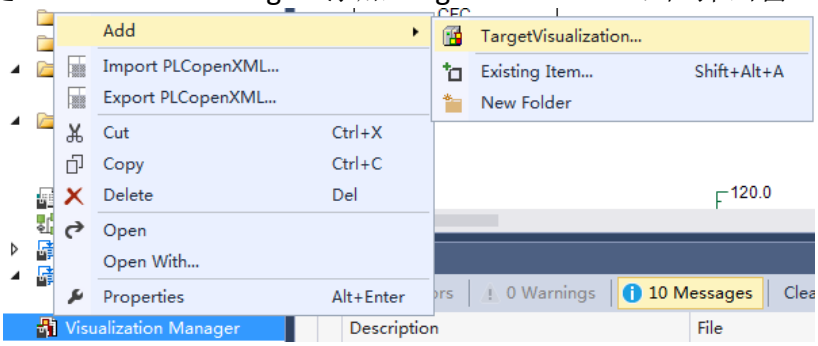
2 TwinCAT3 PLC HMI 在 XP 和 WIN7 系统中的全屏显示方法

2.1 全屏显示方法

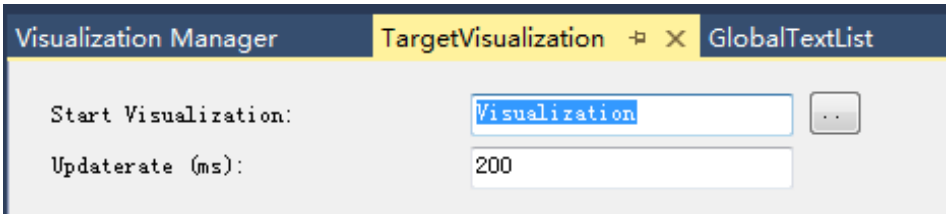
（1） 创建一个 Visualization，会自动添加 Visualizaton manager



（2） 右键 Visualization manager 添加 TargetVisualization，在弹出窗口中点击 OK



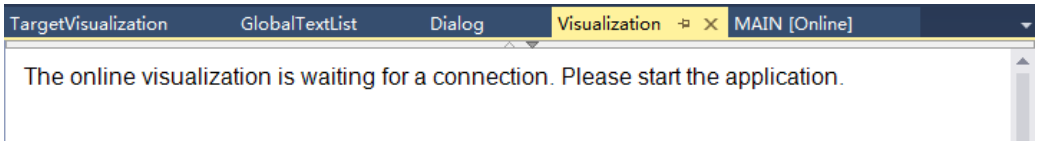
（3） 在这里可以设置起始画面



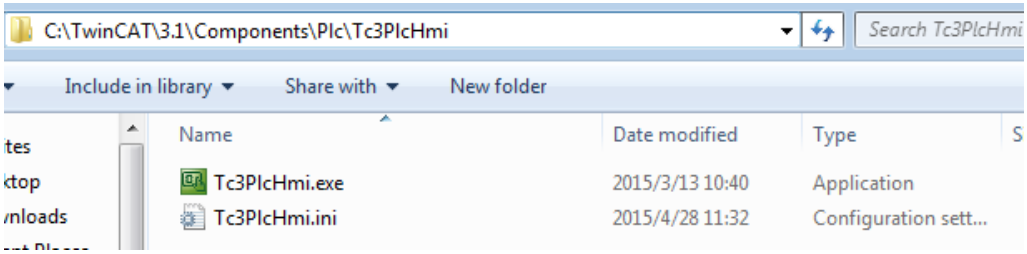
（4） 当然还需要激活相应的 license（注意这个 license 只需要激活在 PLC 所在的 runtime 中即可，如果一台 IPC 远程显示一台 EPC 上的画面，那只需要 EPC 上有 HMI-license，IPC 不需要 HMI-license）。

TF1800	TC3 PLC-HMI	cpu license

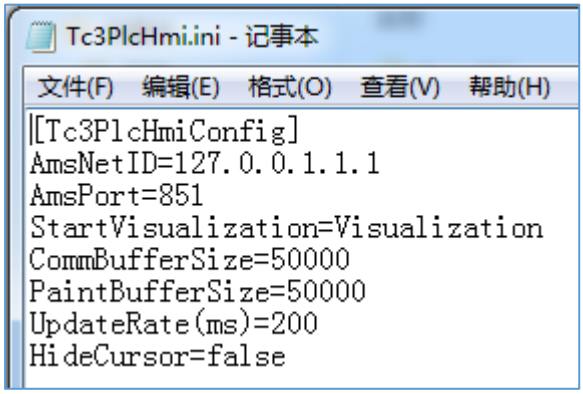
（5） 一般只需要点击 Activate Configuration 激活配置，系统会自动勾选项目需要使用的 License，按照要求输入验证码后，点击 Login 登录出现以下画面



（6） 点击运行后，找到路径：C:\TwinCAT\3.1\Components\Plc\Tc3PlcHmi.exe



（7） 双击打开 Tc3PlcHmi.ini 文件进行设置并保存



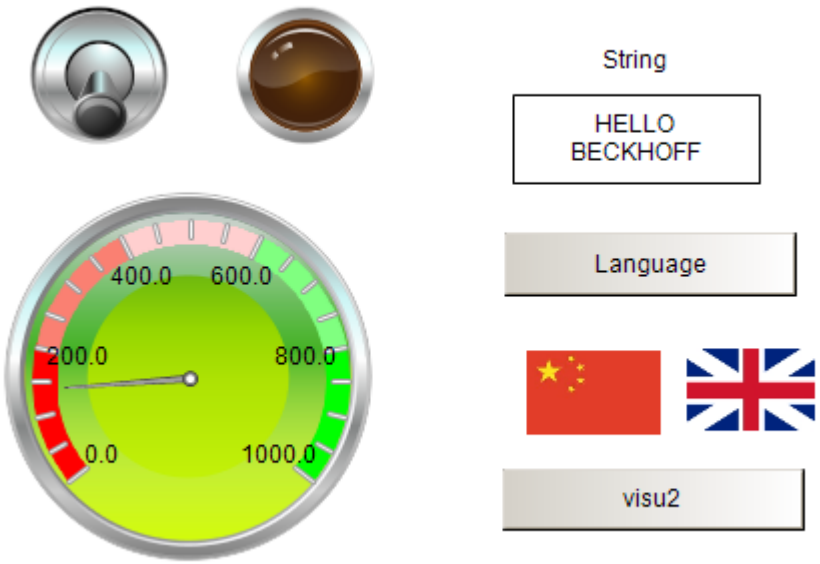
AmsNetID：可视化界面全屏运行所在目标控制器的 Ams 地址

AmsPort：可视化界面所属 PLC 工程的目标控制器 AMS 端口号

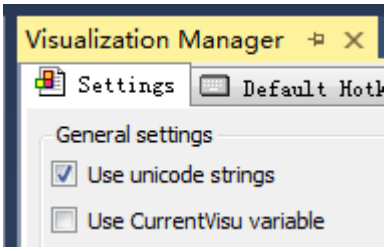
StartVisualization：起始画面设置（实际以 TargetVisualization 的设置为准）

UpdateRate：画面更新周期

（8） 运行 Tc3PlcHmi.exe 就可以在本地全屏显示了



（9） 双击 Visualization Manager，勾选 Use Unicode strings，否则界面上的中文字符会显示成乱码

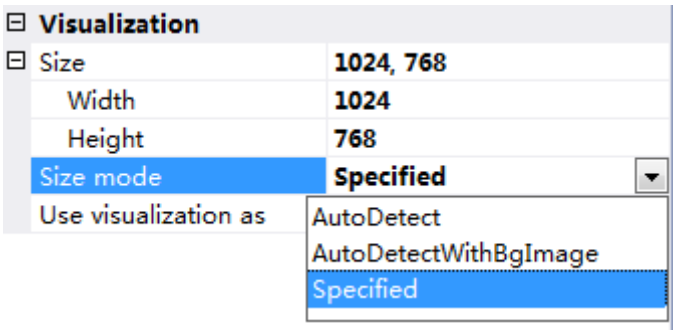


（10） 分辨率的修改

每一台控制器的分辨率不同，需要调整分辨率适应屏幕。

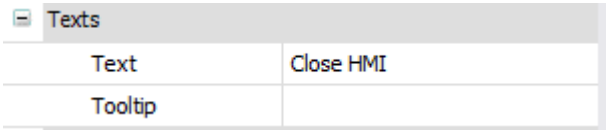
左侧配置管理窗口，单击选中需要更改分辨率的 visualization 文件，右侧会显示该文件属性。

找到 Visualization 标签，展开找到 Size 标签展开，在 Size mode 旁边的下拉菜单中选择 Specified，而后就可以更改该画面的分辨率了

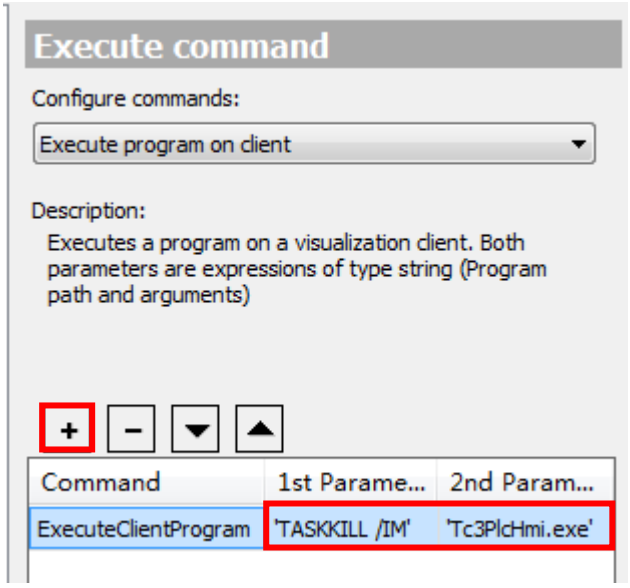


（11） HMI 全屏画面的退出（不针对 CE 系统）

全屏显示中的画面可以通过 Alt+F4 退出，也可以通过在画面中添加相应控件实现，以下以按钮为例，在画面中添加 Toolbox→Common Controls→Button，选中 Button，在右边的属性菜单中输的 Texts→Text 中输入 Close HMI



在 InputConfiguration 中选择 OnMouseClick，在弹出的对话框左边选择 Excute command，在右边的配置窗口中的 Configure commands 下拉选项中点击 Excute program on the client，并且按照选项下方对于这条指令的提示，点击‘+’添加指令描述，填写 Parameter 中的参数，分别填入‘TASKKILL/IM’和‘Tc3PlcHmi.exe’

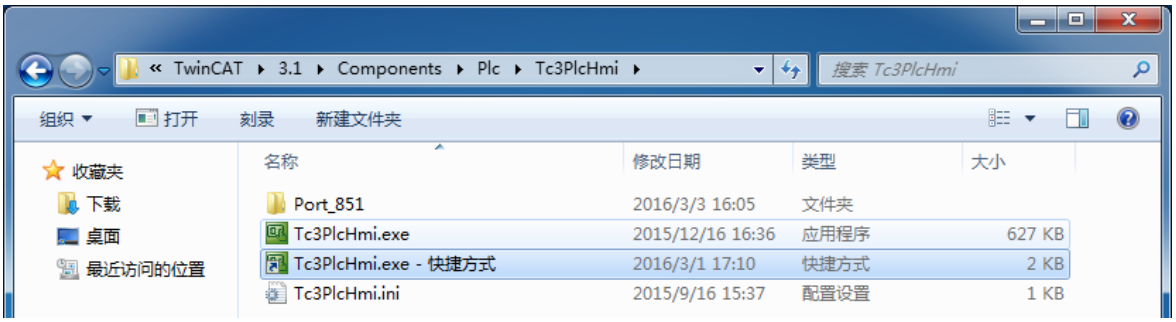


点击 OK 应用配置，激活并运行程序，全屏显示后点击 Close HMI 按钮即可退出全屏

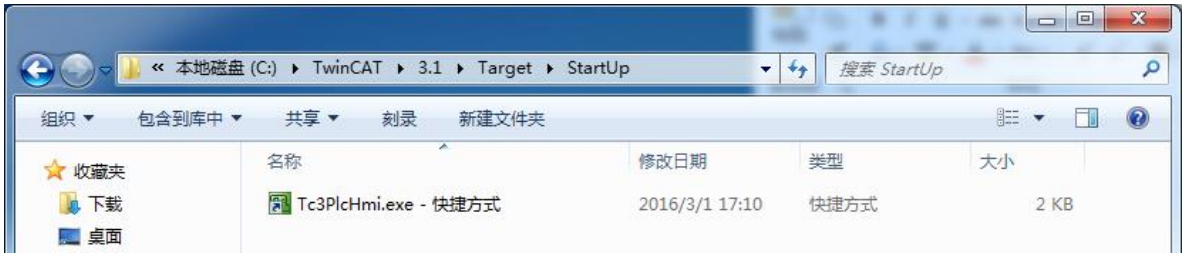
2.2 XP 和 WIN7 系统中 PLC HMI 开机自启动

对于 Boot Project 的生成和设置，具体请参照本书第九章。这里主要集合了在目标设备上的相关设置。

（1） 找到 C:\TwinCAT\3.1\Components\Plc\Tc3PlcHmi 中的 Tc3PlcHmi.exe 这个应用程序，创建快捷方式



（2） 把这个快捷方式放到 C:\TwinCAT\3.1\Target\StartUp 这个文件夹里

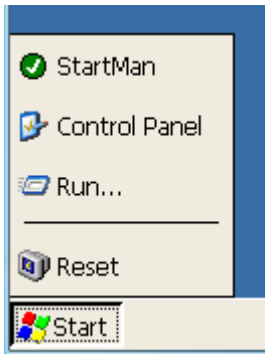


3 在 CE 操作系统上的全屏显示方法

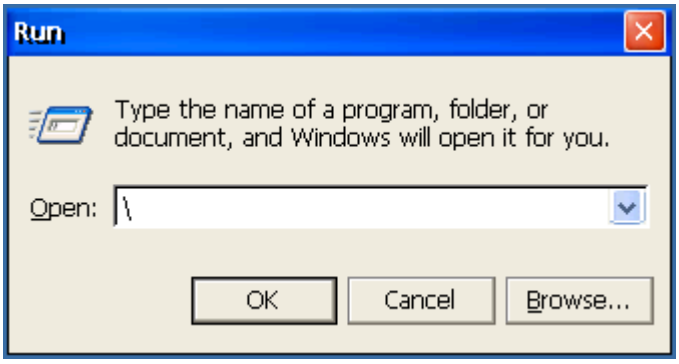
3.1 全屏显示方法

（1）基本操作参照之前 XP 或 Win7 全屏显示方法（1）——（7），只需注意在 CE 系统中，Tc3PlcHmi.exe 这个应用程序的打开方式如下：

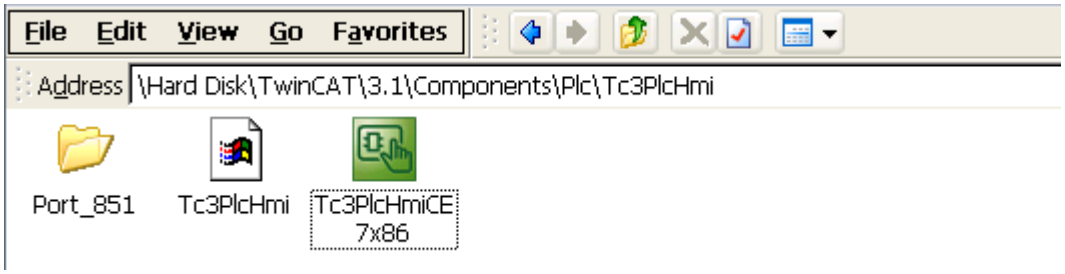
点击 Start→Run…



在弹出的对话框，Open 旁的空白栏里键入反斜杠符号 \ ，点击 OK。



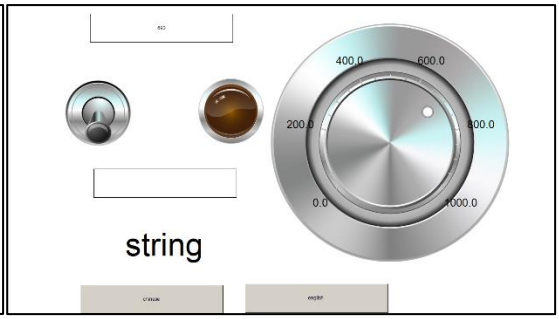
在 \Hard Disk\TwinCAT\3.1\Components\Plc\Tc3LpcHmi 这个目录下，找到 Tc3PlcHmiCE7x86 这个应用程序，双击打开就可以实现全屏显示了



3.2 全屏显示常见问题

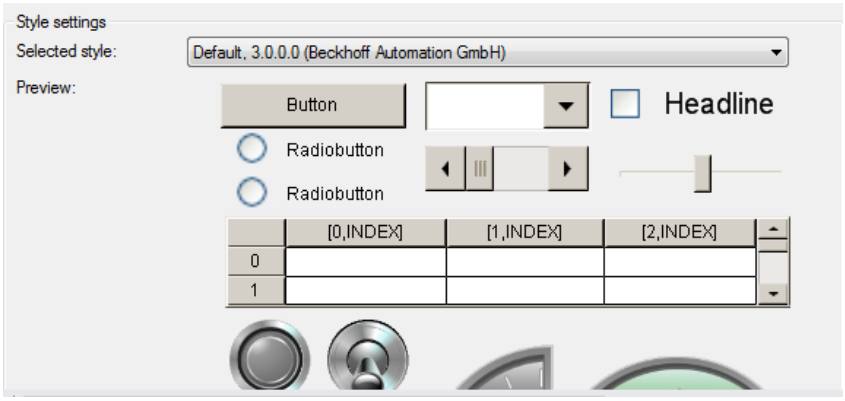
在 CE 操作系统上进行 HMI 的全屏显示可能会遇到以下问题：

（1）画面显示不完整



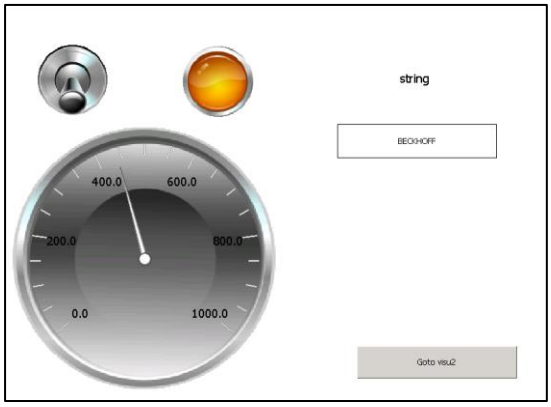
不完整画面　　完整画面

需要注意 CE 中显示的 HMI，其 style 选择必须是 Default,3.0.0.0



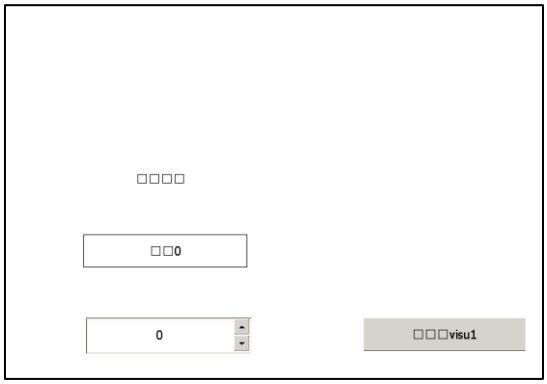
（3） 在 CE 中显示 HMI，所有控件都不支持 gradient color、transparency 和 optimized drawing，否则控件会显示不正常。

（3）如果发现使用 Image 添加的图片无法显示，这是因为 CE 系统不支持 JPG 格式的图片，建议将其保存成 BMP 格式，再重新加入 Image Pool，将 Image 的图片重新添加一下。



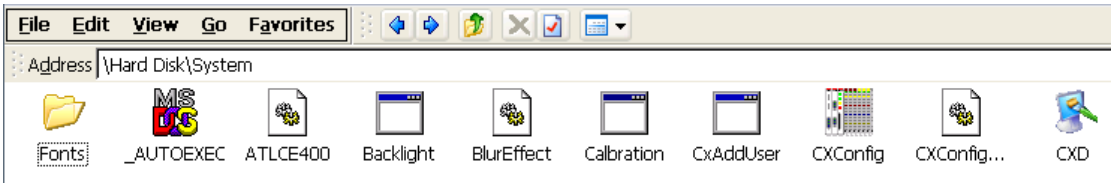
图片为 JPG 格式时的画面　　图片为 BMP 格式时的画面

（4）如果发现中文字符无法正常显示，如下图显示的是□□□，这是因为 CE 系统中缺少中文字库。

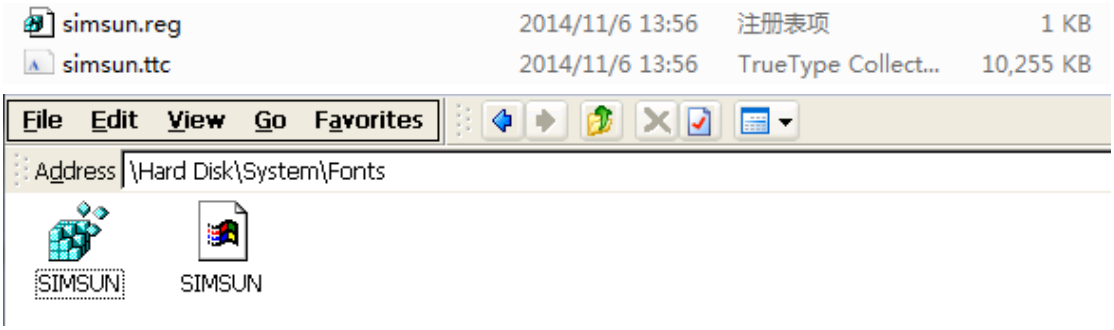


更新办法：

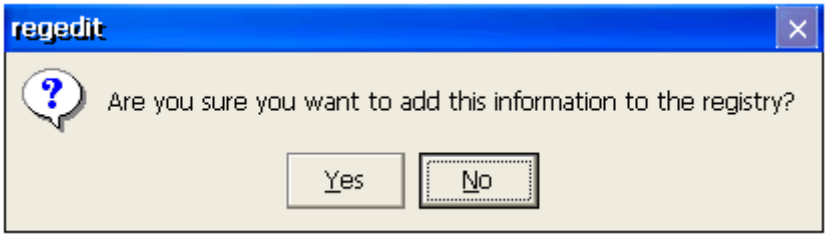
- 在设备的 Hard Disk\System 目录下创建名称为 Fonts 的文件夹。



- 将相应的字体文件和注册表文件复制到这个文件夹（演示使用的是宋体）

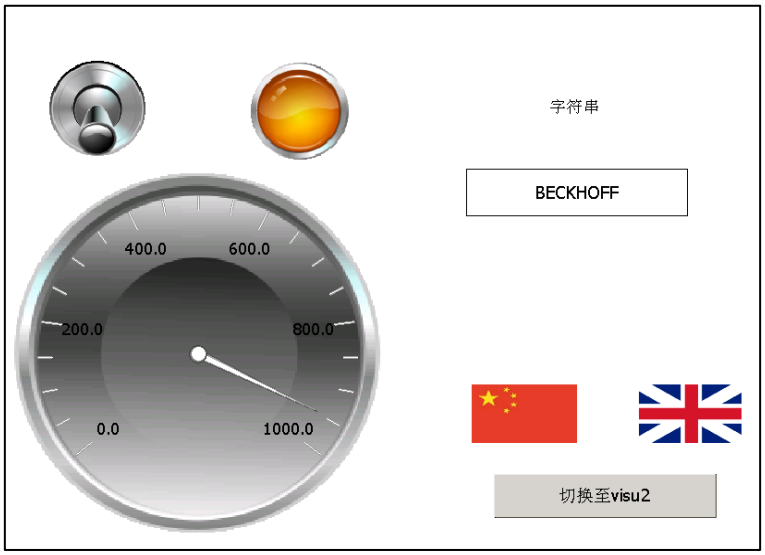


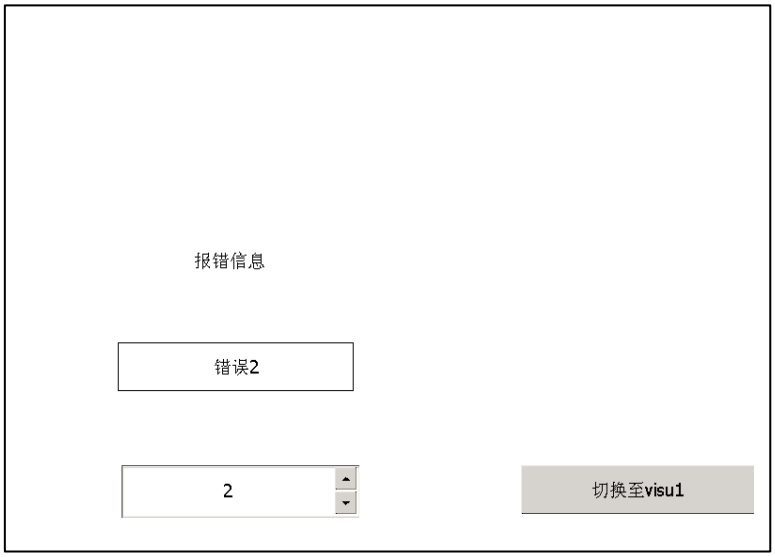
- 双击注册表文件，导入注册值，重新启动系统即可



- 出现上述对话框，询问是否导入注册值，点击 Yes。

- 之后，跳出上述对话框，询问是否进行重启以将设置应用到设备，点击 Yes，设备自动重启。
  然后再进行全屏显示操作，就能发现中文字符都可以正常显示了

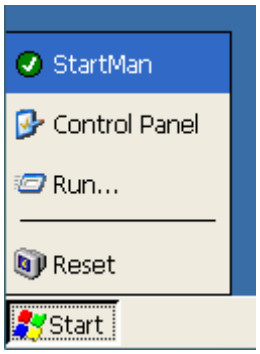




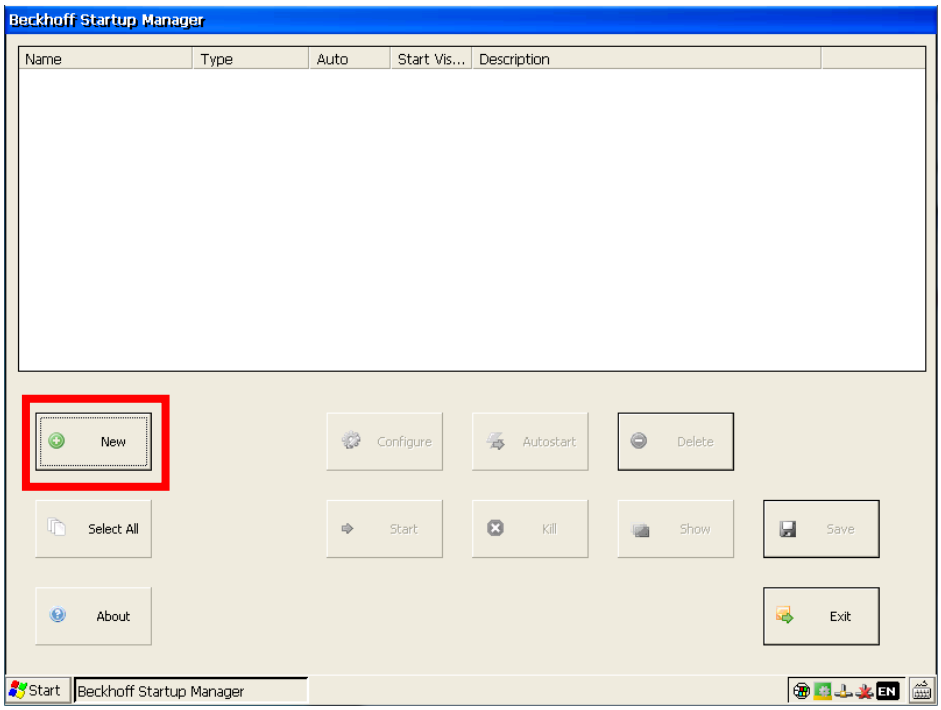
3.3 CE 系统中的开机自启动

对于 Boot Project 的生成和设置，具体请参照本书第九章。这里主要集合了在目标设备上的相关设置。

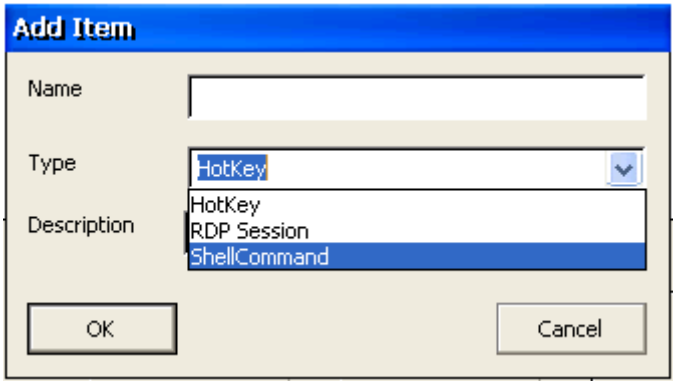
（1）点击 Start→StartMan（Startup Manager）



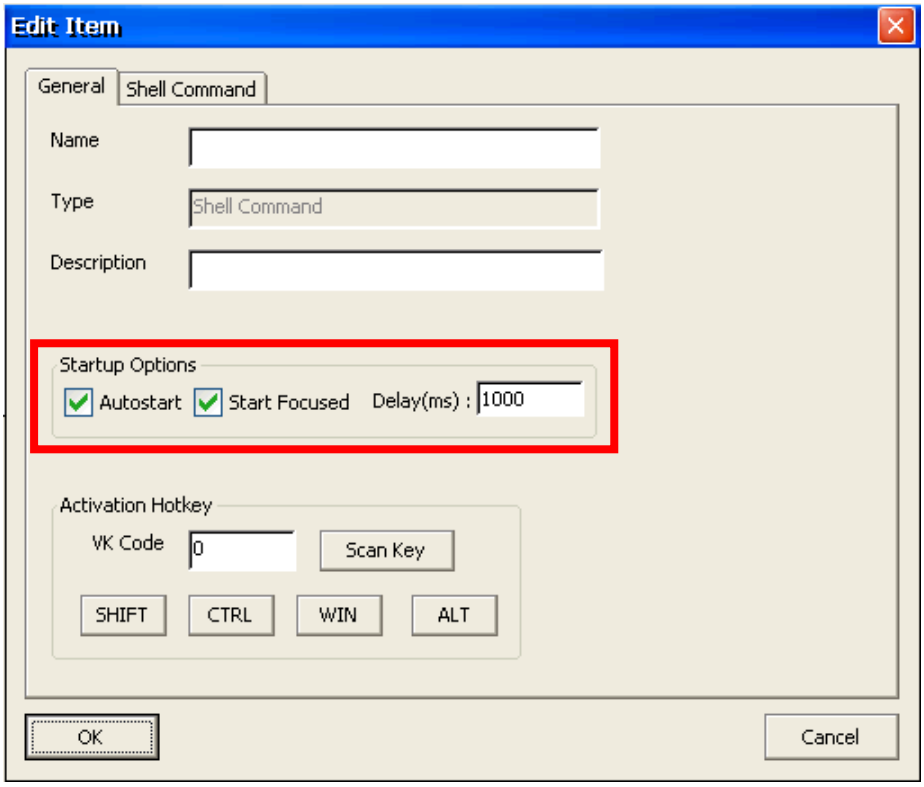
（2）需要在跳出的对话框里进行启动项的添加，这里点击 New 新建



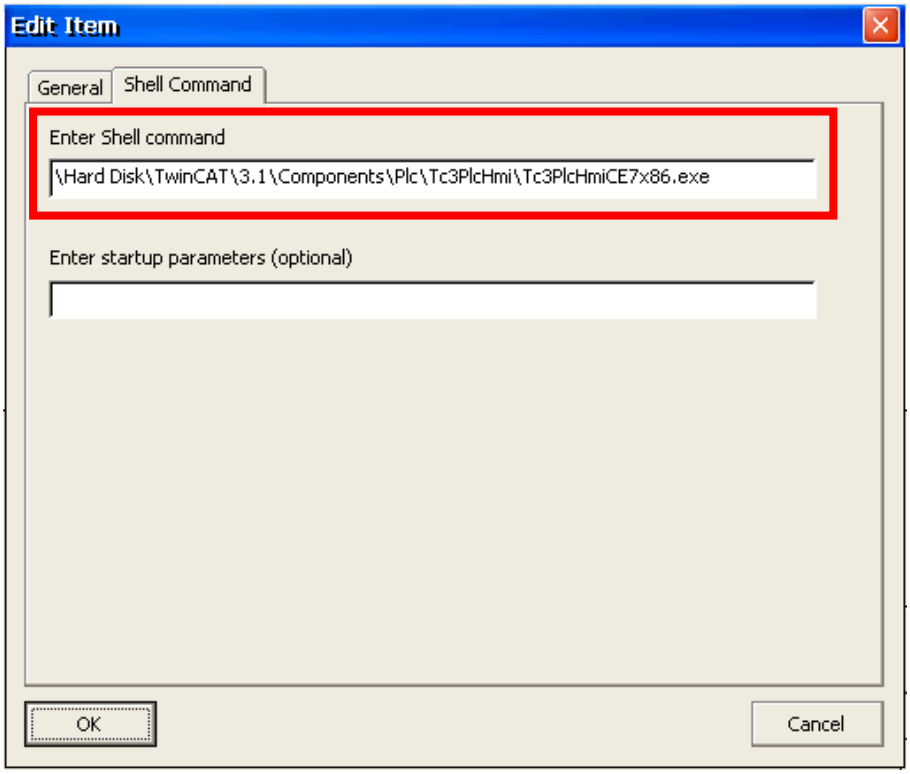
（3）在跳出的 Add Item 这个对话框中，点击 Type 旁边的下拉菜单，选择第三项 ShellCommand。Name 和 Description 不用填写，点击 OK。



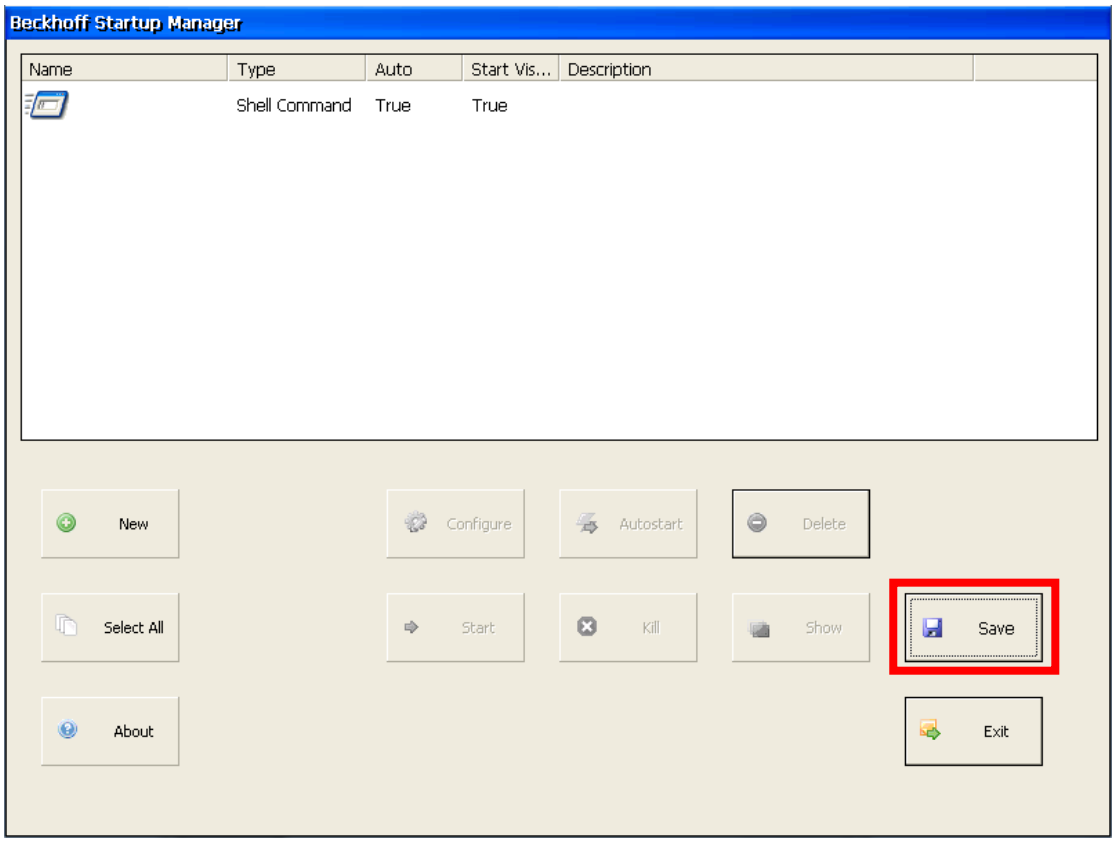
（4）接着跳出 Edit Item 对话框，在第一个选项卡 General 里，注意勾选 Startup Options 下的 Autostart 和 Start Focused 这两个选项，并且设置开机后延时多长时间再自启动，在 Delay（ms）旁边的空白栏写入期望的延时时间。这里设置 1000ms。**注意：不建议延时时间设置为 0！**



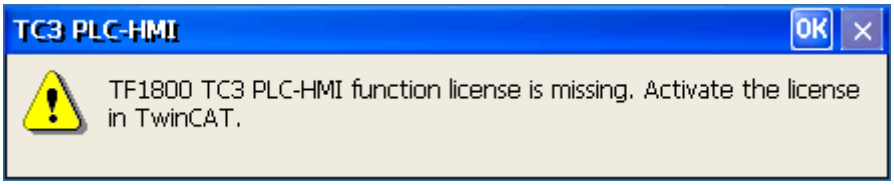
（5）再到 Shell Command 这个选项卡里，在 Enter Shell command 下的空白栏写入路径 \Hard Disk\TwinCAT\3.1\Components\Plc\Tc3PlcHmi\Tc3PlcHmiCE7x86.exe，点击 OK。



（6）点击 Save 保存，然后退出、重启，就可以实现自启动了。



（7）如果开机自启动时，出现如下报错而无法正常运行，可能是由于开机自启动延时时间设置过短造成的。



之前说到，不建议延时时间设置为 0ms。例子中设置为 1000ms，但是在实际工程中，所需延时会因为目标控制器的型号或是程序的大小而有所不同。这在实际工程中需要自行测试合适的延长时间。

4 TwinCAT3 Recipe 功能的实现

配方（Recipe）是一组参数值，它用于提供生产产品和控制生产过程所需的信息，结合 TwinCAT3 提供的 Recipe Manager 和 PLC HMI 就可以在界面当中实现配方的功能。

（1） 首先编写示例程序，为了测试 TwinCAT3 中的 Recipe 支持的数据类型，以及是否对变量的作用域有要求，我们分别创建多种数据类型的全局变量和局部变量，在 Main 程序中定义局部变量如下：

MAIN* GVL

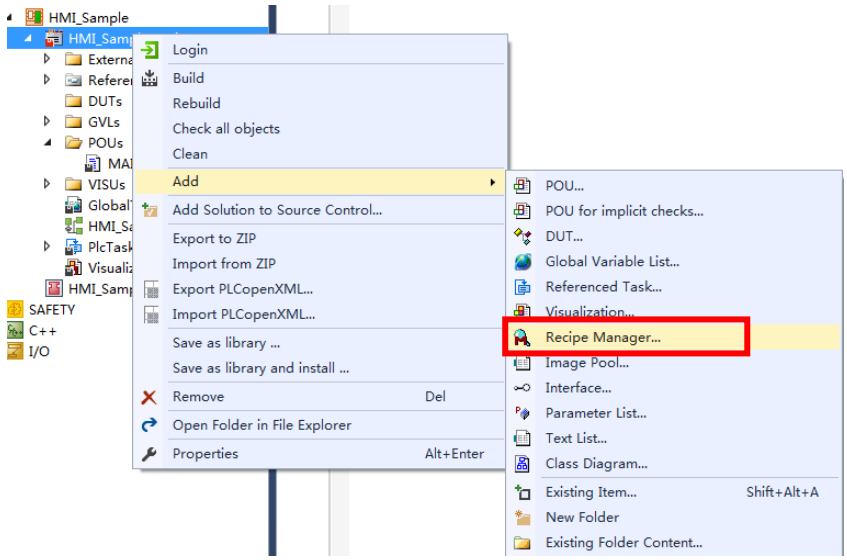
```
PROGRAM MAIN
VAR
    bRecipe:        BOOL;
    nRecipe:        INT;
    rRecipe:        REAL;
    aRecipe:        ARRAY[1..10] OF INT;
    sRecipe:        STRING;
    wsRecipe:       WSTRING;
    TON_Recipe:     TON;
END_VAR
```

在 GVL 文件夹右键→Add→Global Variable List→OK，使用默认 GVL 创建，在新建的 GVL 中定义全局变量如下如下：

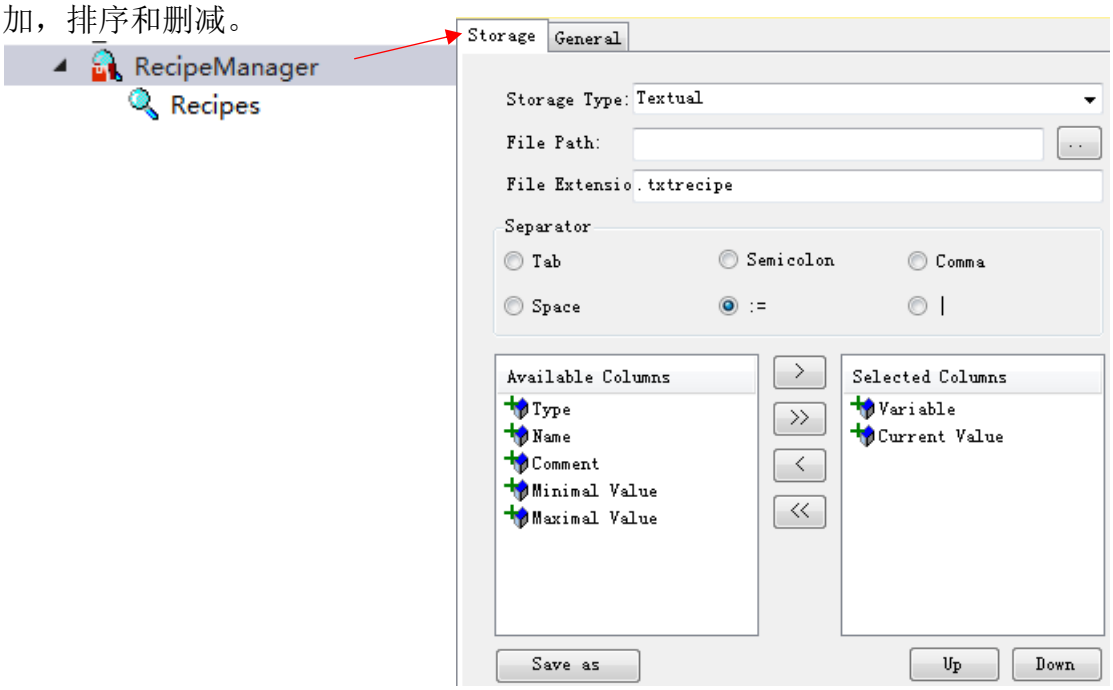
MAIN* GVL*

```
{attribute 'qualified_only'}
VAR_GLOBAL
    bRecipe:        BOOL;
    nRecipe:        INT;
    rRecipe:        REAL;
    aRecipe:        ARRAY[1..10] OF INT;
    sRecipe:        STRING;
    wsRecipe:       WSTRING;
    TON_Recipe:     TON;
END_VAR
```

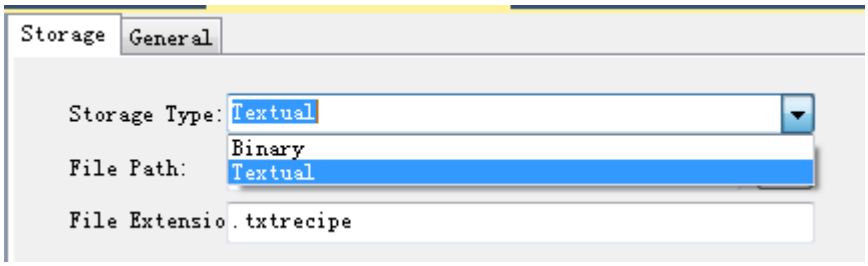
（2） 创建 Recipe Manager，在 PLC 项目上面右键→Add→Recipe Manager，使用默认的名称，点击 open 创建



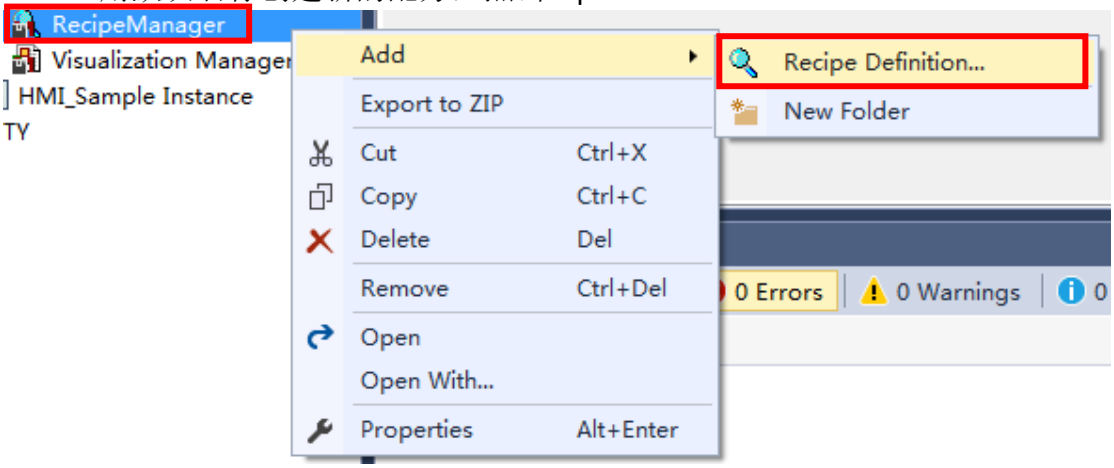
双击创建好的 Recipe Manager 在右边的配置窗口中可以对配方文件的保存格式、路径，文件扩展名进行设置，同时对于配方表格中需要显示的列进行添加，排序和删减。



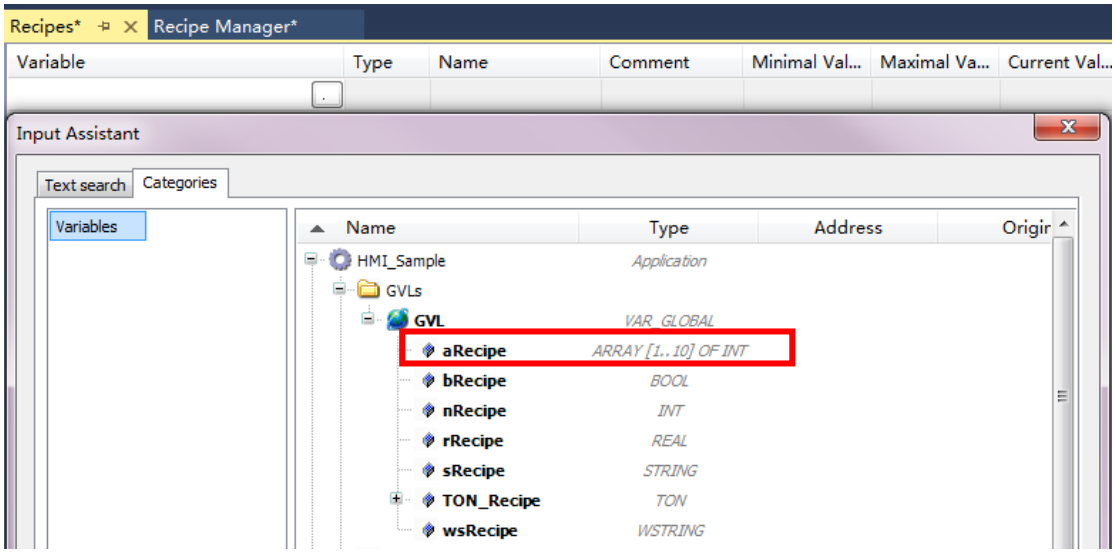
配方文件的保存格式有两种:使用默认的 txt 格式，方便以后对配方文件进行读取、修改等操作；使用 bin 格式，则无法使用其他软件读取、修改保存的配方文件。



（3） 配置完成后，在新建的 Recipe Manager 上右键→Add→Recipe Definition…使用默认名称创建新的配方，点击 Open



（4） 选择新建的 Recipes，在右边的配置窗口中添加配方中所包含的变量。双击表格中 Variable 一列下单元格，通过 F2 或者 . 调用程序中变量，选择全局变量中的 aRecipe。



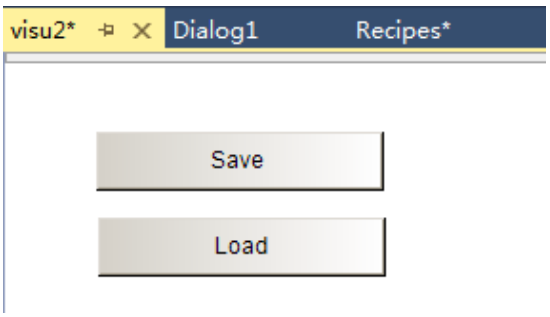
添加完成后，可以在 Recipes 看到全局变量中添加进来的数组变量；根据配方功能继续添加局部变量，由图可见，局部变量也可以添加，功能块，结构体等数据类型都可以添加到配方中，非常方便。

Recipes* Recipe Manager*

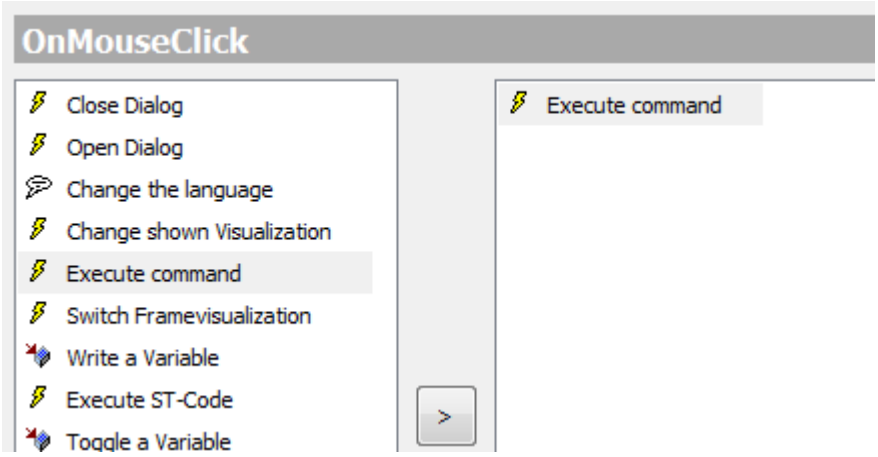
Variable	Type
GVL.aRecipe[1]	INT
GVL.aRecipe[2]	INT
GVL.aRecipe[3]	INT
GVL.aRecipe[4]	INT
GVL.aRecipe[5]	INT
GVL.aRecipe[6]	INT
GVL.aRecipe[7]	INT
GVL.aRecipe[8]	INT
GVL.aRecipe[9]	INT
GVL.aRecipe[10]	INT

MAIN.bRecipe	BOOL
MAIN.nRecipe	INT
MAIN.rRecipe	REAL
MAIN.sRecipe	STRING
MAIN.TON_Recipe.IN	BOOL
MAIN.TON_Recipe.PT	TIME
MAIN.TON_Recipe.Q	BOOL
MAIN.TON_Recipe.ET	TIME
MAIN.TON_Recipe.M	BOOL
MAIN.TON_Recipe.StartTime	TIME
MAIN.wsRecipe	WSTRING

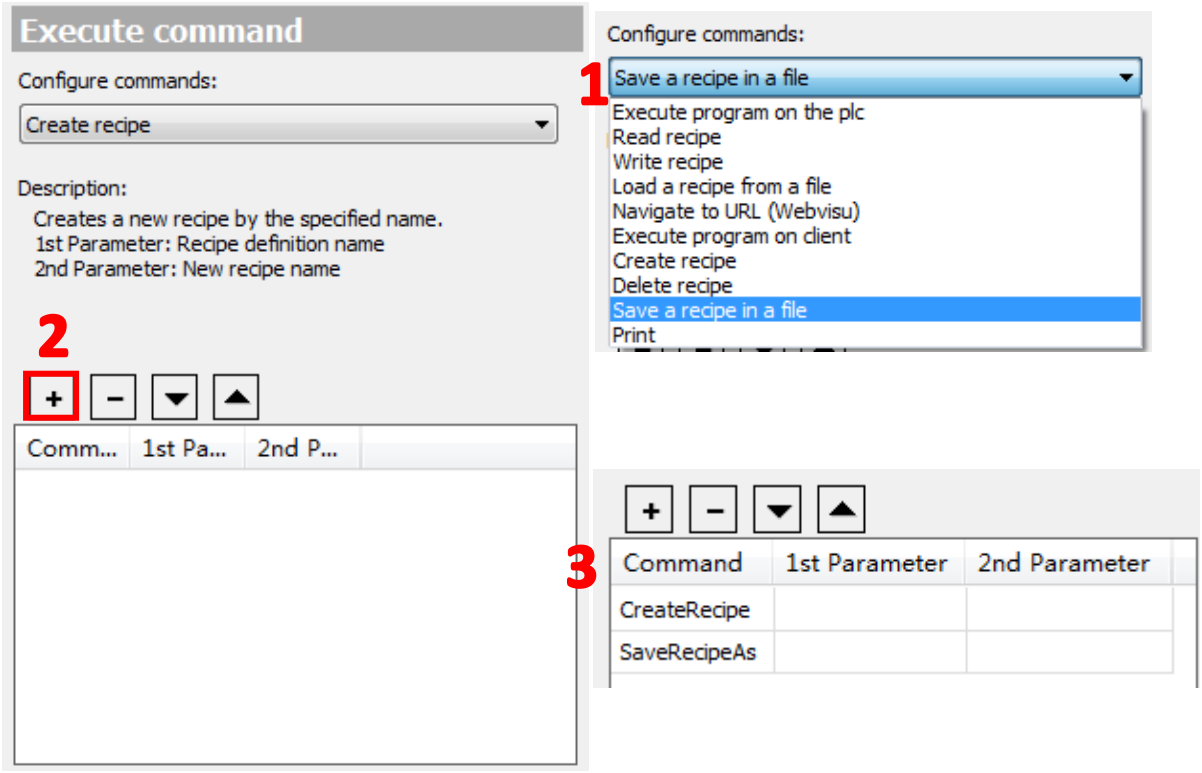
（5） 完成变量的添加后，创建新的 PLC HMI 画面 visu2，在 Toolbox 中找到 Common Controls→Button，添加两个 Button，一个实现 Recipe 的创建和保存，一个实现 Recipe 的调用，选中 Button，在右边属性选项卡中找到 Texts，分别输入 Save 和 Load



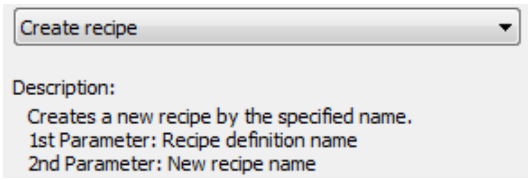
（6） 选中 Save 按钮在属性菜单中找到 Inputconfiguration→OnMouseClick，单击 Configure…对 HMI 中的 Recipe 功能进行配置，在弹出对话框左侧双击添加 Execute command



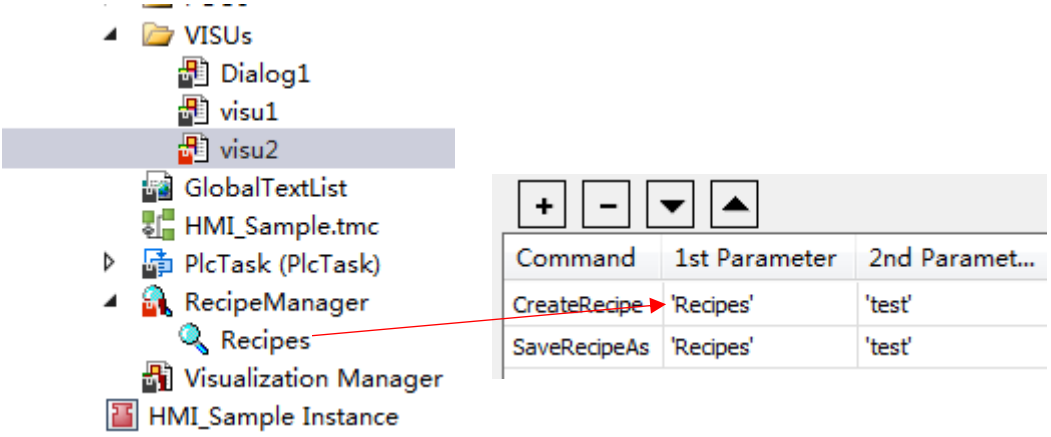
选中 Execute command 在弹出窗口的右边 configure command 下拉选项中选择 Creat recipe，点击下方的 + 加这条指令，同样的添加 Save a Recipe in a file 这条指令



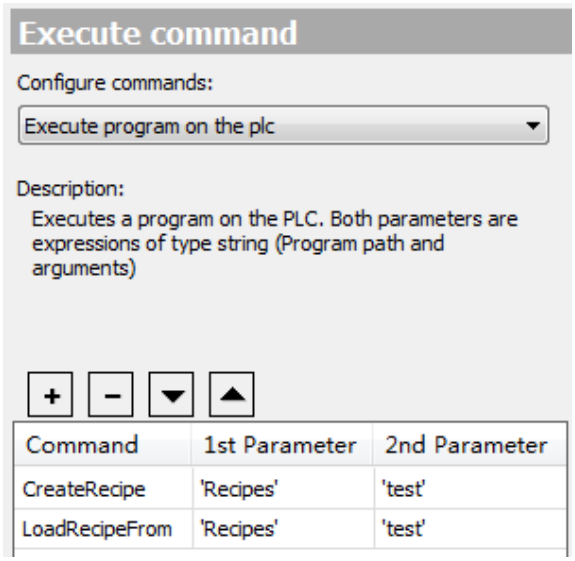
添加完成指令后，需要在 1st Parameter 和 2nd Parameter 中分别填写 Recipe Definition 和 Recipe 的名字，在添加指令的时候同样可以看到相关注释说明



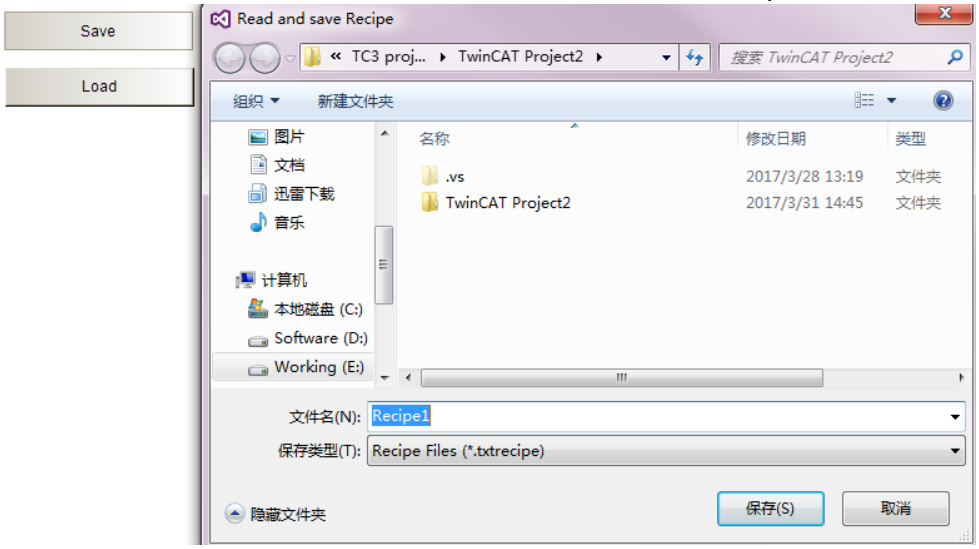
填入之前添加的 Recipe Definition 的名称’Recipes’以及自定义的配方名称’test’，记得在为填写的名称加上单引号



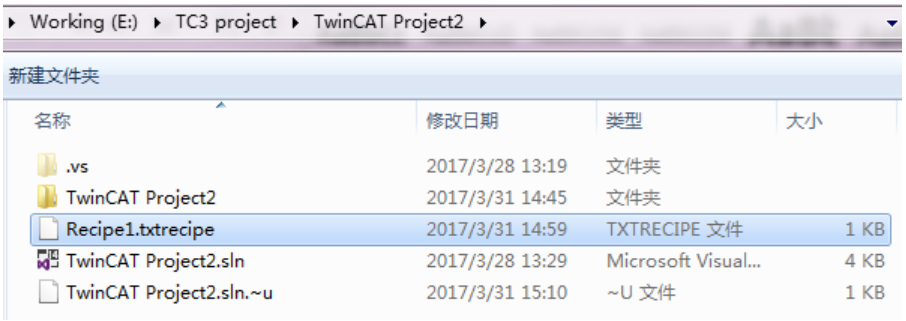
同样的，对 Load 按钮进行类似配置，添加 Creat Recipe 和 Load a Recipe from a file 两个指令，并完成配置，点击 OK 应用



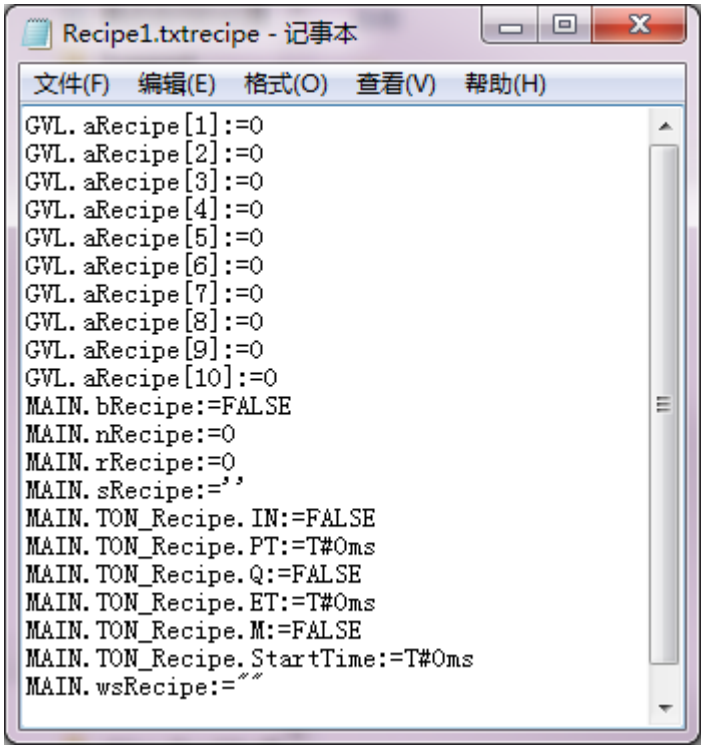
（7） 激活并运行程序，点击画面中的 Save 按钮，可以看到画面中会弹出配方的保存对话框，选择保存路径，在文件名中填入 Recipe1 后点击保存



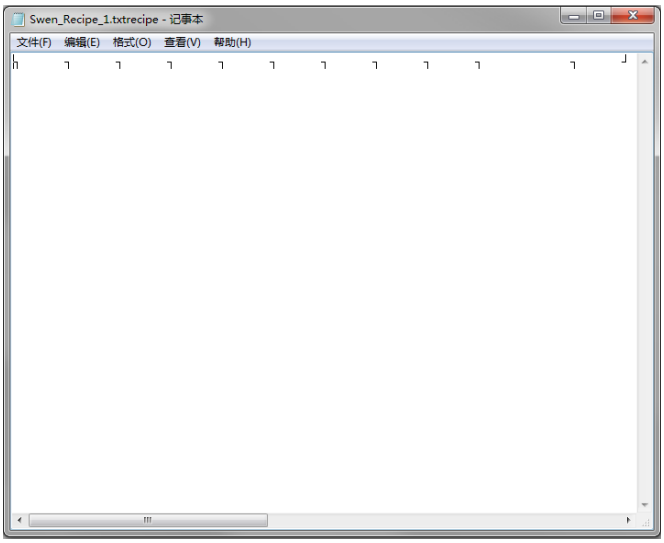
可以得到配方保存的文件：



这个文件可以用记事本打开，并且可以查看里面的数据

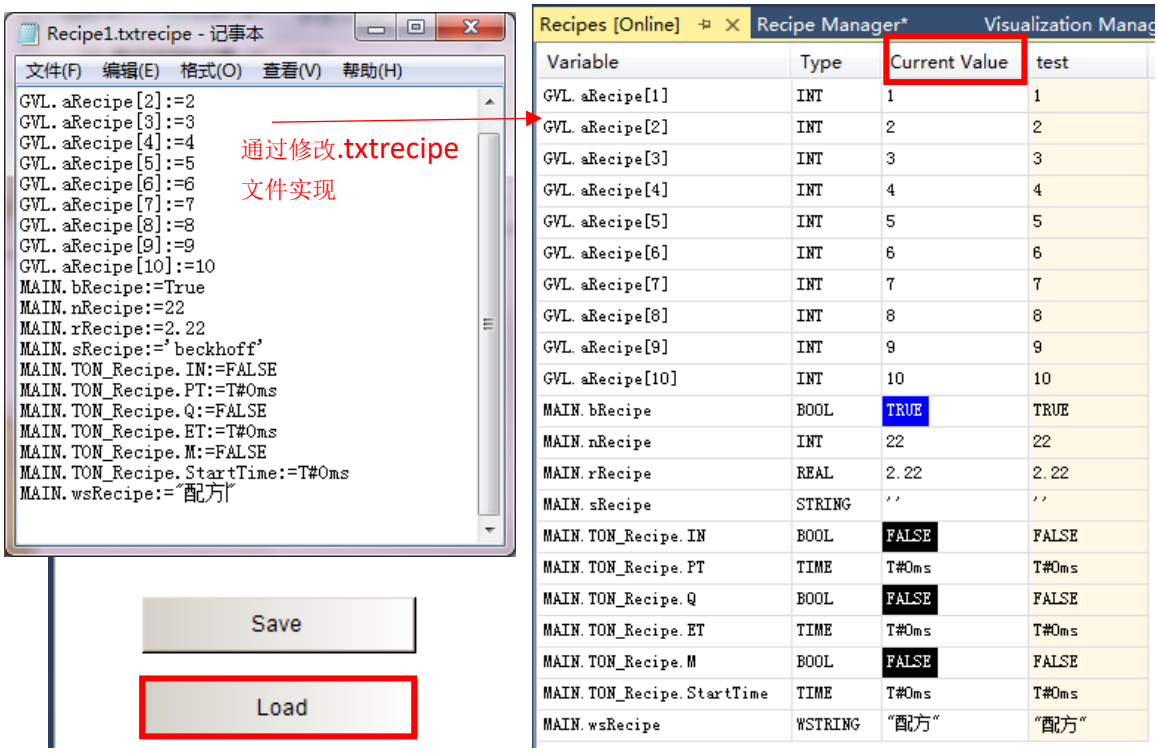


如果在选择保存文件格式的时候选择 bin 的方式，则不能查看配方中的数据，显示为乱码，如下：

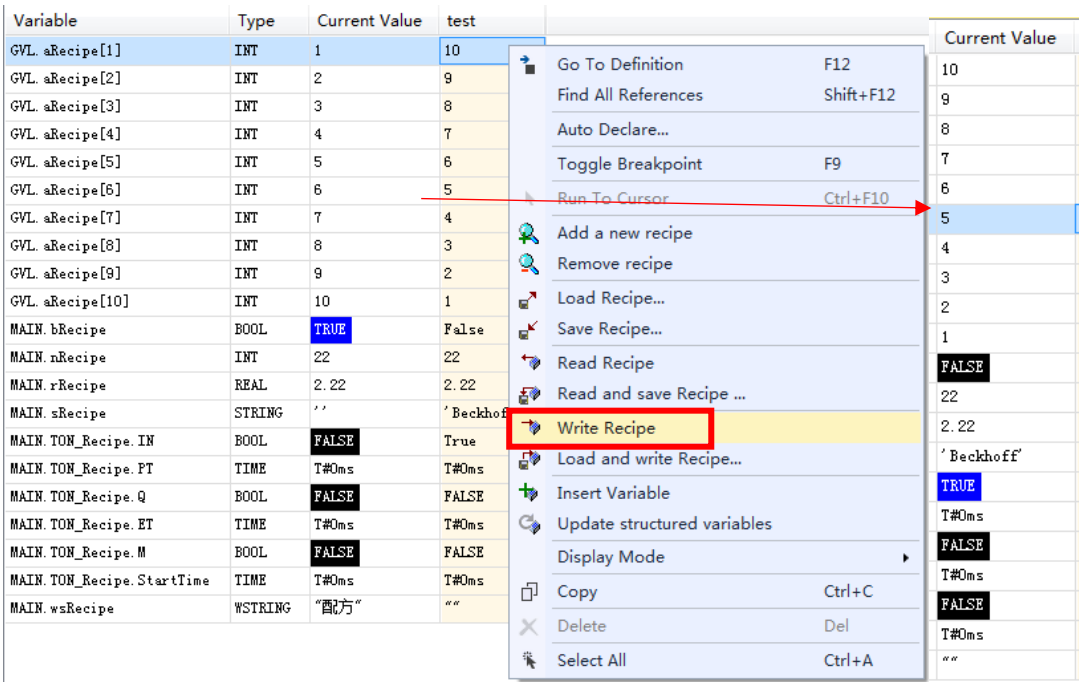


（8） 如果需要对配方进行更改有两种方式，通过修改保存的.txtrecipe 文件，或者直接在 Recipe Definition 中进行

首先第一种，通过记事本修改.txtrecipe 后缀的文件，点击 HMI 界面中的 Load 按钮上载配方，就可以在 Recipe Manager 下的 Recipes 中对当前配方进行查看，其中 Current Value 是在线值



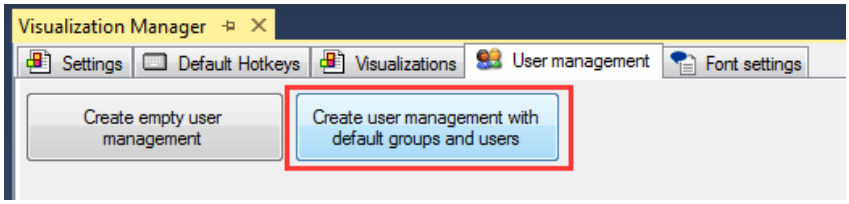
第二种方法就是直接在 RecipeManager 下的 Recipes 中进行，Recipe Definition 不仅可以监控当前配方的值，也可以在这个界面中直接对配方相关参数进行修改



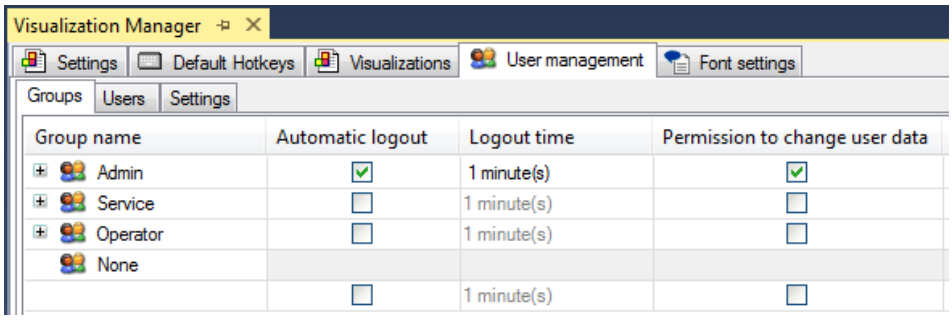
5 TwinCAT3 HMI 用户管理

用户管理方便用户对于界面进行自定义用户登录，不同用户所对应的控件操作权限可以进行自定义配置。

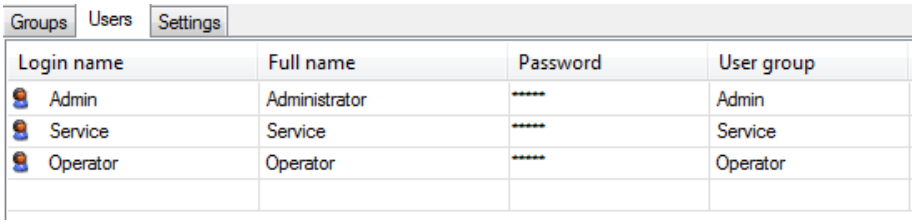
（1）打开前几个小结所编辑的界面，在 Visualization Manager 中创建一个默认的用户管理



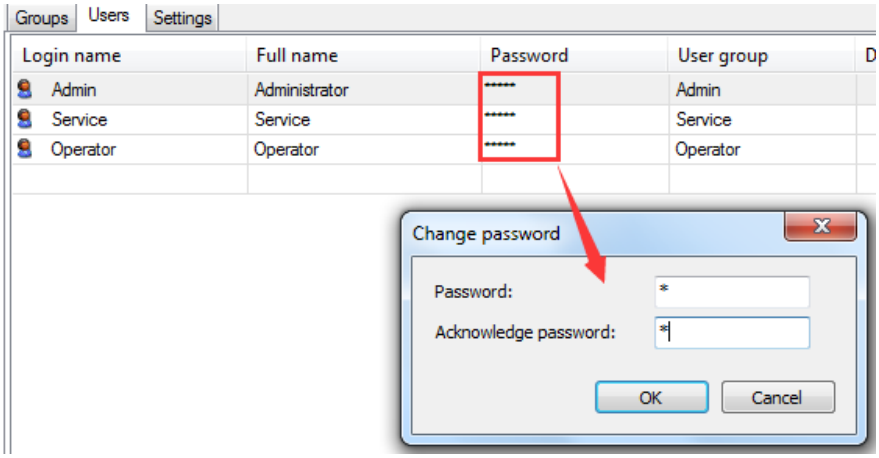
（2）创建好后，可以发现此默认的用户管理中已经创建了 3 个用户组



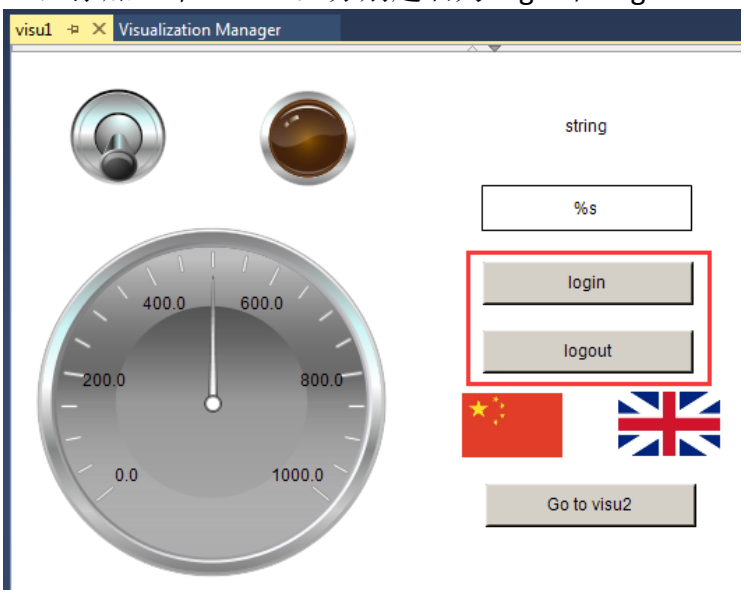
并且在 Users 中可以看到同时也已经创建了 3 个用户，并且对应到了不同组中



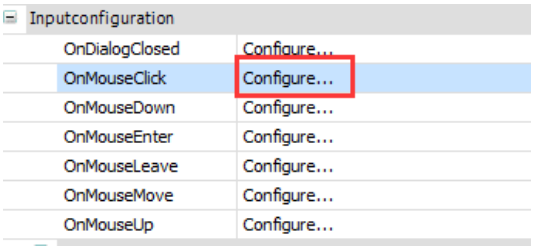
（3）在这里你可以新建用户组或者用户，并且修改登录名和密码，比如我把自带的 3 个用户密码都设置为 1，只需要点击 Password 中对应的一栏即可



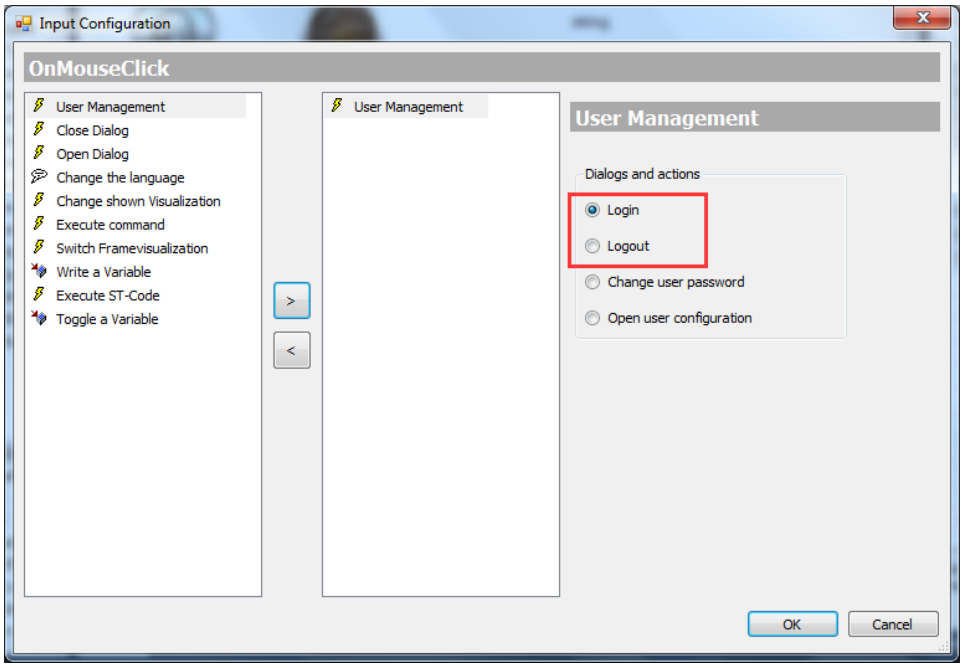
（4）回到 visu1，添加 2 个 Button，分别起名为 login 和 logout



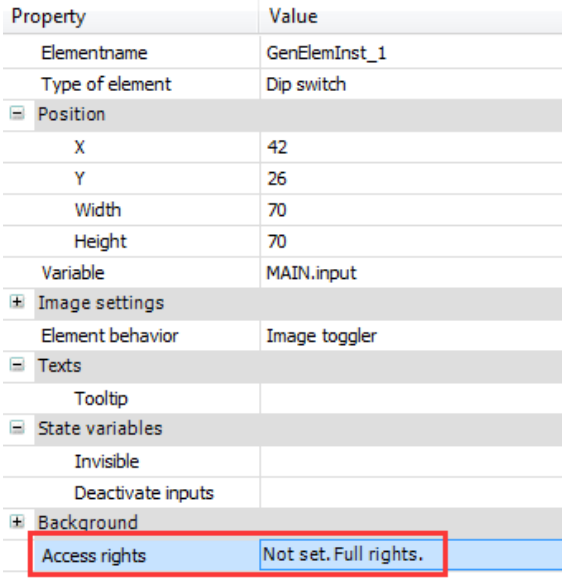
（5）分别配置这 2 个 Button 的 Inputconfiguration→OnMouseClick



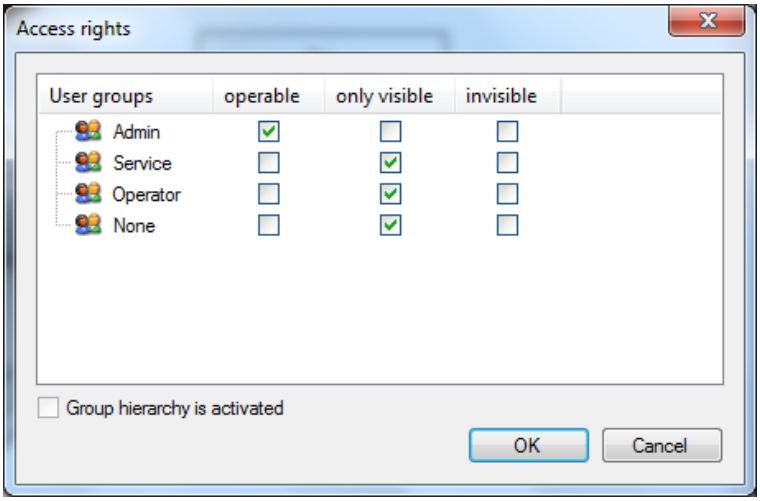
（7） 选择 User Managerment 后，分别把 2 个 Button 配置成 Login 和 Logout



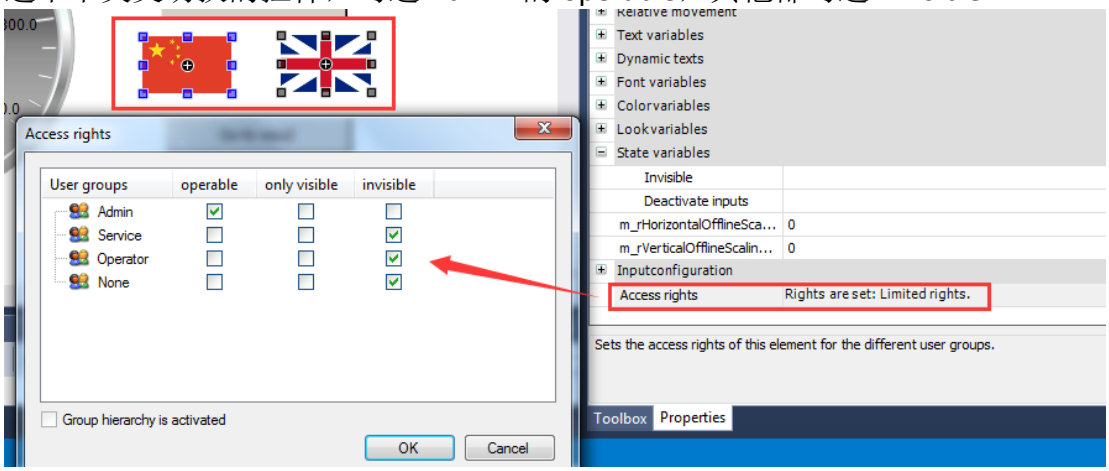
（7）接下来就可以针对不同控件配置不同的访问权限了，比如按钮控件，如果你希望此控件只有 Admin 用户可以操作，其他用户只是可见，那可以点击此控件，在属性栏中选择 Background→Access rights



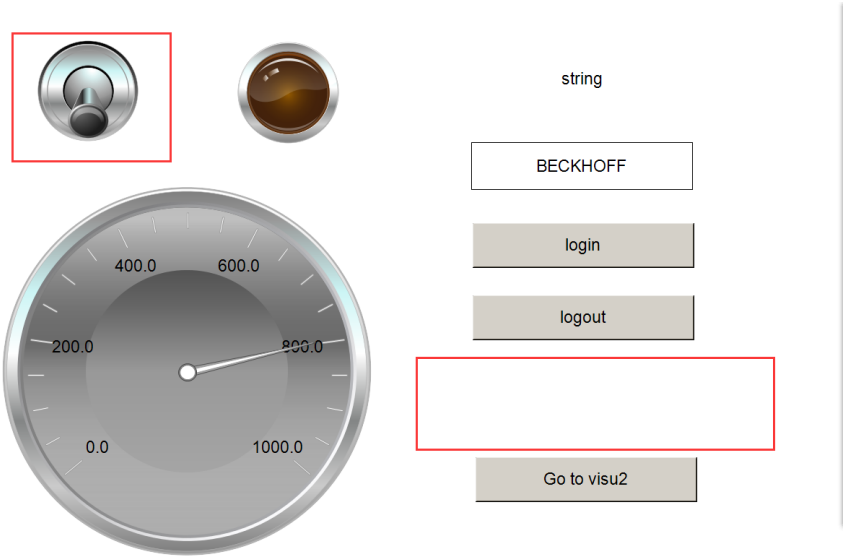
（8）弹出窗口，对应勾选 Admin 的 operable，其他都勾选 only visible



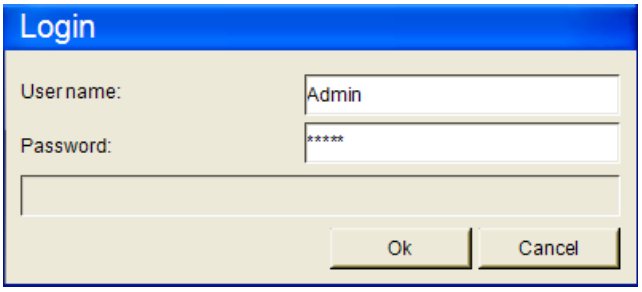
（9）同时如果希望中英文切换只有 Admin 可以操作，其他用户都不可见，可以选中中英文切换的控件，勾选 Admin 的 operable，其他都勾选 invisible



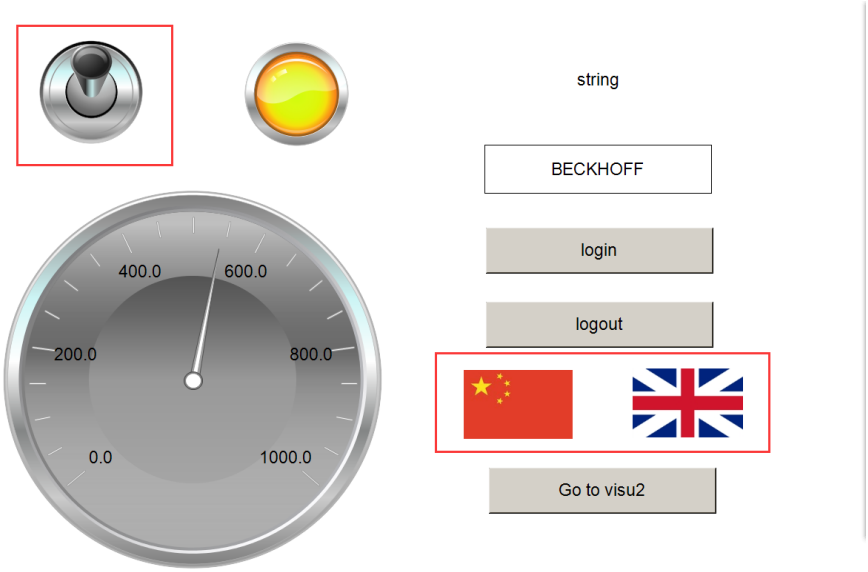
（10）等全部配置好后，重新激活配置并且全屏显示，可以看到如果没有进行任何用户登录，拨码开关可见但无法操作，中英文切换按钮不可见



（11）此时点击 login 后输入用户名和密码，比如 Admin 和 1



（12）随后就可以发现可以对拨码开关进行操作，并且中英文切换按钮可见并且可以操作



6 TwinCAT3 HMI-Web 使用方法

准备工作：在需运行样例的电脑和控制器中安装 TwinCAT3 软件，需安装 TC3 的 **v3.1.4018.13** 及以上版本才可以支持 HMI-Web 功能，以及安装 HMI Web 插件包：TF1810-PLC-HMI-Web

第一种方式：将电脑和控制器通过交换机相连，并且接入交换机的 LAN 口中。

（1）将控制器的网卡设置在 192.168 这个网段，本实例中控制器使用的 IP 地址为：192.168.1.100，如图 1 所示。

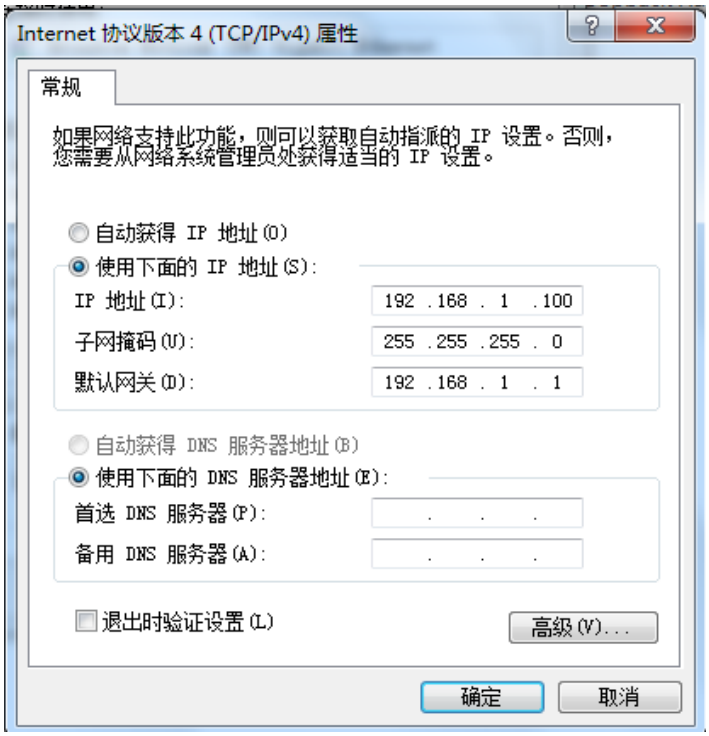


图 1 控制器的网络连接详细状态

（2）此时打开电脑，电脑的网段也设置成为 192.168 这个网段，如图 2 所示。进入下一步配置操作。

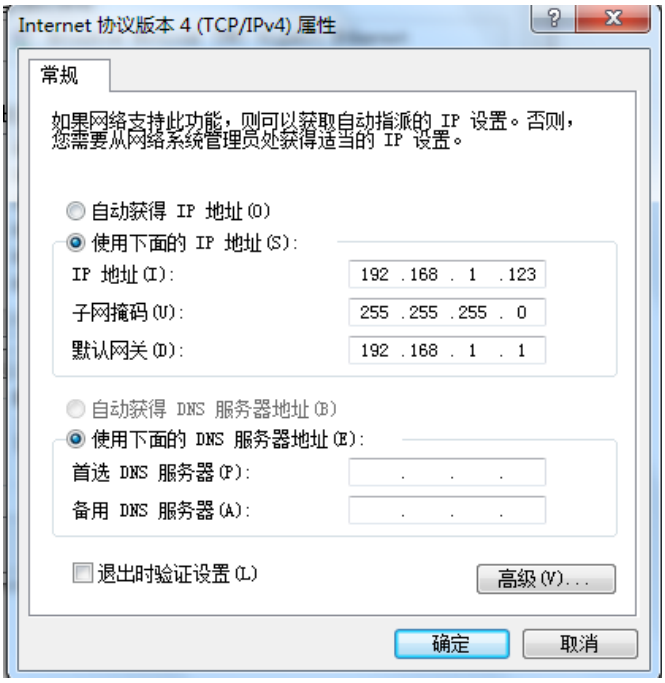


图 2 电脑的网络连接详细状态

（3）打开样例程序，按照正常操作模式，进行“Choose target”连接控制器，“Broadcast search”扫描硬件等操作。提供的样例程序是用一个按钮来控制 LED 的亮灭。

（4）如图 3 所示，在左侧对象管理器找到 Visualization Manager 右击，单击 Add，再单击 WebVisualization

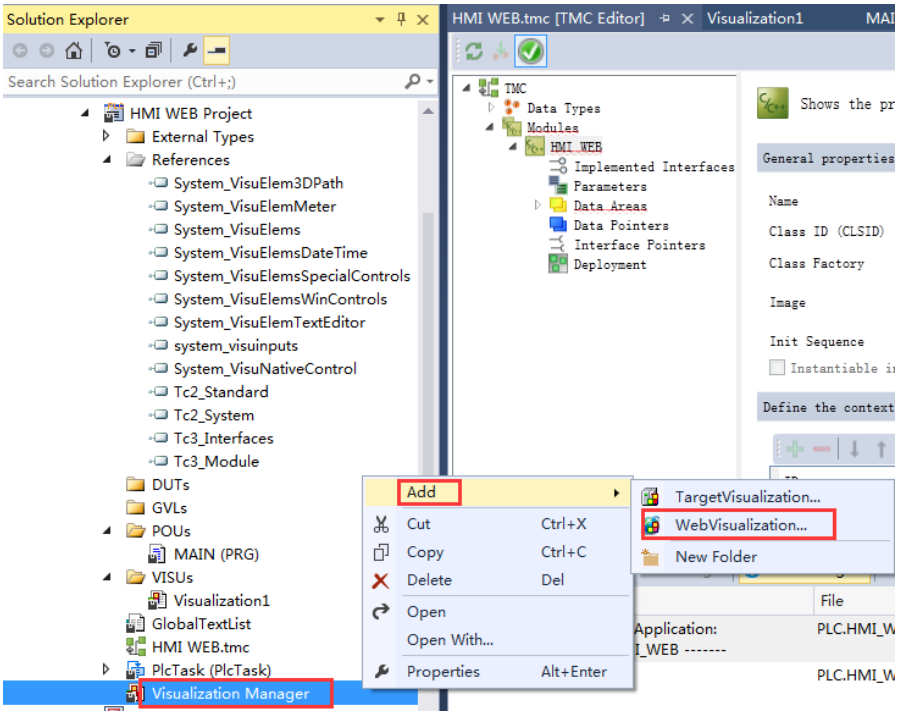


图 3 创建 WebVisualization 界面

（5）弹出界面当中，要求填写新建的 WEB 界面的名字，我们采用默认 WebVisualization，再次激活配置  ，切入运行模式。点击 Log in  和运行  。

（6）操作完成后，可关闭 TwinCAT3 软件。同时，打开电脑的浏览器，输入网址：http://(控制器的 IP 地址)/tc3plchmiweb/port_851/visu/webvisu.htm

本实例中,控制器的 IP 地址为：192.168.1.100，所以输入网址：http:// 192.168.1.100/tc3plchmiweb/port_851/visu/webvisu.htm

（7）本实例中，可以在与交换机相连的任意台电脑（该电脑可不安装 TC3 软件）的浏览器中用按钮控制 LED 的亮灭（如图 4，5 所示）

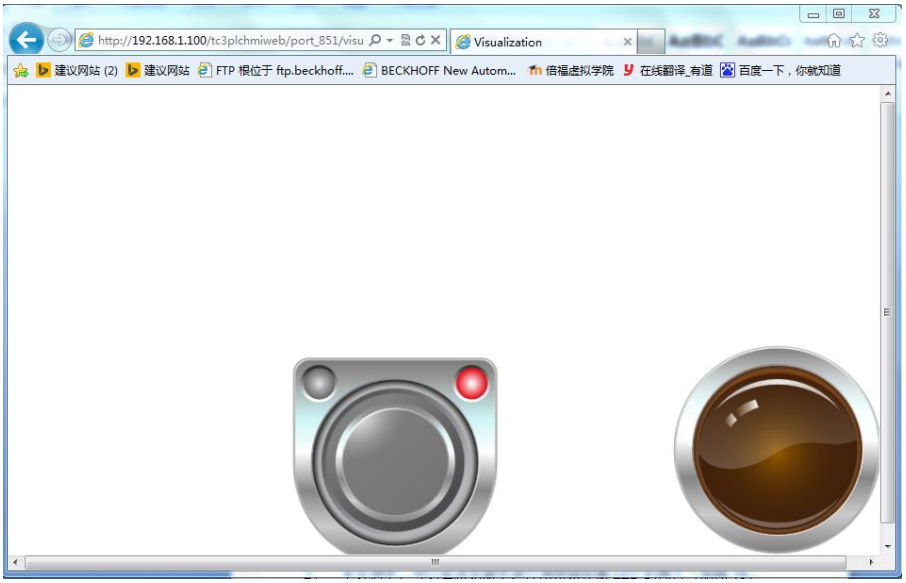


图 4 按钮弹起，LED 未亮

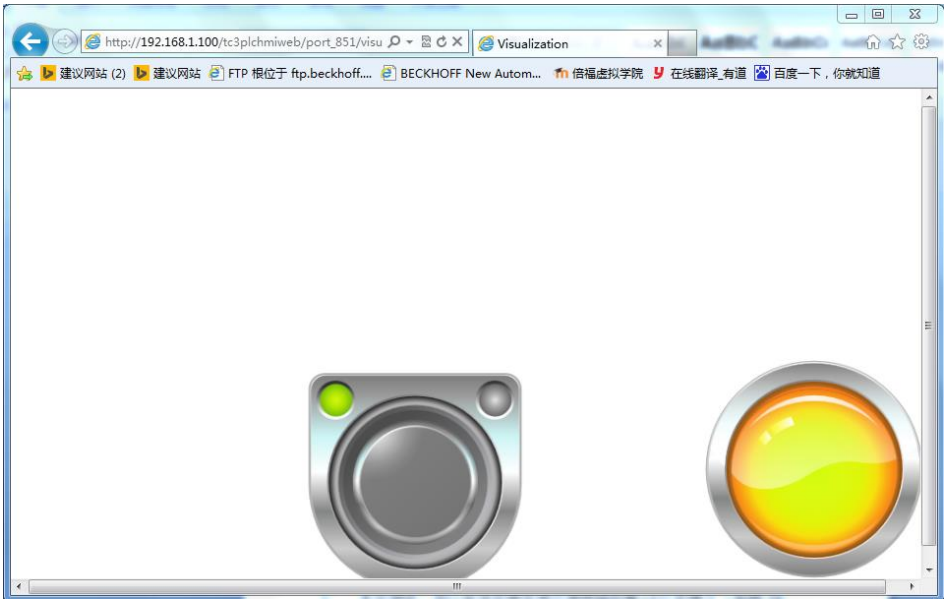


图 5 按钮按下，LED 亮起

第二种方式：将控制器网口与无线路由的 LAN 口通过网线相连，从而实现设备通过无线网络对控制器的 HMI Web 界面进行操作。

（1）将控制器的网卡设置在 192.168 这个网段，本实例中控制器使用的 IP 地址为：192.168.1.100，如图 6 所示。

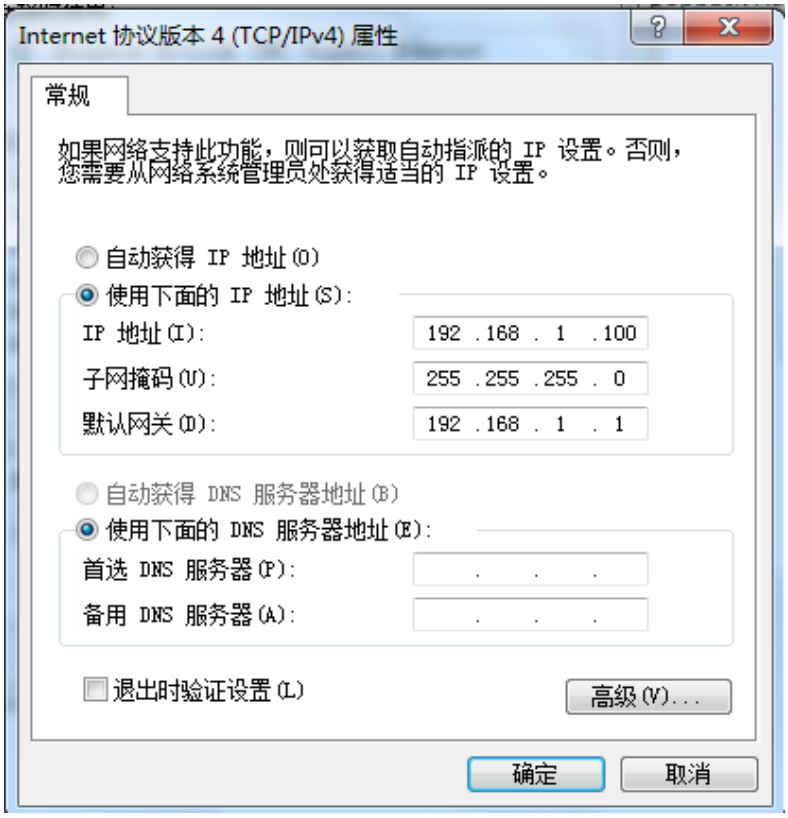


图 6 控制器的网络连接详细状态

（2） 接下来操作与第一种方式中的操作步骤 3-5 相同。

（3）电脑打开无线网，搜索无线信号，本实例中无线路由的名称为：TP-LINK，密码为：12345。连接成功。（其他电脑，手机等设备可用同样的方式，连接上无线路由）

（4）操作完成后，可关闭 TwinCAT3 软件。同时，打开电脑的浏览器输入网址：

http://(控制器的 IP 地址)/tc3plchmiweb/port_851/visu/webvisu.htm

本实例中,控制器的 IP 地址为：192.168.1.100，所以输入网址：

http:// 192.168.1.100/tc3plchmiweb/port_851/visu/webvisu.htm

（5）本实例中，在执行样例程序的电脑，或者其他任意电脑的浏览器，输入网址进入网页，在该网页中可用按钮控制 LED 的亮灭（如图 7，8 所示）。

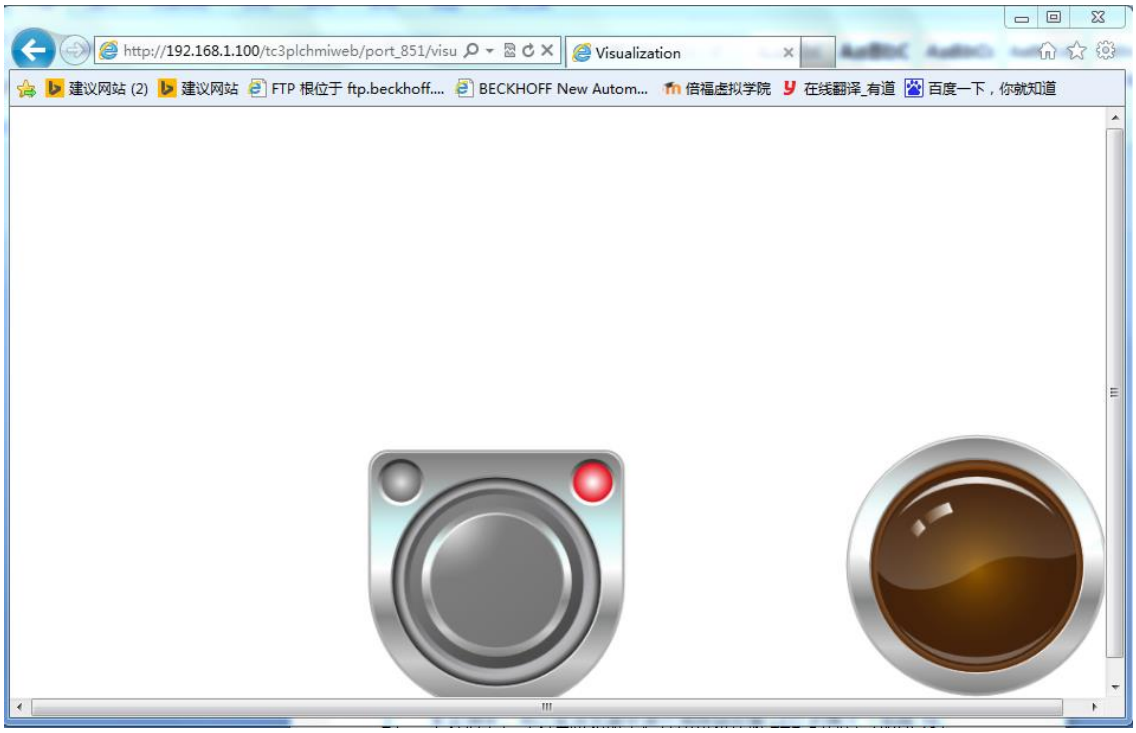


图 7 按钮弹起，LED 未亮

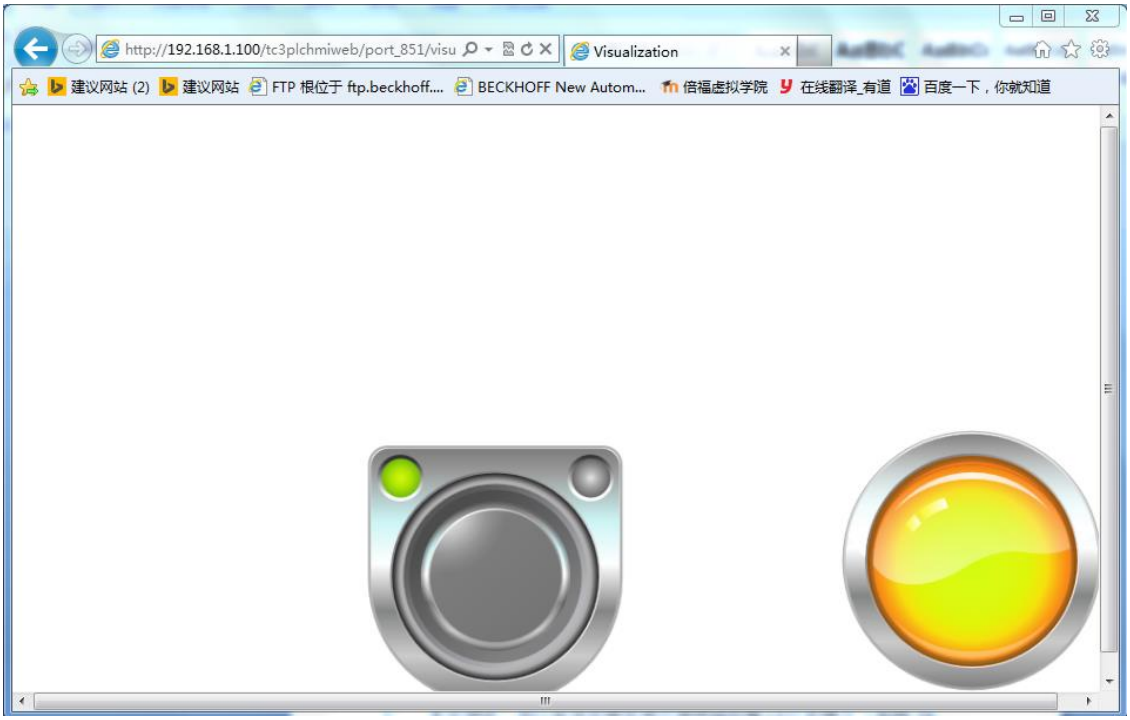


图 8 按钮按下，LED 亮起

（6）也可以用已连接上无线路由 TP-LINK 的手机，打开浏览器，输入网址：**http://192.168.1.100/tc3plchmiweb/port_851/visu/webvisu.htm**，同样可以显示出 HMI Web 界面，并对之进行操作（如下图 9，10 所示）

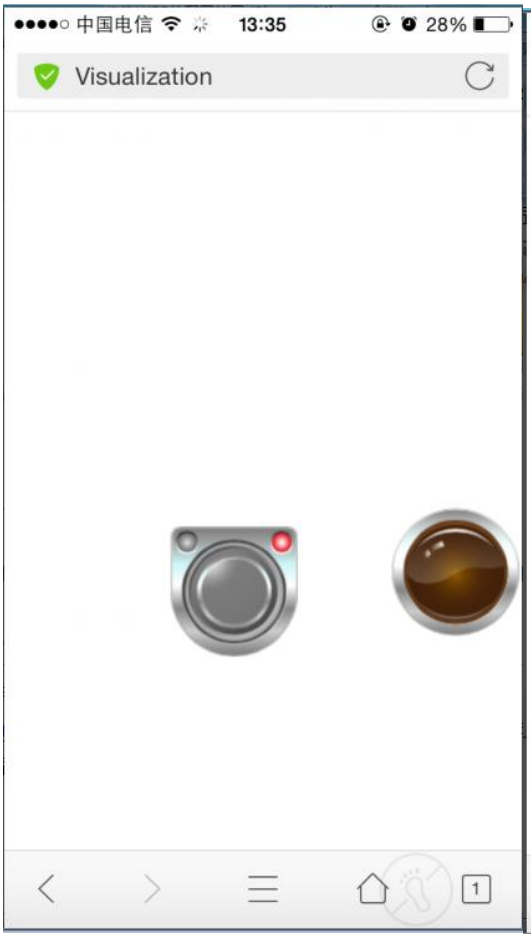


图 9 按钮弹起，LED 未亮

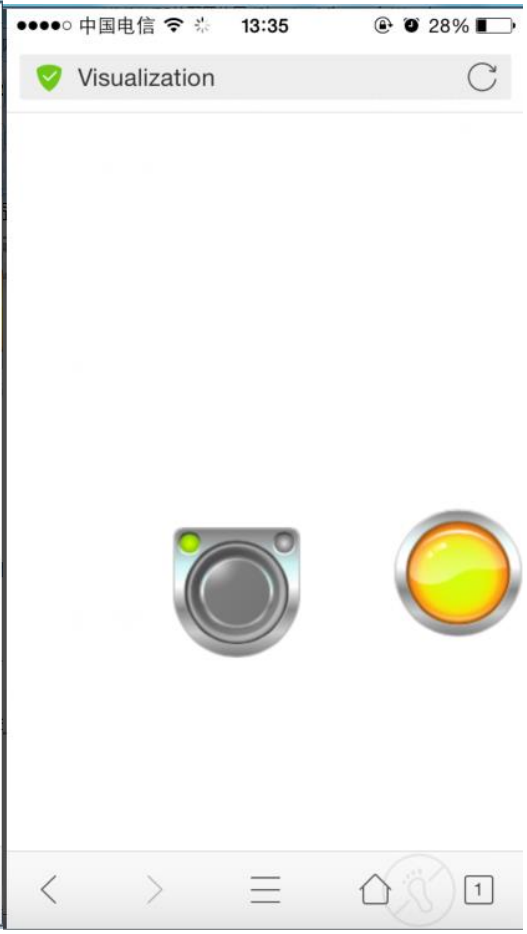


图 10 按钮按下，LED 亮起

七、 TwinCAT-3 Scope View 的使用

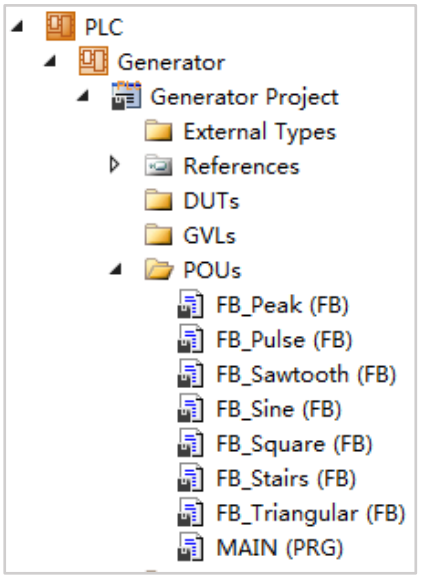
TwinCAT3_Scope view 和早期版本的产品一样是一种图形化输出的分析工具，可以从 TwinCat3 程序中获取变量以记录并显示图形，通过它可以对本地及远程的设备进行系统和变量的评估，具有 cursor 游标和 trigger 触发器等功能。

1. TwinCAT 3-scope view 的加载和简单使用

1.1 ScopeYT Project 使用（**TC3.1.4022 版本**）

本小节需要用到的程序如下，PLC 工程的新建和程序的编写此处不多做赘述，程序编写如下，本例程已包含在样例程序中：

功能块创建：



功能块具体代码：

FB_Triangular

```
FB_Triangular*
FUNCTION_BLOCK FB_Triangular
VAR_OUTPUT
    fTriangular:REAL;
END_VAR
VAR
    fSum:REAL;
    fDelta:REAL;
    n: INT;
    fIncrement: LREAL;
    fPhase: INT:=180;
END_VAR
```

FB_Triangular*

```
FOR n := 1 TO 10 DO
    IF (fIncrement >= 360) THEN
        fIncrement := fIncrement - (fIncrement/360)*360;
    END_IF
    IF fIncrement < fPhase/2 THEN
        fDelta := 0.001;
    ELSIF fIncrement >= fPhase/2 AND fIncrement < 3*fPhase/2 THEN
        fDelta := -0.001;
    ELSIF fIncrement >= 3*fPhase/2 THEN
        fDelta := 0.001;
    END_IF
    fIncrement := fIncrement + 0.018;
    fSum := fSum + fDelta;

        IF fSum > 5 THEN
        fSum := 5;
    ELSIF fSum < -5 THEN
        fSum := -5;
    END_IF
    fTriangular := fSum;
END_FOR
```

FB_Sine

FB_Sine*

```
FUNCTION_BLOCK FB_Sine
VAR_OUTPUT
    fSine: LREAL;
END_VAR
VAR
    fProduct: LREAL;
    n: INT;
    fIncrementSample:REAL:=0.0036;
    fIncrement: REAL;
    fIncrementCycle: REAL:=0.036;
END_VAR
```

FB_Sine*

```
FOR n:= 1 TO 10 DO
    fProduct := 5*SIN(2*PI*((fIncrement+(n-1)*fIncrementSample)/360));
END_FOR

fIncrement := fIncrement + fIncrementCycle;

fSine:=fProduct;
```

FB_Peak

FB_Peak

```
FUNCTION_BLOCK FB_Peak
VAR_OUTPUT
    fPeak: REAL;
END_VAR
VAR
    fDetal: REAL;
    fExponential: REAL;
    n: INT:=1;
END_VAR
```

FB_Peak

```
FOR n := 1 TO 10 DO
    IF fDetal < 1000 THEN
        fExponential:=EXP((0.0023 * fDetal));
        fDetal := fDetal + 1;
    ELSE
        fDetal := -9000;
    END_IF
END_FOR
fPeak:=fExponential-5;
```

FB_Stairs

FB_Stairs

```
FUNCTION_BLOCK FB_Stairs
VAR_OUTPUT
    fStairs: INT;
END_VAR
VAR
    n: INT;
    fDetal: INT;
    nActPosCycle: INT;
END_VAR
```

FB_Stairs*

```
FOR n := 1 TO 10 DO
    nActPosCycle := nActPosCycle + 1;
    IF nActPosCycle > 16667 THEN
        IF fDetal < 5 THEN
            fDetal := fDetal + 1;
        ELSE
            fDetal := 0;
        END_IF
        nActPosCycle := 0;
    END_IF
    fStairs := fDetal;
END_FOR
```

FB_Square

FB_Square

```
FUNCTION_BLOCK FB_Square
VAR_OUTPUT
    fSquare: INT;
END_VAR
VAR
    n: INT;
    nDetal: INT;
    fAngularIncrement: LREAL;
END_VAR
```

FB_Square*

```
FOR n := 1 TO 10 DO
    IF (fAngularIncrement >= 360) THEN
        fAngularIncrement :=
            fAngularIncrement - DINT_TO_LREAL(LREAL_TO_DINT(fAngularIncrement/360))*360;
    END_IF
    IF (fAngularIncrement < 180) THEN
        nDetal := 1;
    ELSIF (fAngularIncrement >= 180) THEN
        nDetal := -1;
    END_IF
    fAngularIncrement := fAngularIncrement + 0.018;
    fSquare := nDetal*5;
END_FOR
```

FB_Sawtooth

FB_Sawtooth

```
FUNCTION_BLOCK FB_Sawtooth
VAR_OUTPUT
    fSawtooth: LREAL;
END_VAR
VAR
    n: INT;
    fLastSignalValue: LREAL;
    fAngularIncrement: LREAL;
END_VAR
```

FB_Sawtooth*

```
FOR n := 1 TO 10 DO
    IF (fAngularIncrement >= 360) THEN
        fAngularIncrement :=
            fAngularIncrement - DINT_TO_LREAL(LREAL_TO_DINT(fAngularIncrement/360))*360;
        fLastSignalValue := fLastSignalValue -5;
    END_IF
    fAngularIncrement := fAngularIncrement + 0.018;
    fSawtooth := fLastSignalValue + 0.00025;
    fLastSignalValue:=fSawtooth;
END_FOR
```

FB_Pulse

FB_Pulse

```
FUNCTION_BLOCK FB_Pulse
VAR_OUTPUT
    fPulse: INT;
END_VAR
VAR
    n: INT;
    fDetal: INT;
    fAngularIncrement: LREAL;
END_VAR
```

FB_Pulse*

```
FOR n := 1 TO 10 DO
    IF (fAngularIncrement >= 360) THEN
        fAngularIncrement :=
            fAngularIncrement - DINT_TO_LREAL(LREAL_TO_DINT(fAngularIncrement/360))*360;
    END_IF
    IF (fAngularIncrement < 180) THEN
        fDetal := 1;
    ELSIF (fAngularIncrement >= 180) THEN
        fDetal := 0;
    END_IF
    fAngularIncrement := fAngularIncrement + 0.0036;
    fPulse := fDetal*5;
END_FOR
```

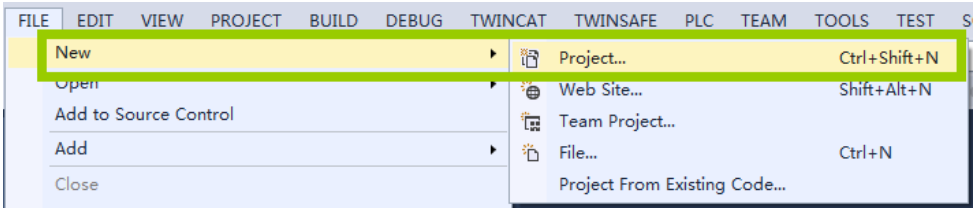
功能块实例化和变量声明：

```
PROGRAM MAIN
VAR
    FB_Triangular:FB_Triangular;
    FB_Sine:FB_Sine;
    FB_Peak:FB_Peak;
    FB_Stairs:FB_Stairs;
    FB_Square:FB_Square;
    FB_Sawtooth:FB_Sawtooth;
    FB_Pulse:FB_Pulse;
    fTriangular: REAL;
    fSine: LREAL;
    fPeak: REAL;
    fStairs: INT;
    fSquare: INT;
    fSawtooth: LREAL;
    fPulse: INT;
END_VAR
```

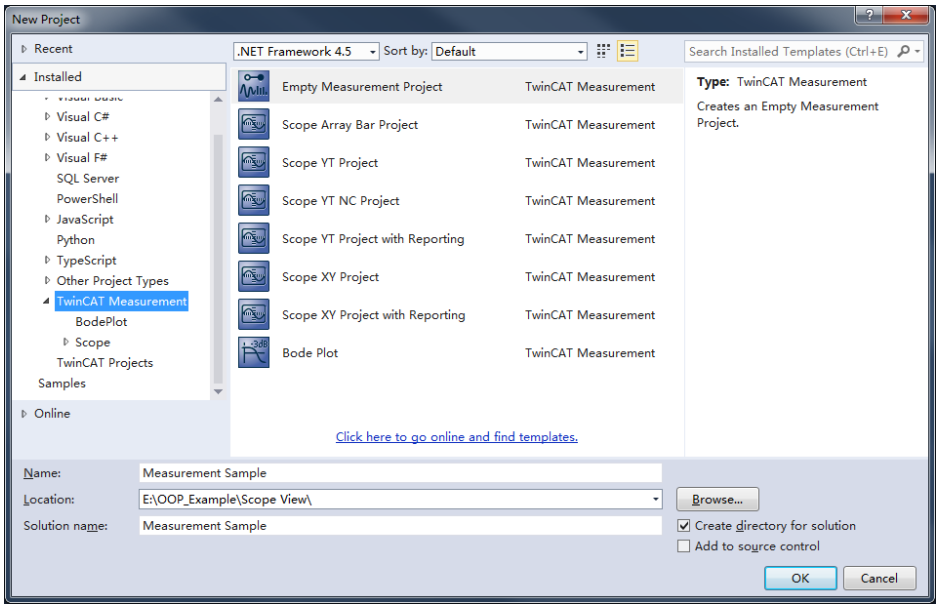
主程序调用：

```
FB_Triangular(fTriangular=> fTriangular);
FB_Sine(fSine=> fSine);
FB_Peak(fPeak=> fPeak);
FB_Stairs(fStairs=> fStairs);
FB_Square(fSquare=> fSquare);
FB_Sawtooth(fSawtooth=> fSawtooth);
FB_Pulse(fPulse=> fPulse);
```

（1）首先新建一个 TwinCAT3 项目。菜单栏选择 File-> New-> Project



弹出窗口选择 TwinCAT Measurement，有很多工程模板可供选择。

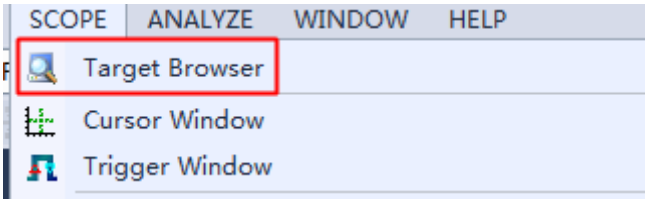


Empty Measurement Project：空项目
Scope Array Bar Project：用柱状图监控数组变量
Scope YT project：单个变量随着时间变化而变化进行监控。
Scope YT NC project：创建后自动添加 NC 轴中主要的一些变量，只需要指定相对于的 NC 轴的编号就可以直接进行监控。
Scope YT project with Reporting：带 reporting 功能的 YT 曲线图。
Scope XY project：横纵坐标都可以制定为变量，观察 2 个变量的曲线图。
Scope XY project with Reporting：带 reporting 功能的 XY 曲线图。
Bode Plot：波特图可以观察系统的频率响应，以及不同频率下，系统增益的大小及相位，也可以看出大小及相位随频率变化的趋势等。

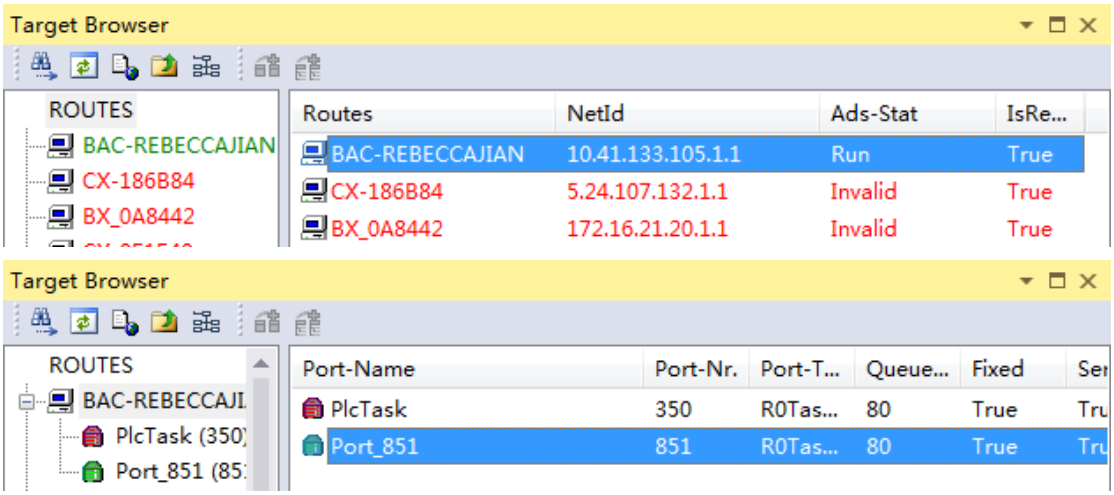
此处以 Scope YT project 为例，在模板中选择 YT project 并且点击 OK 进行创建。

在开始添加变量观察变量之前，我们需要将源程序激活并进行下载运行。这么做的目的是为了创建 851 端口，希望 Scope 能通过 ADS 协议获取到变量。编译无报错后，点击 Activate Configuration 并且切换到 Run Mode。点击 Login 和 Start，下载并运行程序。

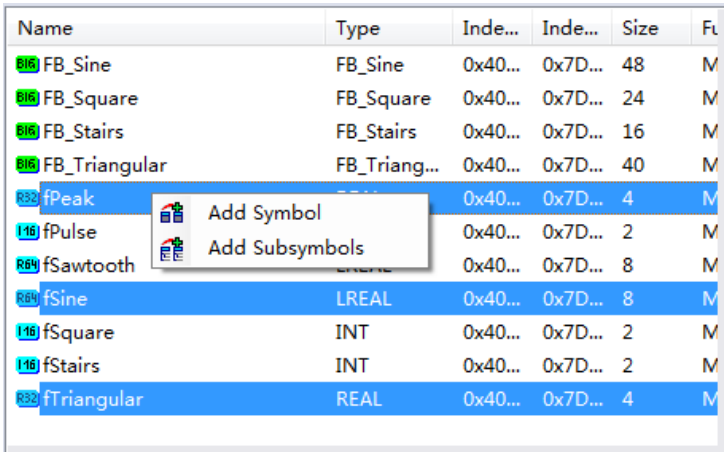
（1）在菜单栏中点击 SCOPE 选择 Target Browser。



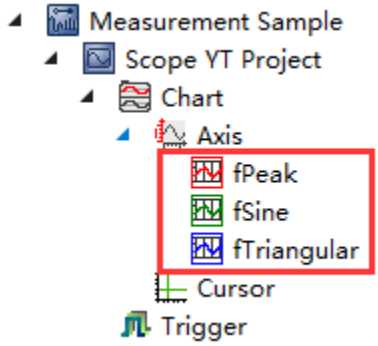
（2） 选择对应的目标控制器，选择 PLC 端口号 851（在 TC3 中 PLC 端口号从 851-899，起始第一个是 851），找到 MAIN 程序中对应的三个变量：fTriangular、fSine、fPeak。



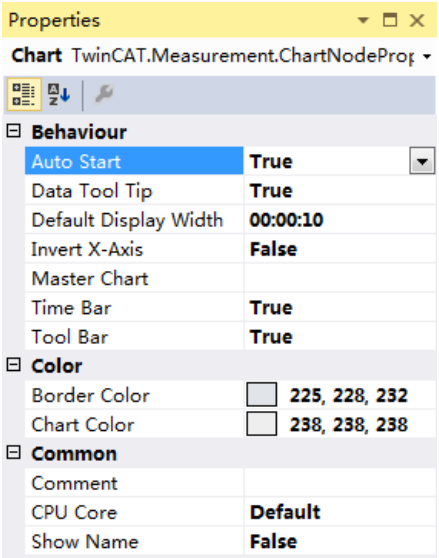
（3） 右键需要添加的变量 Add Symbol 或者双击。

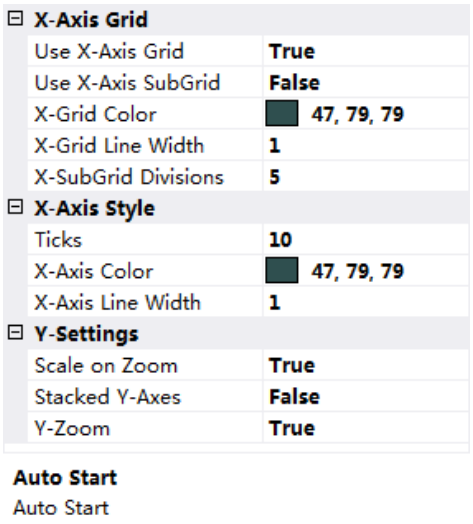


三个变量就被添加到了 Axis 中。



（4） 在 chart 的属性里，可对 chart 窗口的一些属性进行更改：





Behavior 性能

Auto Start（自动开始）：决定当这个图表有新记录时是否启用即时模式。

Default Display Time（默认显示时间）：这个时间是当记录开始时或重新缩放时，整张图表显示的时间跨度。

Invert X-Axis（X 轴反转）：决定 X 轴是由小到大还是由大到小。

Master Chart（主图标）：可绑定附属图表，达到观测同步。

Time Bar（时间条）：在图表中显示或隐藏时间条。

Tool Bar（工具条）：在图表中显示或隐藏工具条。

Color 颜色

Border Color（边框颜色）：图表框架颜色。

Chart Color（图表颜色）：图表的画布颜色。

Common 普通

Comment（注释）：在此输入注释。

CPU Core（CPU 核）：将对应 chart 的记录工作分配到不同核里去。默认则由 windows 系统分配。

Show Name（显示名称）：在绘图区显示或隐藏图标名称。

X-Axis Grid X 轴网格

Use X-Axis Grid（网格）：开启或关闭 X 轴网格。

Use X-Axis SubGrid（子网格）：开启或关闭 X 轴子网格。

X-Axis Color（颜色）：网格颜色。

X-Axis Line Width（线宽）：网格的像素宽度。

X-SubGrid Divisions（子网格分隔数）：只在开启 X 轴子网格时有效，具体将大网格细分的格数。

X-Axis Style X 轴类型

Ticks（格数）：X 轴分割的最大格数。

X-Axis Color（颜色）： 轴的颜色。

X-Axis Line Width（线宽）：轴的像素宽度。

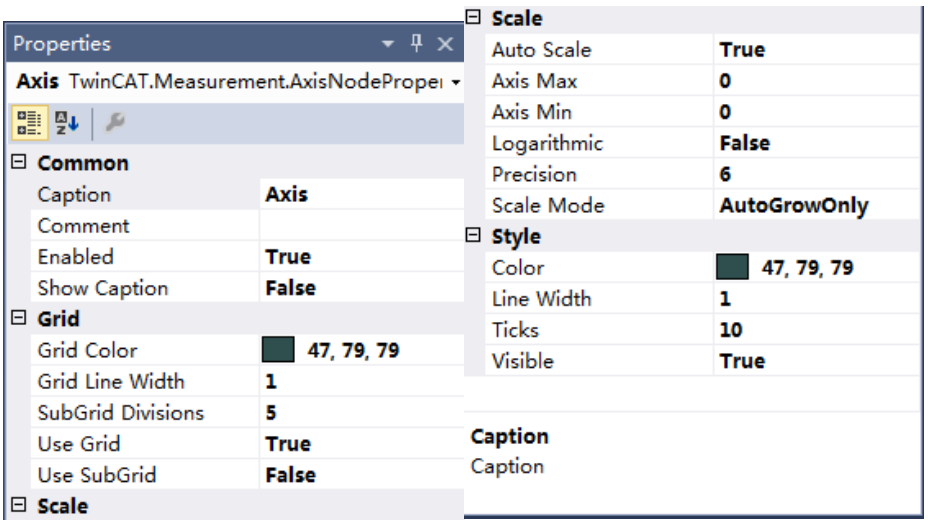
Y-Settings Y 轴设定

Scale on Zoom（缩放比例）：如果选 true 则图表会在缩放后引导所有轴遵循自动比例。

Stacked Y-Axes（Y 轴堆叠）：如果图表中添加了多于 1 条 Y 轴，应使用 "Stacked Y-Axes"来改变关联显示区域的布置。如果未选择堆叠选项，轴将会使用相同区域来显示通道的值。否则这些区域则是上下叠放的。

Y-Zoom（缩放）：决定缩放时是否只在 X 方向生效或 Y 方向同样生效

（5） 在 Axis 的属性选项卡里，可对 Axis 的一些属性进行更改：



Common 普通

Caption（轴名称）：Y 轴的名称。

Comment（注释）：在此输入注释。

Enabled（使能）：开启或关闭轴的记录。

Show Caption（显示名称）：在绘图区显示或隐藏轴名称。

Grid 网格

Grid Color（颜色）：网格颜色。

Grid Line Width（线宽）：网格线的像素宽度。

SubGrid Divisions（子网格分隔数）：只在子网格开启时有效，具体将 Y 轴大网格细分的格数。

Use Grid（网格）：开启或关闭 Y 轴网格。

Use SubGrid（子网格）：开启或关闭 Y 轴子网格。

Scale 刻度

Auto Scale（自动）：开启或关闭自动调整。 使用自动调整模式，轴的最小值（和最大值） 将会自动随关联通道的最小值或最大值调整。

Min / Max（最大/最小值）:如果不选择自动刻度模式则轴的最大和最小值按照文本框中输入的值来设定。

Logarithmic（轴对数）：对轴取对数显示。

Precision（精确度）：调整精确度。

Scale Mode（刻度模式）：两种模式切换。AutoGrowOnly 是 X 轴随 X 变量变化，但是最终显示 X 所达到过的历史最大最小值；AutoGrowNShrink 则是 X 轴的实时显示，就是当前 X 达到的最大最小值，实际上就是对应到显示时间了。

Axis Style 轴类型

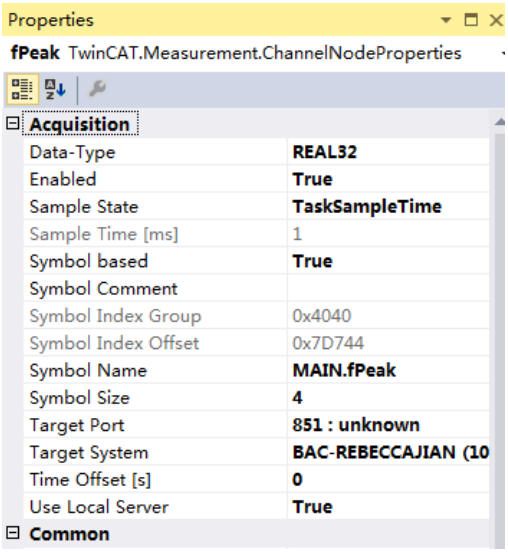
Color（颜色）：轴的颜色。

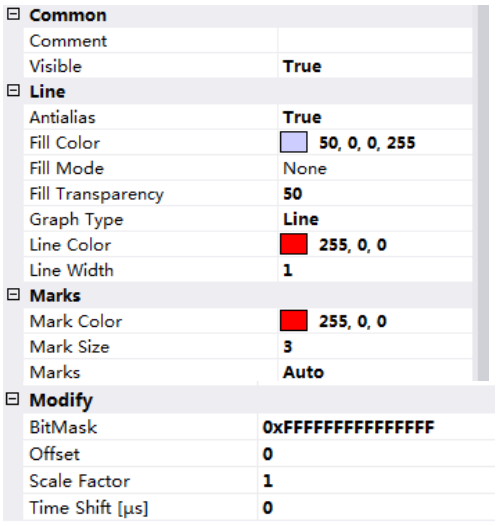
LineWidth（线宽）：轴的像素宽度。

Ticks（格数）：轴分割的最大格数。

Visible（可见）：显示/隐藏图表中的轴。

（6） 在变量的属性里，可对变量的一些属性进行更改：





Acquisition

Data-Type（数据类型）：所观察变量的数据类型，只读项不可更改。

Enable（变量记录使能）：所观察变量是否记录，只能在 Stop Record 之后才可以更改。

Sample State（采样周期状态）：默认和 PLC 周期的一致，另一种模式是 Free Sample，可自由定义大于 PLC 的采样周期。

Sample Time [ms]（采样周期）：采样周期，只有在 Sample State 为 Free Sample 的情况下才能改动。

Symbol based（信号识别模式）：通过地址组别或通过变量名。常规不更改。

Symbol Comment（注释）：在此输入注释。

Symbol Index Group（信号组别）：只在 Symbol Based 为 False 时可改，变量在系统中所存地址组别地址。常规不更改。

Symbol Index Offset（信号组别偏移）：只在 Symbol Based 为 False 时可改，变量在系统中所存地址组别偏移地址。常规不更改。

Symbol Name（信号变量名）：在此可通过修改对应变量名更换所需观察变量

Symbol Size：这个值对应数据类型的存储大小，只读项，修改可能造成无法记录。

Target Port（端口号）：这是 scope 获取到的变量所处端口号。

Target System（目标系统）：目标控制器的名称。

Time Offset[s]（时间偏移）：时间偏移。

Use Local Server（服务器选择）：可选是否使用本地 Server 进行采集，多用于远程较多时分配。

Common 普通

Comment（注释）：在此输入注释。已连接的 ADS 标志中的注释会自动插入。

Visible（可见）：显示或隐藏曲线。

Line 线条

Antialias（平滑）：平滑属性决定了曲线在绘制过程中是以美观（平滑度）或快速为主。

Fill Color（颜色）：在画布上可以选择区域进行色彩填充以示区分。此处可选填充色。

Fill Mode（填色模式）：四种模式，None 无填充色；Horizontal Zero 以零做分割线，包含在曲线和零线之间的部分均会被填充；Bottom 所有在曲线以下的部分均会被填充；Top 所有在曲线以上的部分均会被填充。

Fill Transparency（透明度）：填色透明度

Line Color（线色）：曲线的颜色。

Line Width（线宽度）：曲线的像素线宽。

Marks 标记

On：开启数据点高亮。

Auto（自动）：使数据点高亮与否取决于当前缩放。

Off：关闭数据点高亮。

Size（尺寸）：标记的尺寸。

Color（颜色）：标记的颜色。

Modify 调整

Bit Mask（位掩码）：位掩码用于从通道值中剪裁单个比特位。显示的值是通道值与位掩码相与后的结果。浮点型的值不能被任何掩码改变。这个选项在观察

状态字的单独位时十分有用。

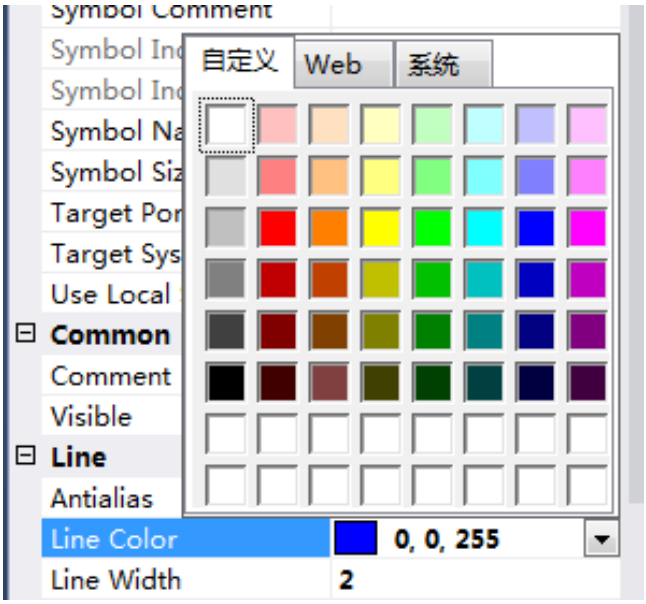
Offset（偏移）：将给通道添加一个偏移量使其在图表中安排得更合理。

Scale（刻度）：通过将刻度系数设为 1 可以给显示的通道值标上刻度。这个功能对于像以可读性更好的角度制来显示弧度制的角度值很有帮助。 如 Scale = 360/(2*Pi) = 57.296。

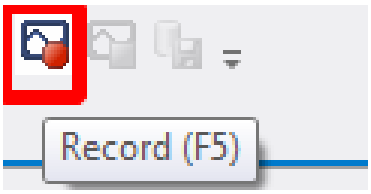
Time Shift（时移）：时移是时间线上的偏移量。它有助于对已知停止时间的通道进行比较。

比较常用到的比如 symbol name、，可以直接更改变量，可以改为 main.b，那么监测的就是 main 函数里 b 变量的值。

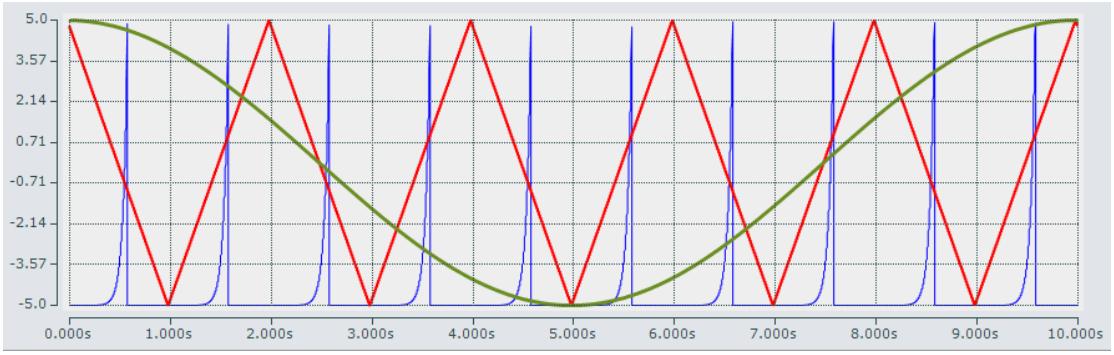
或者是 line 的属性，颜色和线宽，TwinCat3 会自动给不同变量不同的颜色以示区分，我们也可以在这里手动更改，如果觉得线宽默认 1 太细，也可以往上加。



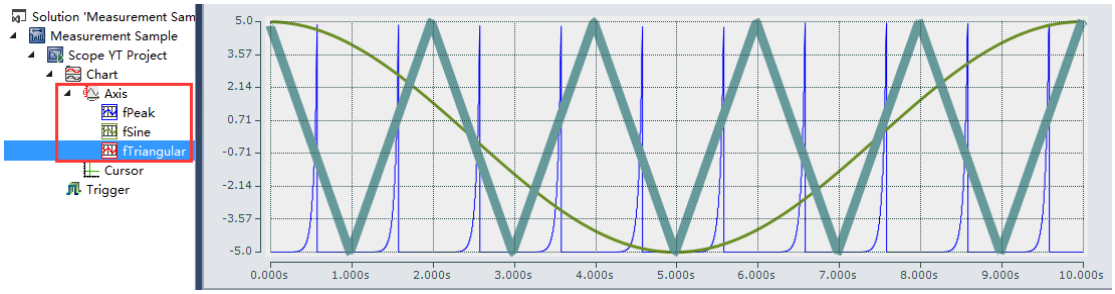
（8）点击工具栏中的 record。此时 PLC 程序要在运行当中才能采集到实际值。



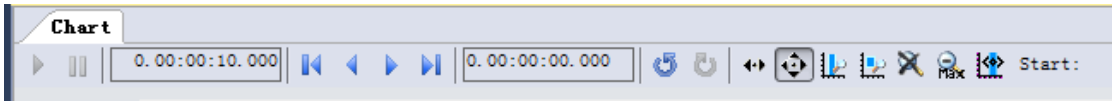
（9）可以开始观察变量曲线。



（10）双击 axis 或者 axis 下的变量，TwinCat scope 就会高亮显示变量曲线，方便在多变量时候区别出来。



（11）Chart 工具栏包含了操纵连接数据的主要命令。以下是按照如下所示图片从左到右的分类解释：



Play（开始）：当 Scope 在记录模式时，开始即时显示模式。当前的帧是标绘在图表上的。

Pause（暂停）：停止计时绘图，对数据进行操作。

Display-Width（显示宽度）：以"hh:mm:ss.fff"的格式改变显示宽度（X 轴）。您也可以用鼠标来改变显示宽度：选中一张图表，按下 <Ctrl>转动滚轮。当你放开 <Ctrl>时画面将会刷新显示宽度。

Scroll Buttons（滚动按钮）：两边的滚动按钮按整个画面的宽度移动。中间的两个的按钮以较小幅度滚动。

Position（位置）：以 "hh:mm:ss.fff"的格式在此设定图表的当前位置。

Undo/ Redo Time/Position（撤销/恢复 时间/位置）：按发生的次序撤销位置或显示宽度的改变。也可使用右键。当一个动作被撤销，可以通过恢复选项将它复原。

Zoom-Mode（缩放模式）：当缩放模式选中时，您可以用鼠标缩放选中区域。在 Chart 窗口中选择 Y-Zoom” 选项可以激活 Y 方向缩放。

Panning Horizontal（水平平移）：在图表上单击并拖动鼠标以在 X 方向上移动当前范围。

Panning Free（自由平移）：在图表上单击并拖动鼠标以在 X 或 Y 方向上移动当前范围。

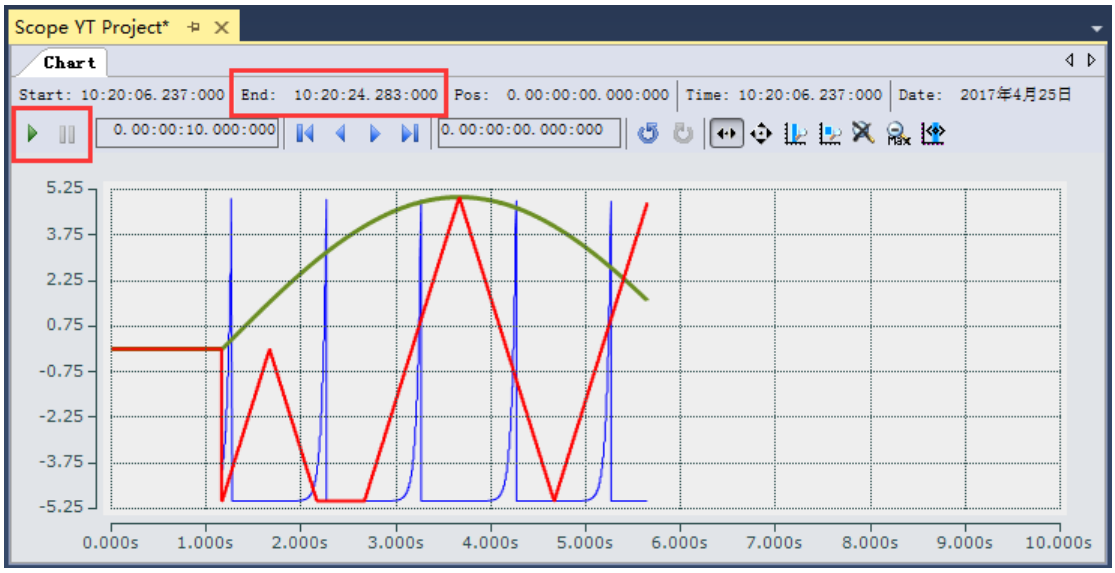
Zoom Horizontal（水平缩放）：使用鼠标在 x 轴上跨越一定范围以显示新的时间夸度。

Zoom Free（自由缩放）：在图表上使用鼠标跨越一定范围缩放至当前画面。

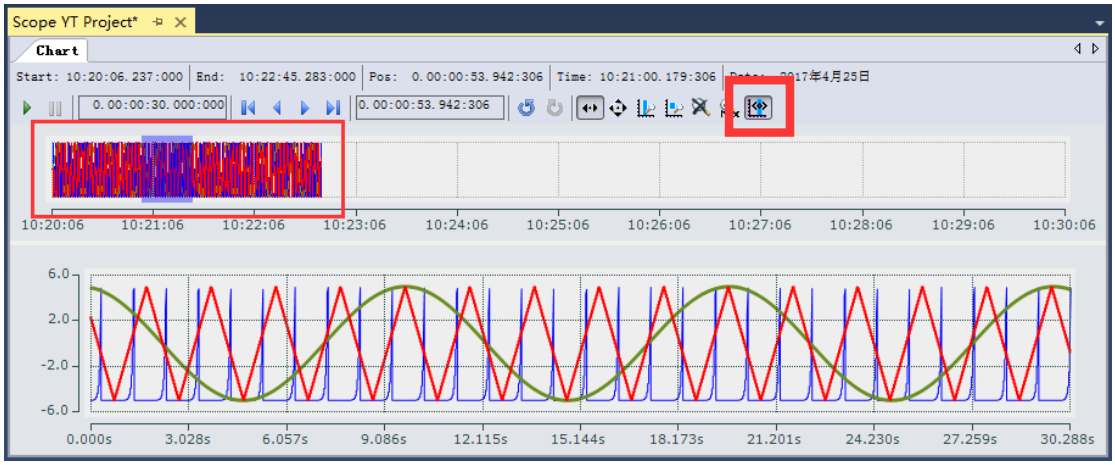
Rescale All（重缩放）： 调整所有 Y 轴重新缩放。X 轴设定成默认显示宽度。

Zoom Out Max（全部缩放）：将 X 轴范围设置成一个较大的时间跨度或等于当前记录长度。

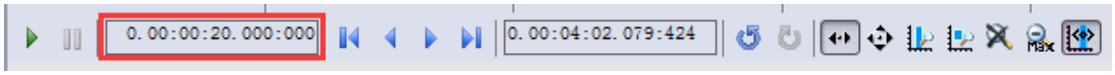
比如在下图中，点击 toolbar 最左侧是 Start/Stop Display 按钮，位于下方的图表停止了，而记录仍在后台运行，end 值一直在增加。之后可以对画面做一些操作，例如可以设置游标或对曲线进行缩放。



toolbar 右侧有个 Overview 功能，可以多出一个轴查看整个监控过程，从中可以选择一部分进行查看。



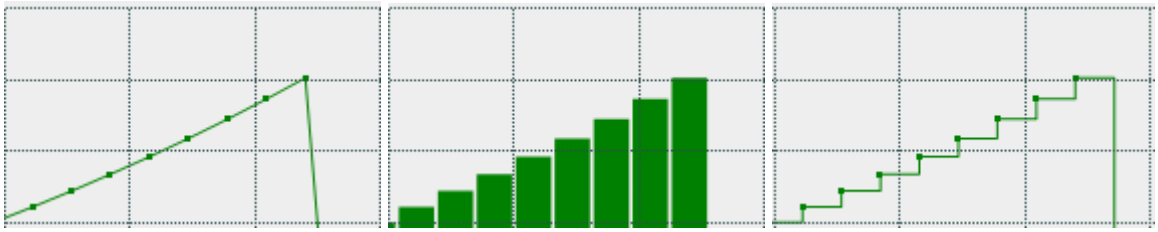
不论是否停止显示，都可以在 Chart 窗口的左边第一个格中设置显示宽度。



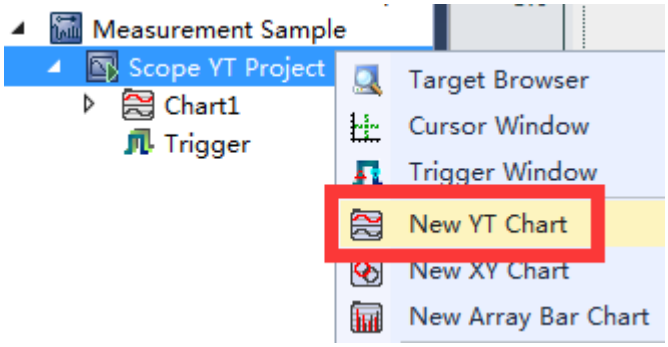
（12）为便于观察数值变化，可将 fPeak 这个尖波函数的显示类型作一些更改。选中 Axis1 下面对应的 fPeak 轴，在 Properties 选项卡下修改 Graph Type。

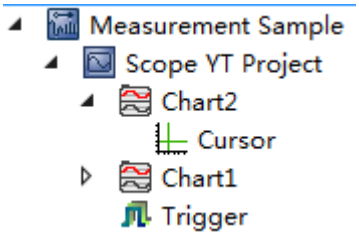
关于波形的显示模式有三种：Line（线型）、Bar（柱状图型）、Stair（阶梯型）。以

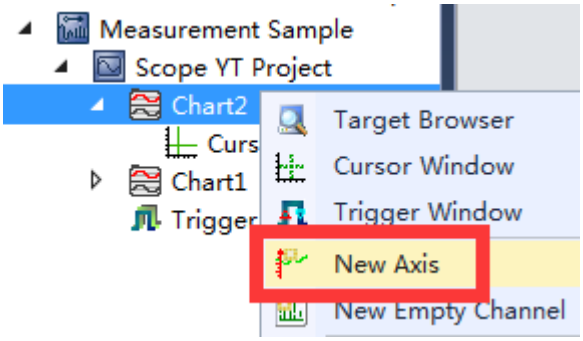
尖波信号为例放大观察具体值。

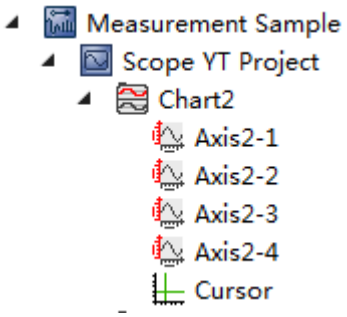


（13）再新建一个 Chart，并且添加四个 Axis，然后对应每个轴从 MAIN 程序中添加一个变量，分别添加：fStairs、fSquare、fSawtooth、fPulse

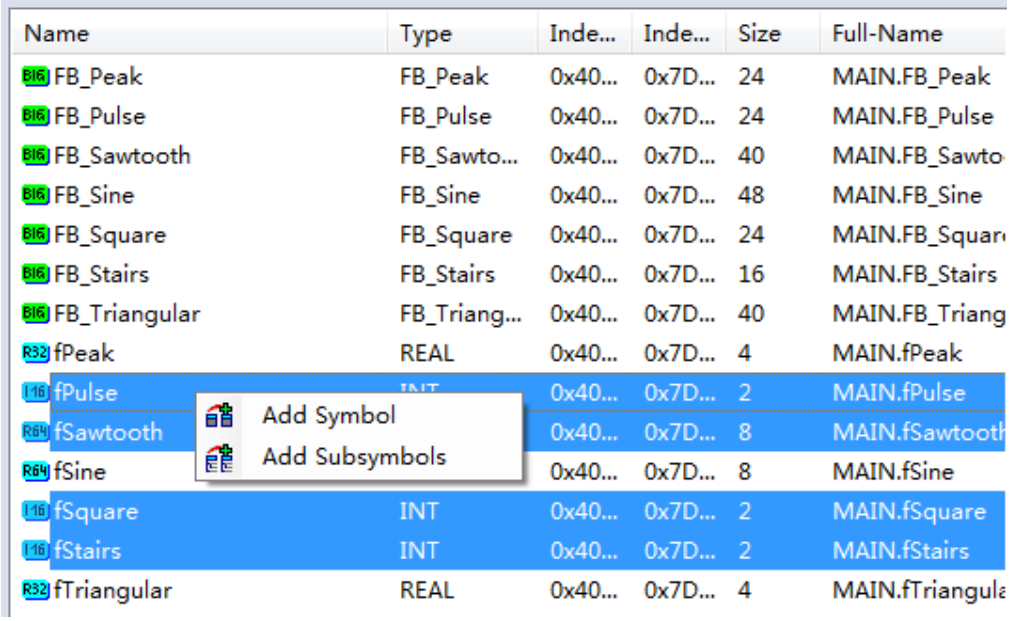




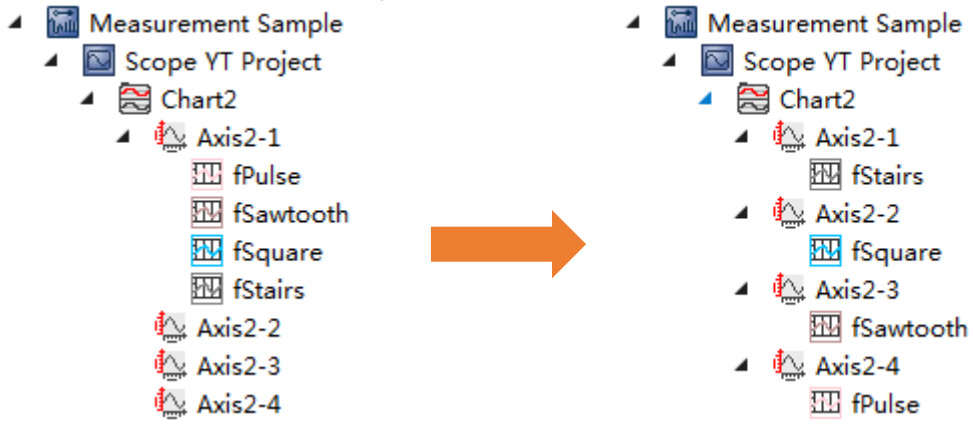




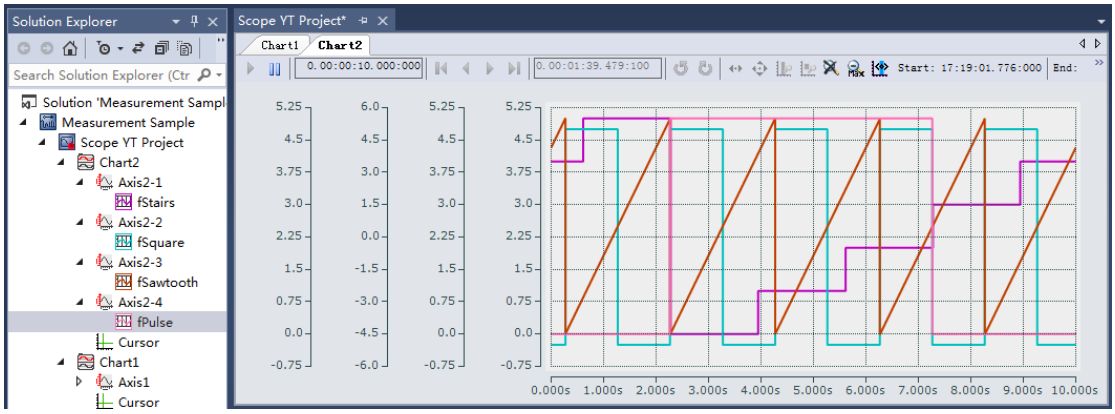
右键需要添加的变量 Add Symbol 或者双击。



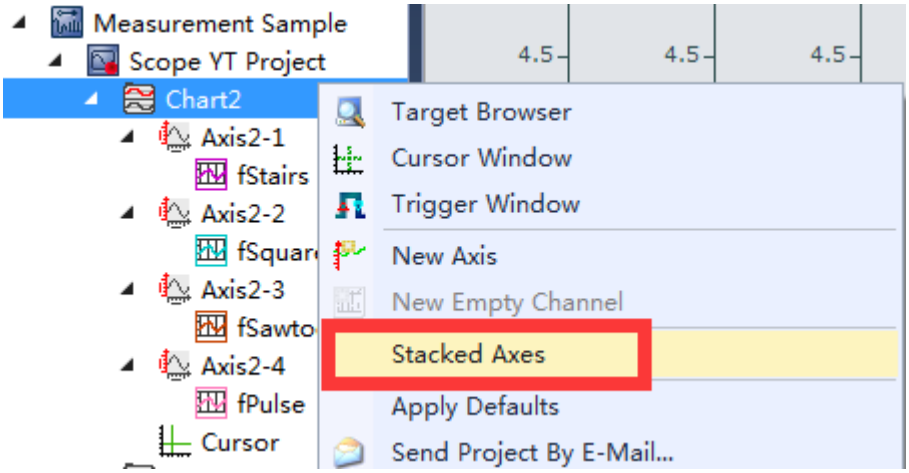
四个变量就被添加到了 Axis 中。手动把四个轴变量拖到独立轴里。



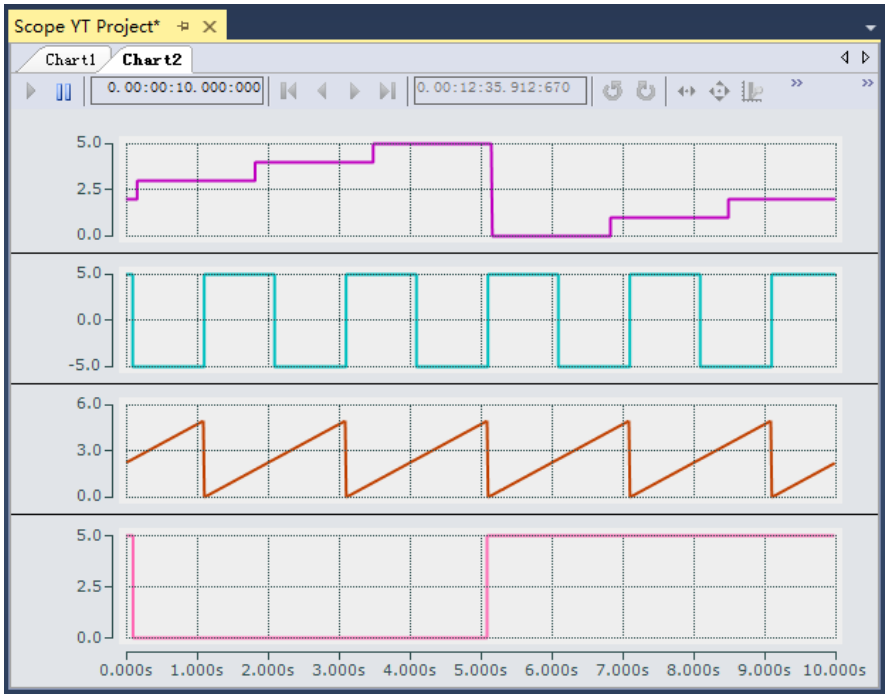
（14）可以开始观察变量曲线。



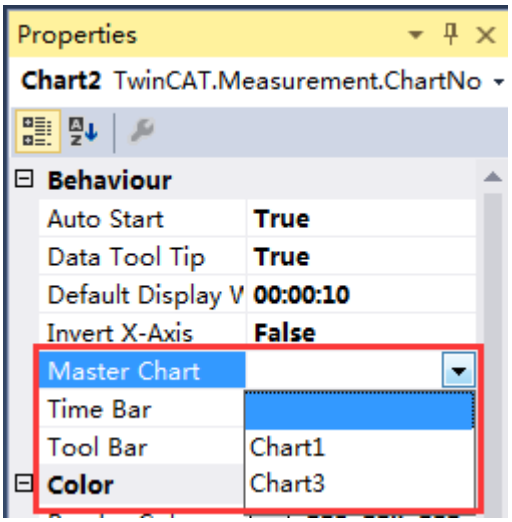
（15）并且独立轴之间可以做到 Y 轴相互独立查看。右键 Chart 选择 Stacked Axes。



就可以把四个轴独立成像，方便观测。

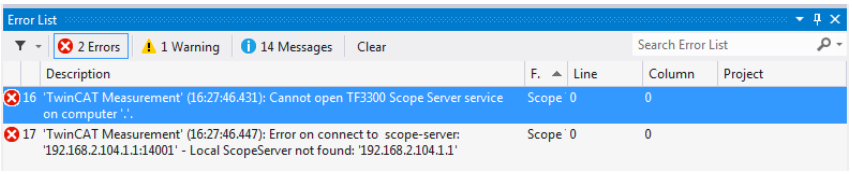


（16）对于多个 Chart 的情况，还可绑定主从轴。这样从轴的所有显示都会跟随主轴。以 Chart2 为例，选中 Chart2，在 Properties 选项卡中选择 Master Chart，然后可以绑定给 Chart1。

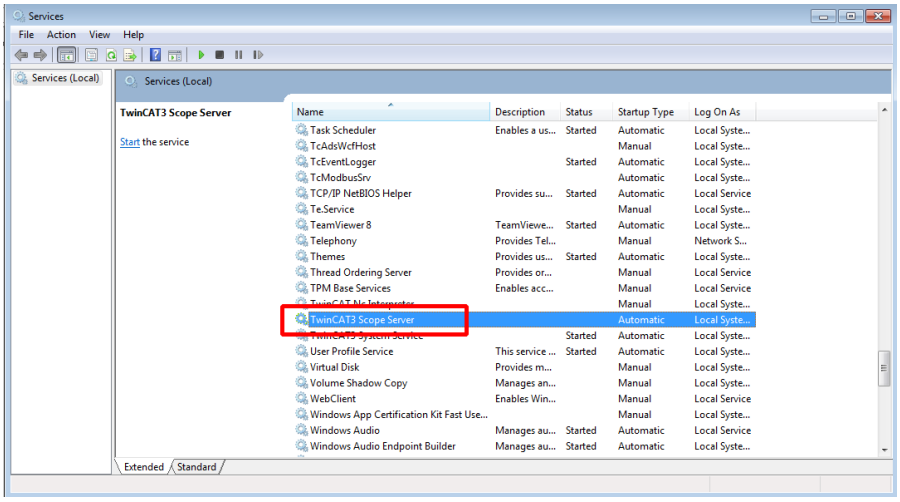


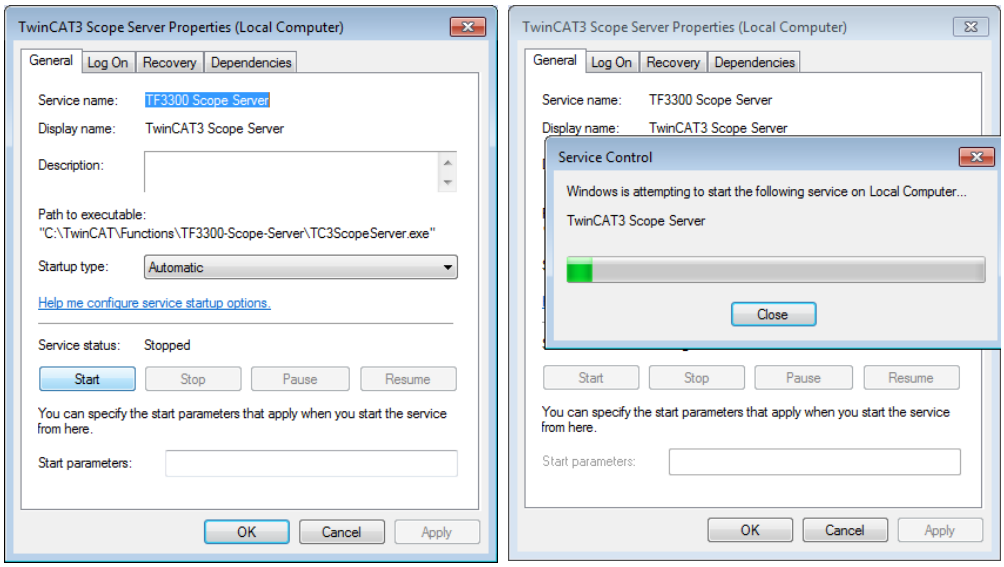
如此，Chart1 就变成主轴，如果 Chart1 停止显示，Chart2 也会停止。再比如，进行 Overview 查看时选择区域，两个 Chart 也是同步的。

附：如果在开始监控，也就是点击 record 后有以下报错。

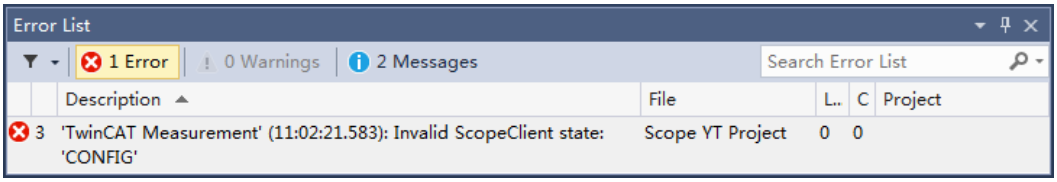


解决方式很简单，只需要在服务中把 TwinCAT3 scope service 手动开启就可以了。





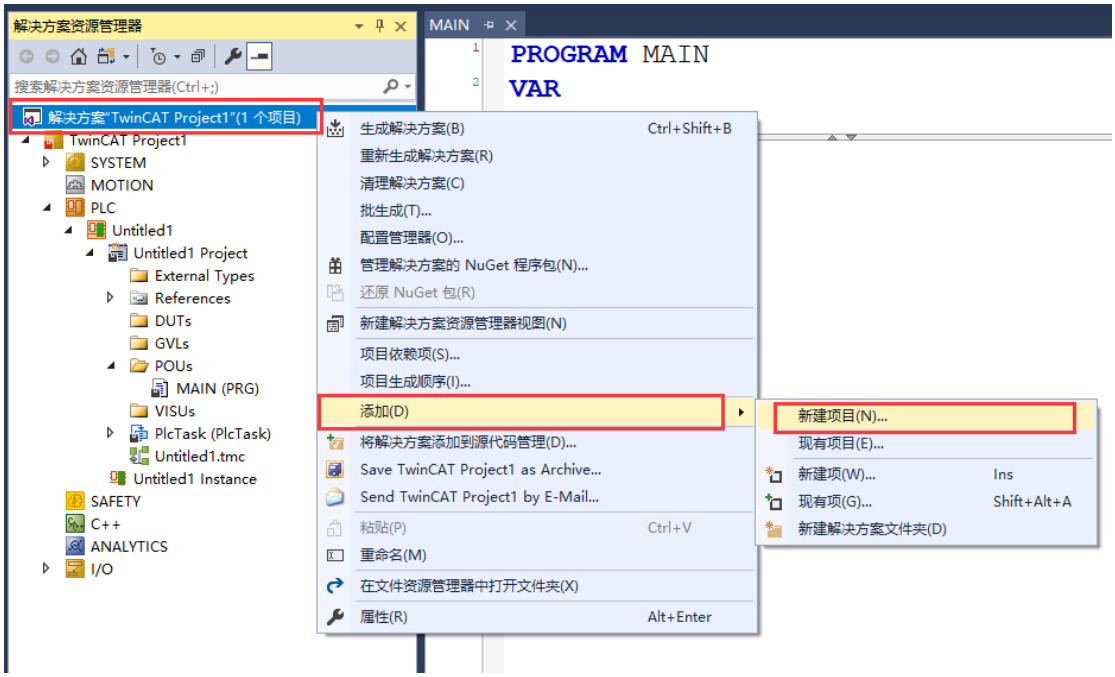
如果点击 record 后有以下报错，一般是原工程重新做过激活之后没有 Login 过，那么即便是处于 Run Mode 状态还是可能出现下面的报错。只要在重新激活后 Login 运行，就可以记录了。



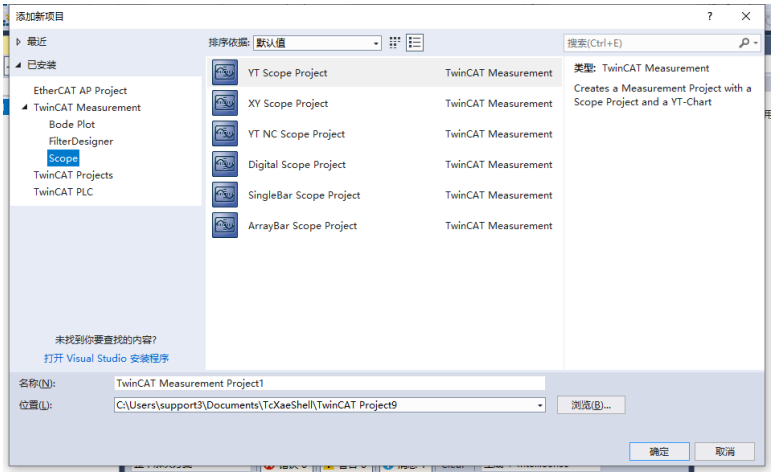
1.2、ScopeYT Project 使用（**TC3.1.4024 版本**）

本小节需要用到的程序如下，PLC 工程的新建和程序的编写此处不多做赘述。（程序部分同 1.1 节代码 请大家参考后完成功能块创建和具体代码编写。）

（1）首先新建一个 TwinCAT3 项目。“右键解决方案-> 添加-> 新建项目”



弹出窗口选择 TwinCAT Measurement，有很多工程模板可供选择。



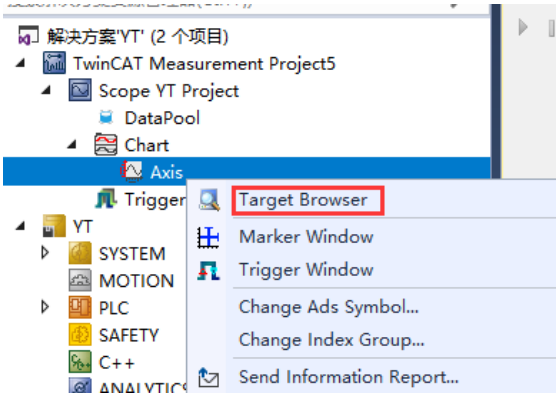
YT Scope project：单个变量随着时间变化而变化进行监控。
XY Scope project：横纵坐标都可以制定为变量，观察 2 个变量的曲线图。
YT NC Scope project：创建后自动添加 NC 轴中主要的一些变量，只需要指定相对于的 NC 轴的编号就可以直接进行监控。
Digital Scope Project：以数码管的方式显示变量的值。
SingleBar Scope Project：对单个变量的值使用柱状图显示。
Array Bar Scope Project：用柱状图监控数组变量。

此处以 Scope YT project 为例，在模板中选择 YT project 并且点击 OK 进行创建。

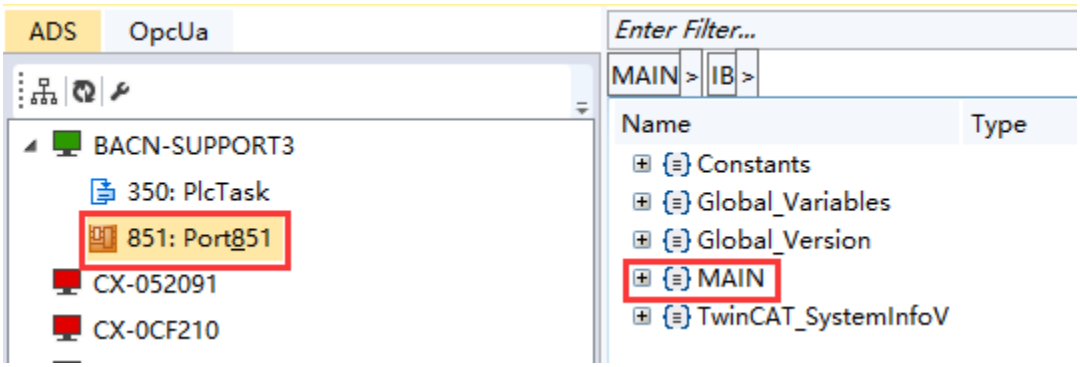
在开始添加变量观察变量之前，我们需要将源程序激活并进行下载运行。这么做

的目的是为了创建 851 端口，希望 Scope 能通过 ADS 协议获取到变量。编译无报错后，点击 Activate Configuration 并且切换到 Run Mode。点击 Login 和 Start，下载并运行程序。

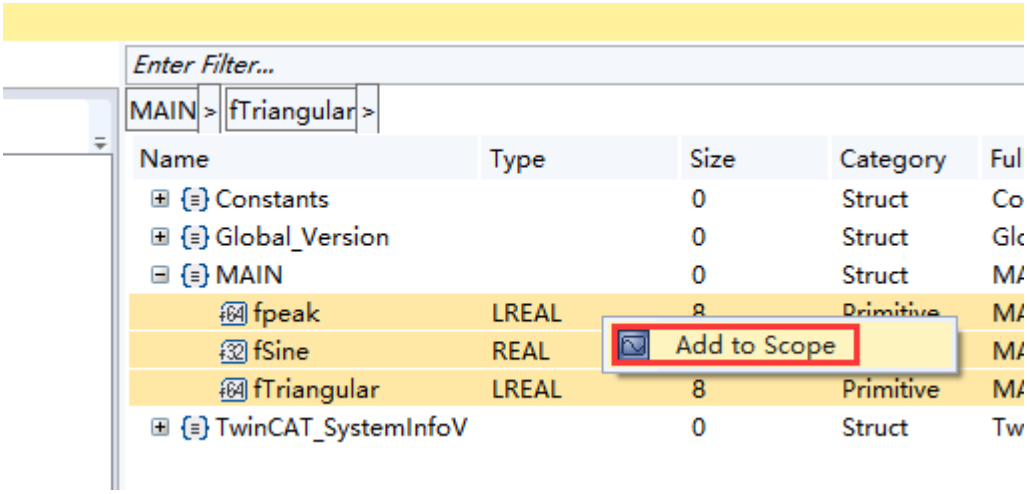
（2）右键 Axis 选择 Target Browser 进行在线搜索变量（只有 PLC 运行起来才可以找到变量）。



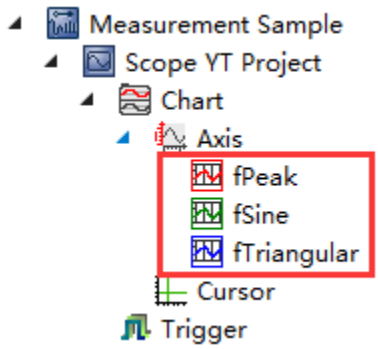
（3）选择对应的目标控制器，选择 PLC 端口号 851（在 TC3 中 PLC 端口号从 851-899，起始第一个是 851），找到 MAIN 程序中对应的三个变量：fTriangular、fSine、fPeak。



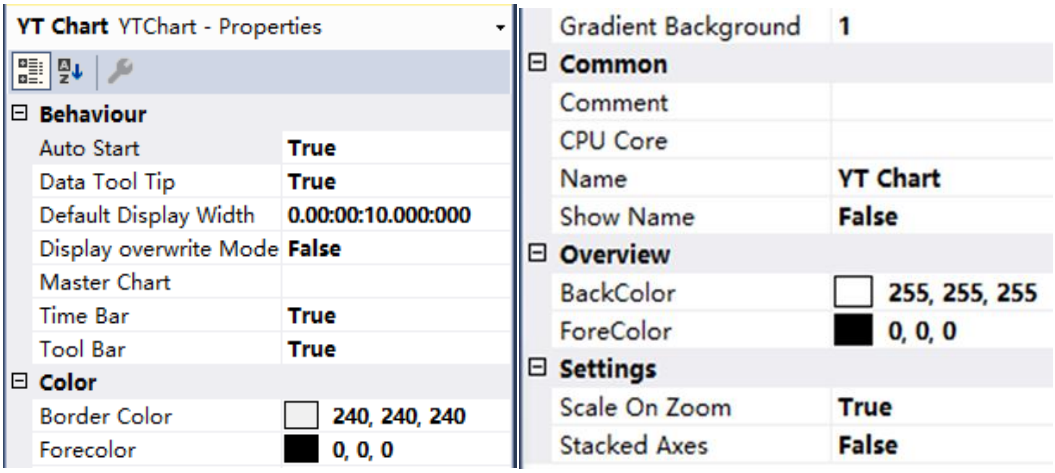
（4）右键需要添加的变量 Add to Scope 或者双击。



三个变量就被添加到了 Axis 中。



（5）在 Chart 的属性里，可对 Chart 窗口的一些属性进行更改：



Behavior 性能

Auto Start（自动开始）：决定当这个图表有新记录时是否启用即时模式。

Data Tool Tip（数据工具提示）：开启数据工具提示。

Default Display Time（默认显示时间）：这个时间是当记录开始时或重新缩放时，整张图表显示的时间跨度。

Display Overwrite Mode（显示覆盖模式）：设置是否启用显示覆盖模式。

Master Chart（主图标）：可绑定附属图表，达到观测同步。

Time Bar（时间条）：在图表中显示或隐藏时间条。

Tool Bar（工具条）：在图表中显示或隐藏工具条。

Color 颜色

Border Color（边框颜色）：图表框架颜色。

Forecolor（前景色）：修改图表的前景色。

Gradient Background（梯度背景）：设置图表的梯度背景。

Common 普通

Comment（注释）：在此输入注释。

CPU Core（CPU 核）：将对应 chart 的记录工作分配到不同核里去。默认则由 windows 系统分配。

Name（名称）：图标名称

Show Name（显示名称）：在绘图区显示或隐藏图标名称。

Overview：图表概况

BackColor（背景颜色）：修改图表的背景色彩。

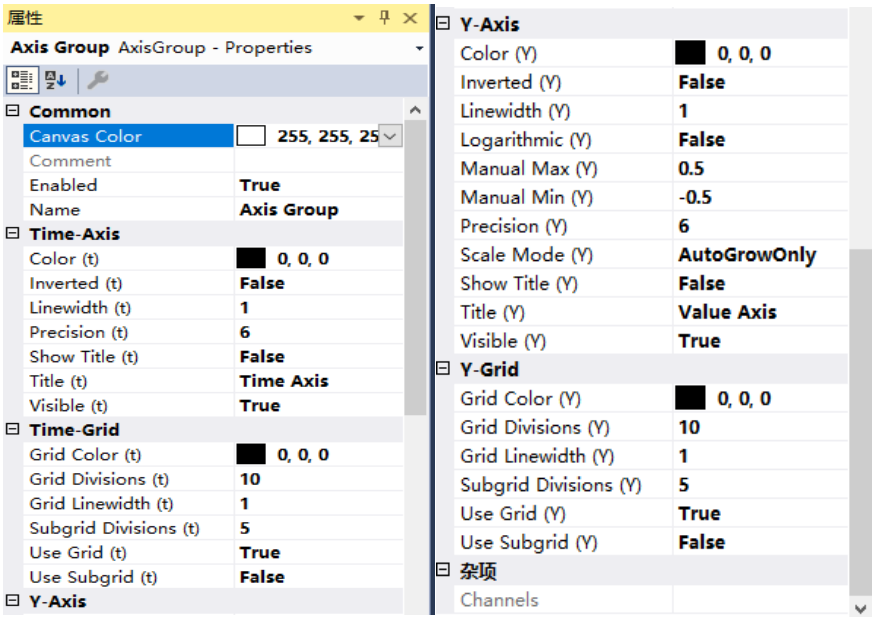
ForeColor（前景颜色）：修改图表的前景颜色。

Settings （Y 轴设定）

Scale on Zoom（Y 轴缩放）：如果选 true 则图表会在缩放后引导所有轴遵循自动比例。

Stacked Axes（Y 轴堆叠）：如果图表中添加了多于 1 条 Y 轴，将其置为 true 后可改变关联显示区域的布置，区域会被改变为上下叠放。如果未选择堆叠选项，轴将会使用相同区域来显示通道的值。

（6）在 Axis 的属性选项卡里，可对 Axis 的一些属性进行更改：



Common 普通

Canvas Color（画布颜色）：修改表格绘图区颜色。

Comment（注释）：在此输入注释。

Enabled（使能）：开启或关闭轴的记录。

Time-Aixs(时间轴)

Color(t)（颜色）：修改时间轴颜色。

Inverted(t)（T 轴反转）：决定 T 轴是由小到大还是由大到小。

Linewidth(t)（线宽）：调整 T 轴线宽。

Precison(t)（精度）：调整 T 轴精度。

Show Title(t)（显示标题）:选择是否显示标题。

Title(t)（标题）：设置 T 轴标题。

Visible(t)（可视化）；选择 T 轴是否可见。

Time-Grid:T 轴网格

Grid Color（t）:设置 T 轴网格颜色

Grid Divisons（t）（网格分隔数）:设置 T 轴大网格的分隔数。

Grid Linewidth（t）（网格线宽）:设置网格线宽。

Subgrid Divisions(t)（子网格分隔数）：只在子网格开启时有效，具体将 T 轴大网格细分的格数。

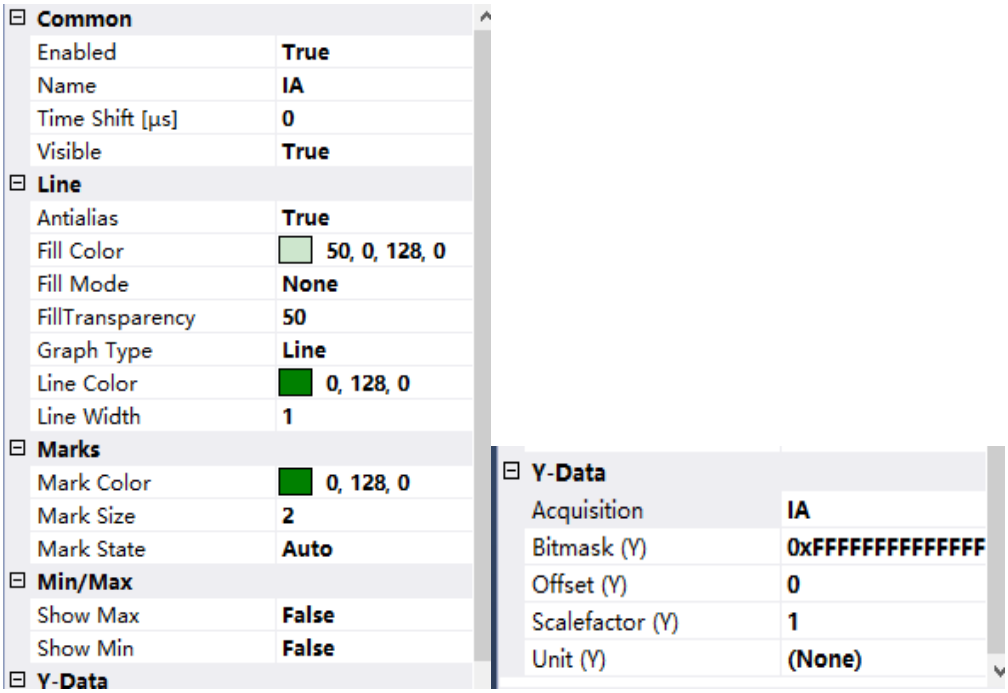
Use Grid(t) （网格）：开启或关闭 Y 轴网格。

Use SubGrid（子网格）：开启或关闭 Y 轴子网格。

Y-Axis(Y 轴)

Color（颜色） :设置 Y 轴颜色。
Lnverted（反向）:设置 Y 轴是否反向。
Linewidth (Y)（线宽）:设置 Y 轴线宽。
Logarithmic (Y)（对数模式）:设置 Y 轴是否为对数模式。
Manual Max (Y)（手动最大值）:可手动设置最大值。
Manual Min (Y)（手动最小值）:可手动设置最小值。
Precision (Y)（精确度）:设置 Y 轴精确度。
Scale Mode (Y)（模式选择）：可选择 Y 轴刻度模式：AutoGrowonly（自动增长模式）、AutoGrowNShrink（自动收缩模式）、Manual（手动模式）
Show Title (Y)（显示标题）:选择是否显示标题。
Title (Y) （标题）:设置标题
Visible (Y)（可视化）:设置 Y 轴是否可见。
Y-Grid（Y 轴网格）
Grid Color (Y)（网格颜色）：设置 Y 轴网格颜色。
Grid Divisions (Y)（网格分隔数）:设置 Y 轴大网格的分隔数。
Grid Linewidth (Y)(网格线宽)：设置 Y 轴网格线宽。
Subgrid Divisions (Y) （子网格分隔数）：只在子网格开启时有效，具体将 Y 轴大网格细分的格数。
Use Grid (Y)（使用网格）：设置是否使用网格。
Use Subgrid (Y)（使用子网格）；设置是否使用子网格。

（7） 在变量的属性里，可对变量的一些属性进行更改：



Common（通常设置）
Enabled（变量记录使能）：所观察变量是否记录，只能在 Stop Record 之后才可以更改。

Name（名称）：变量名称。

Time Shift（时移）：时移是时间线上的偏移量。它有助于对已知停止时间的通道进行比较。

Visible（可视化）：选择变量是否可视。

Line（线条）

Antialias（平滑）：平滑属性决定了曲线在绘制过程中是以美观（平滑度）或快速为主。

Fill Color（颜色）：在画布上可以选择区域进行色彩填充以示区分。此处可选填充色。

Fill Mode（填色模式）：四种模式，None 无填充色；Horizontal Zero 以零做分割线，包含在曲线和零线之间的部分均会被填充；Bottom 所有在曲线以下的部分均会被填充；Top 所有在曲线以上的部分均会被填充。

FillTransParency（透明度）：填色透明度。

Line Color Line Color（线色）：曲线的颜色。

Line Width（线宽度）：曲线的像素线宽。

Mark：标记

Mark Color：标记颜色

Mark Size（标记尺寸）：可设置标记的尺寸大小

Mark State（标记状态）：可设置为 auto、off、on 状态

Min/Max（最大/最小值）

Show Max（显示最大值）：设置是否显示最大值。

Show Min（显示最小值）：设置是否显示最小值。

Y-Data（Y 轴数据）

Bitmask (Y) （位掩码）：位掩码用于从通道值中剪裁单个比特位。显示的值是通道值与位掩码相与后的结果。浮点型的值不能被任何掩码改变。这个选项在观察状态字的单独位时十分有用。

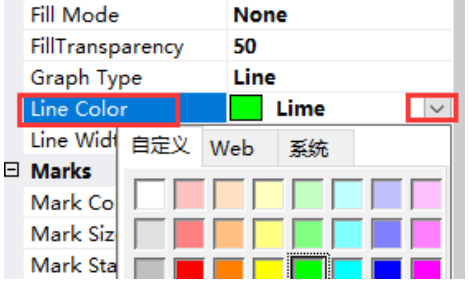
Offset (Y) （偏移）：将给通道添加一个偏移量使其在图表中安排得更合理。

Scalefactor (Y) （刻度系数）：通过将刻度系数设为 1 可以给显示的通道值标上刻度。这个功能对于像以可读性更好的角度制来显示弧度制的角度值很有帮助。如 Scale = 360/(2*Pi) = 57.296。

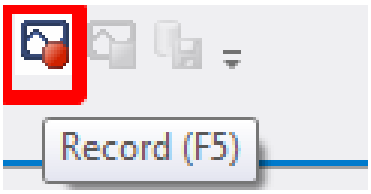
Unit (Y)（单位）：设置变量的单位.

比较常用到的比如 Acquisition、，可以直接更改变量，可以改为 main.b，那么监测的就是 main 函数里 b 变量的值。

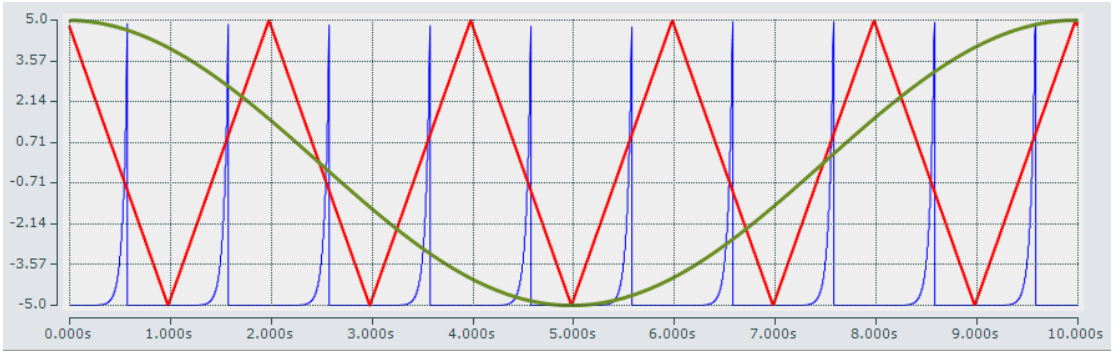
或者是 line 的属性，颜色和线宽，TwinCat3 会自动给不同变量不同的颜色以示区分，我们也可以在这里手动更改，如果觉得线宽默认 1 太细，也可以往上加。



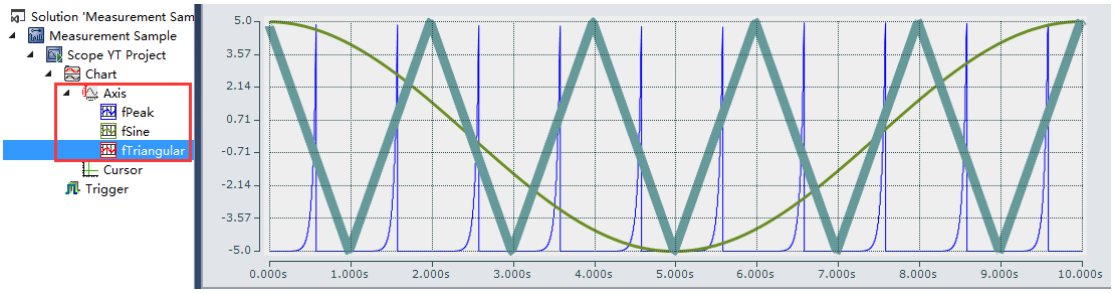
（8）点击工具栏中的 record。此时 PLC 程序要在运行当中才能采集到实际值。



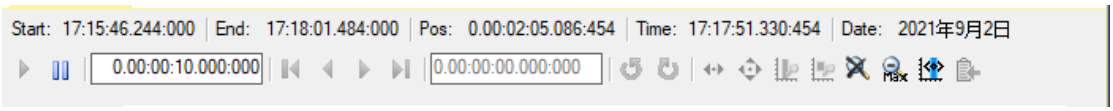
（9）可以开始观察变量曲线。



（10）双击 axis 或者 axis 下的变量，TwinCat scope 就会高亮显示变量曲线，方便在多变量时候区别出来。



（11）Chart 工具栏包含了操纵连接数据的主要命令。以下是按照如下所示图片从左到右的分类解释：



最上方的内容分别是开始时间，结束时间，当前位置，以及记录的时间

下方的内容从左往右分别是

Play（开始）：当 Scope 在记录模式时，开始即时显示模式。当前的帧是标绘在图表上的。

Pause（暂停）：停止计时绘图，对数据进行操作。

Display-Width（显示宽度）：以"hh:mm:ss.fff"的格式改变显示宽度（X 轴）。您也可以用鼠标来改变显示宽度：选中一张图表，按下 <Ctrl>转动滚轮。当你放开 <Ctrl>时画面将会刷新显示宽度。

Scroll Buttons（滚动按钮）：两边的滚动按钮按整个画面的宽度移动。中间的两个的按钮以较小幅度滚动。

Position（位置）：以 "hh:mm:ss.fff"的格式在此设定图表的当前位置。

Undo/ Redo Time/Position（撤销/恢复 时间/位置）：按发生的次序撤销位置或显示宽度的改变。也可使用右键。当一个动作被撤销，可以通过恢复选项将它复原。

Zoom-Mode（缩放模式）：当缩放模式选中时，您可以用鼠标缩放选中区域。在 Chart 窗口中选择 Y-Zoom” 选项可以激活 Y 方向缩放。

Panning Horizontal（水平平移）：在图表上单击并拖动鼠标以在 X 方向上移动当前范围。

Panning Free（自由平移）：在图表上单击并拖动鼠标以在 X 或 Y 方向上移动当前范围。

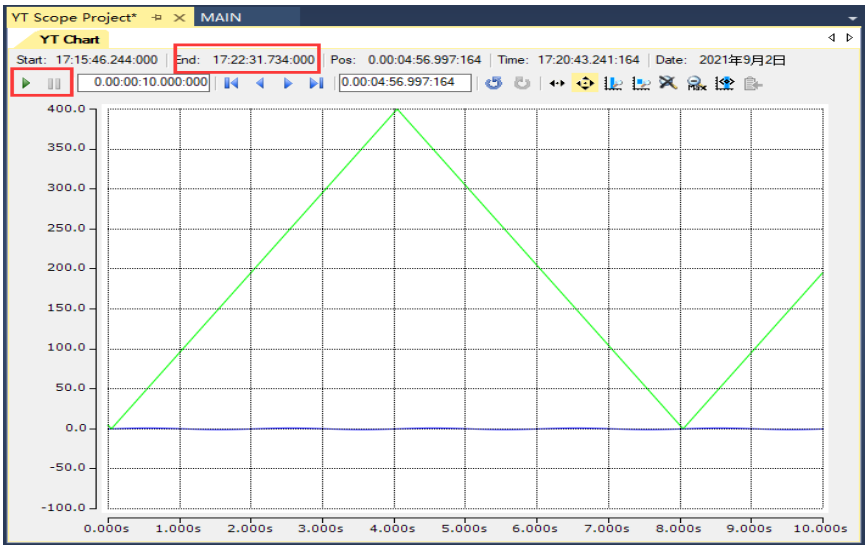
Zoom Horizontal（水平缩放）：使用鼠标在 x 轴上跨越一定范围以显示新的时间夸度。

Zoom Free（自由缩放）：在图表上使用鼠标跨越一定范围缩放至当前画面。

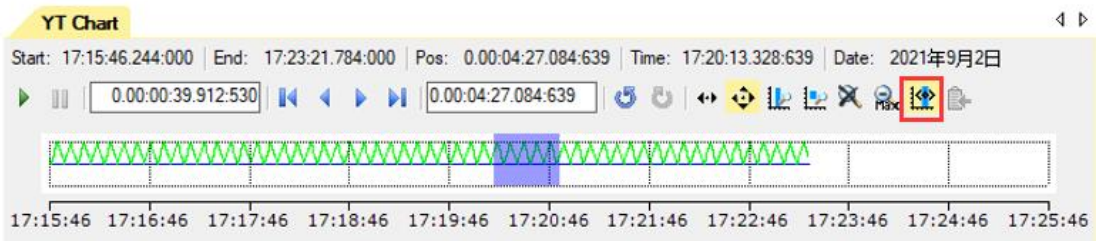
Rescale All（重缩放）： 调整所有 Y 轴重新缩放。X 轴设定成默认显示宽度。

Zoom Out Max（全部缩放）：将 X 轴范围设置成一个较大的时间跨度或等于当前记录长度。

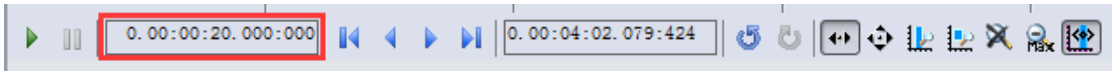
比如在下图中，点击 toolbar 最左侧是 Start/Stop Display 按钮，位于下方的图表停止了，而记录仍在后台运行，end 值一直在增加。之后可以对画面做一些操作，例如可以设置游标或对曲线进行缩放。



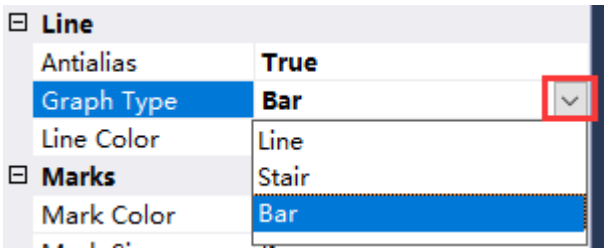
toolbar 右侧有个 Overview 功能，可以多出一个轴查看整个监控过程，从中可以选择一部分进行查看。



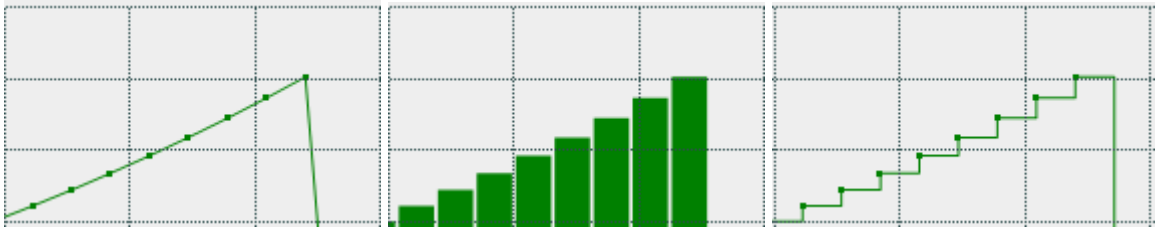
不论是否停止显示，都可以在 Chart 窗口的左边第一个格中设置显示宽度。



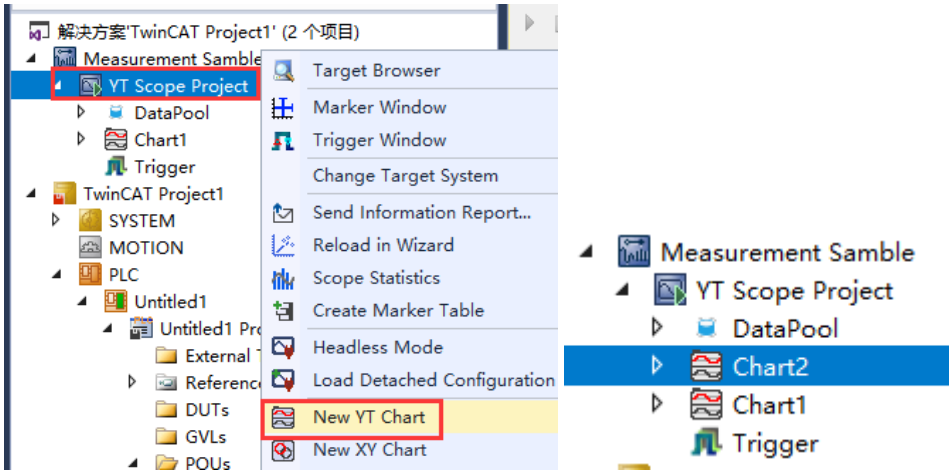
（12）为便于观察数值变化，可将 fPeak 这个尖波函数的显示类型作一些更改。选中 Axis1 下面对应的 fPeak 轴，在 Properties 选项卡下修改 Graph Type。

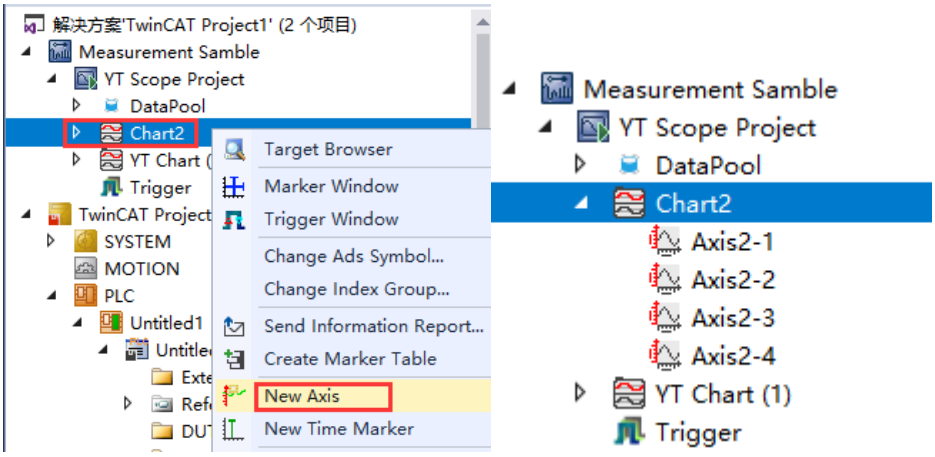


关于波形的显示模式有三种：Line（线型）、Bar（柱状图型）、Stair（阶梯型）。以尖波信号为例放大观察具体值。



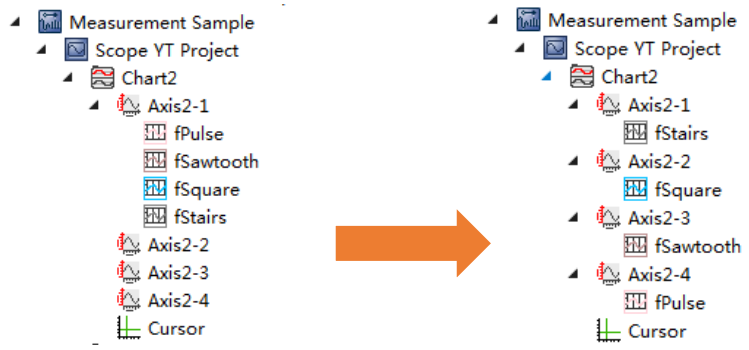
（12） 再新建一个 Chart，并且添加四个 Axis，然后对应每个轴从 MAIN 程序中添加一个变量，分别添加：fStairs、fSquare、fSawtooth、fPulse



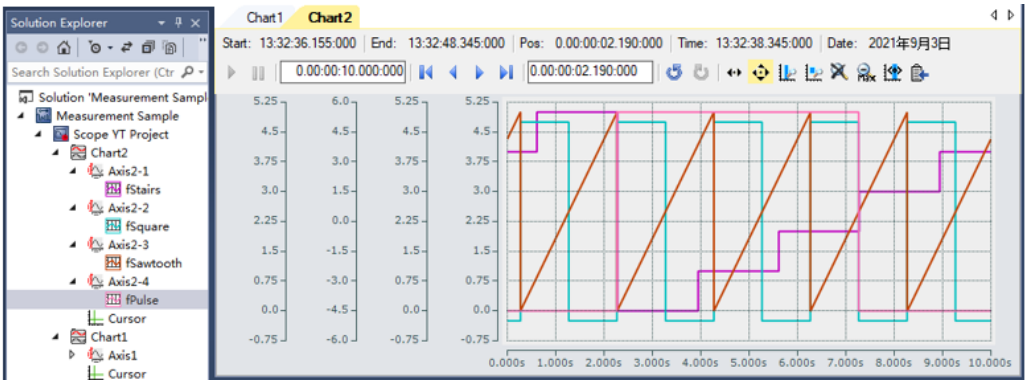


按照先前所示添加变量的步骤添加变量

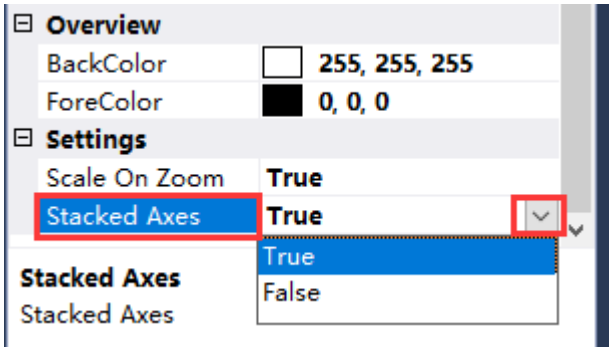
四个变量被添加到了 Axis 后。手动把四个轴变量拖到独立轴里。



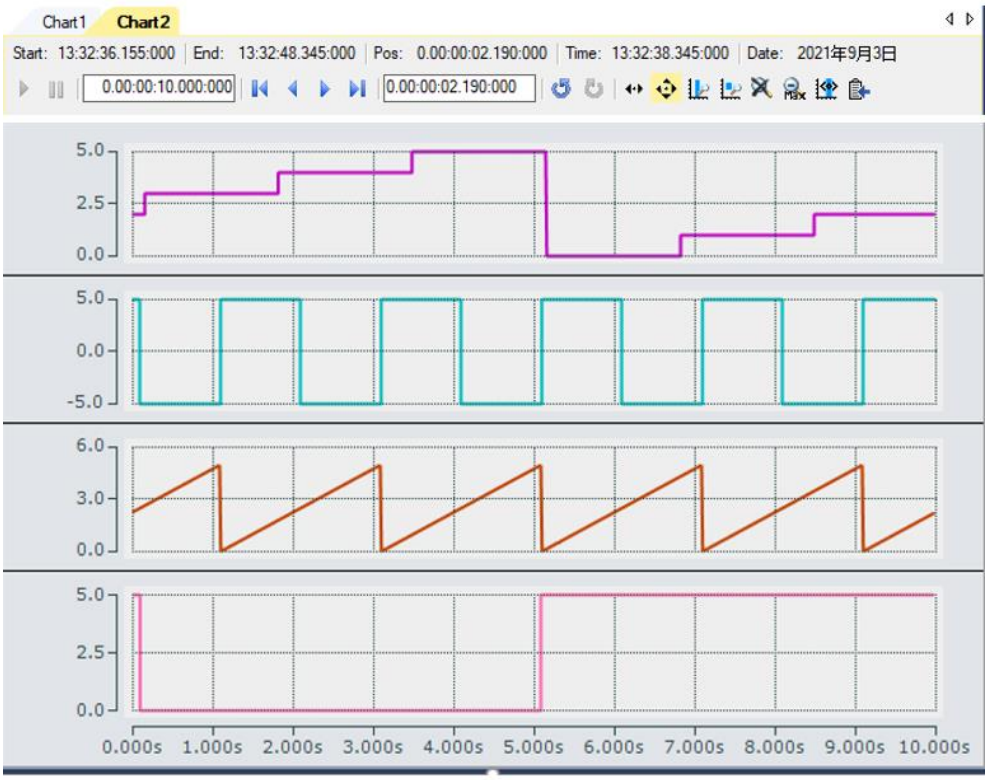
（14）可以开始观察变量曲线。



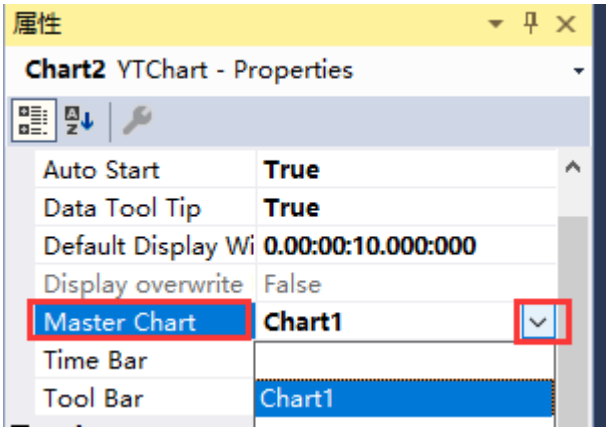
（15）并且独立轴之间可以做到 Y 轴相互独立查看。在 Chart 属性中选择 Stacked Axes，点击右边的下拉按钮后选择 true。



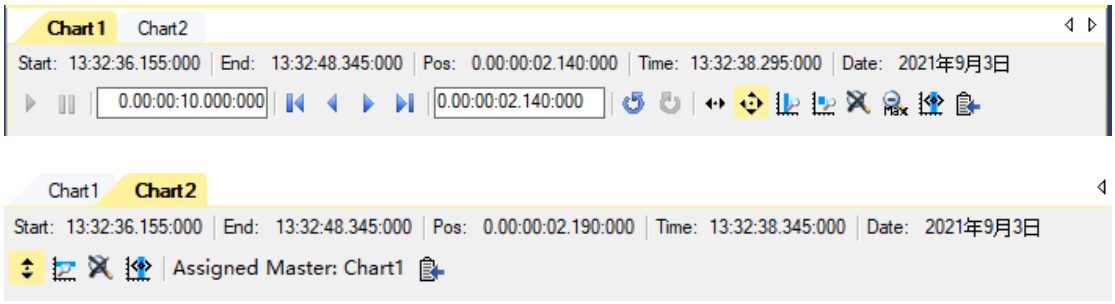
就可以把四个轴独立成像，方便观测。



（16）对于多个 Chart 的情况，还可绑定主从轴。这样从轴的所有显示都会跟随主轴。以 Chart2 为例，选中 Chart2，在 Properties 选项卡中选择 Master Chart，然后可以绑定给 Chart1。



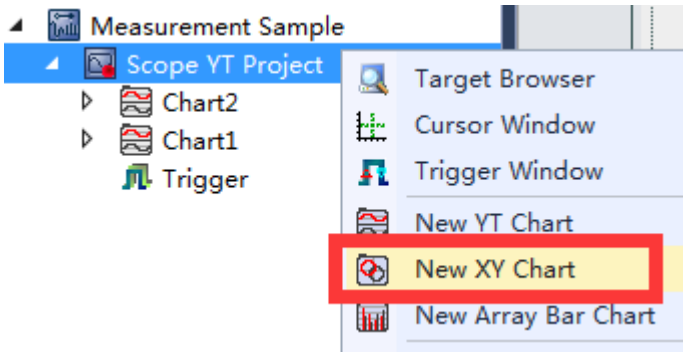
如此，Chart1 就变成主轴，如果 Chart1 停止显示，Chart2 也会停止。再比如，进行 Overview 查看时选择区域，两个 Chart 也是同步的。



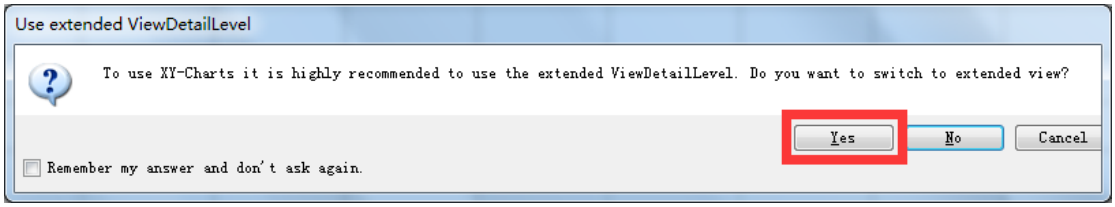
1.3 、ScopeXY Project 使用

接下来使用 Scope XY project，XY 都以 t 为自变量。

（1） 再新建一个 XY 轴，对应 X 轴 Y 轴添加变量 fSine 和 fTriangular

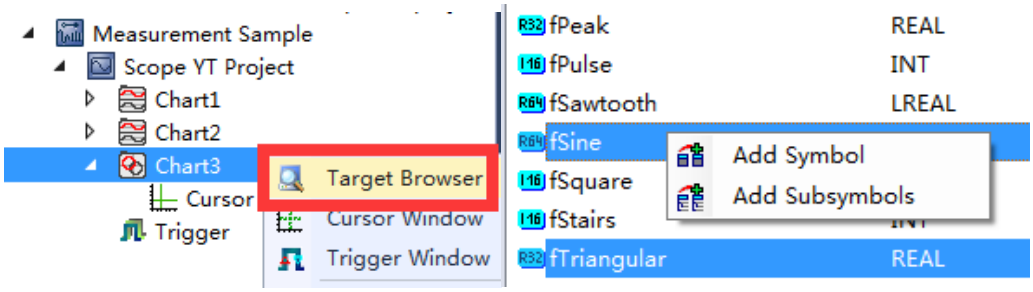


添加时提示 XY 模板需要扩展视觉等级。选择 Yes 进行切换，则将每个轴都更新成 XY 轴模板。

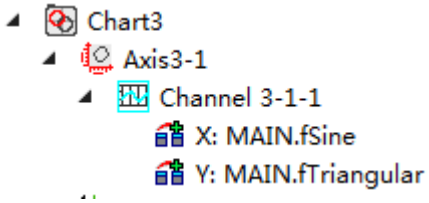


这个模板和我们在项目中新建的 XY project 是一样的。

（2） 在 Chart3 中添加变量 fSine 和 fTriangular，两者分别被添加至 X 轴和 Y 轴。

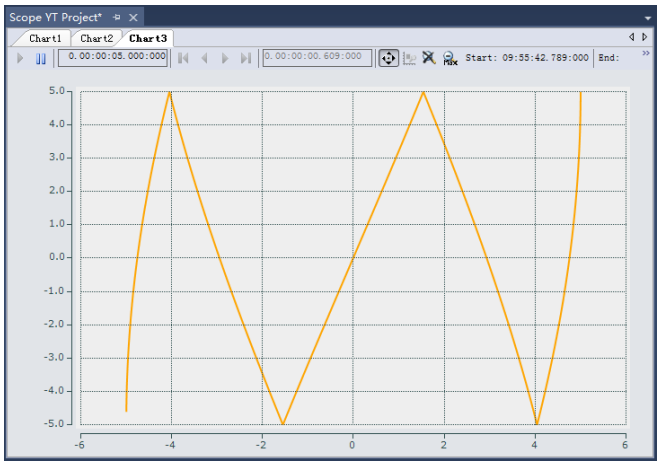


可以看到在 0 通道下出现了 XY 两个变量。不同于 YT 单个 Y 变量。

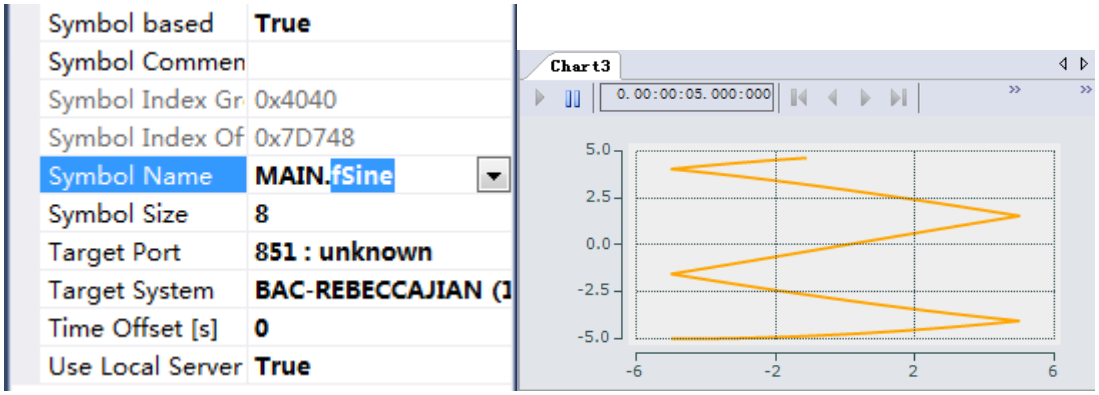


（3） 将显示时间改为 5s，现在可以开始观察变量曲线。

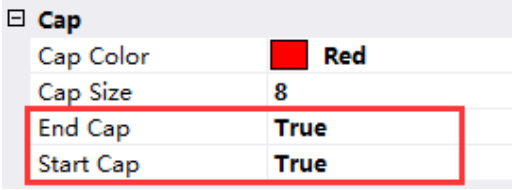
（4）



（5） 如果需要坐标轴对换，只要选中对应的 X 轴或者 Y 轴变量，在 Properties 选项卡中 Symbol Name 项进行修改即可。对换后的监控效果如下：



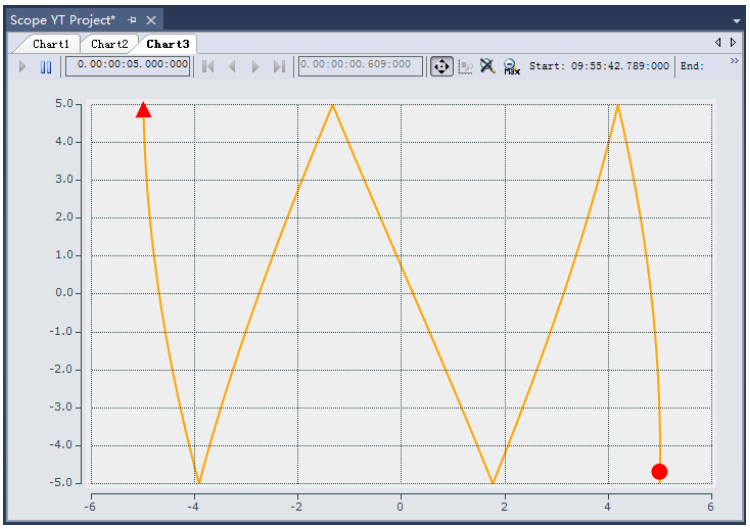
（6） 在这种不能明确分辨起始点和结束点的情况下，可以给图形添加对应的标识。选中他的 Channel，在右边的 Properties 窗口找到 Cap 功能展开。



将 End Cap 和 Start Cap 置 True。

还可以对 Cap Size 进行设置，使标识更加清晰。

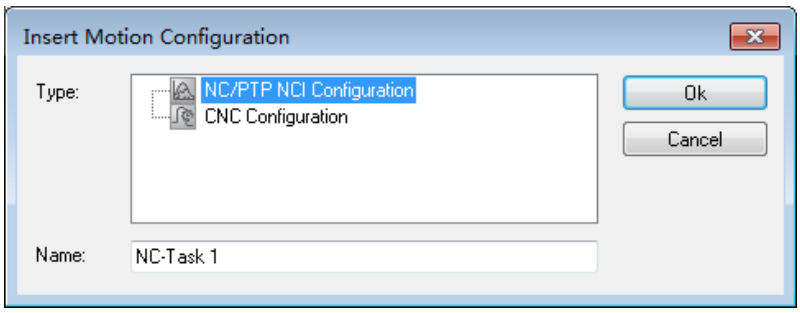
变量曲线会有如下显示。



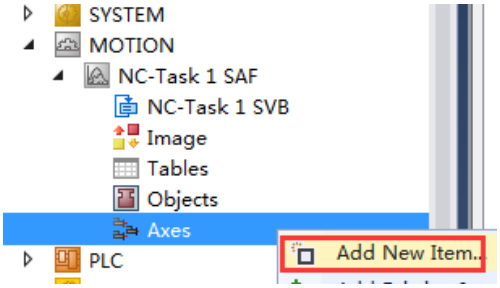
1.2 接下来建立 Scope YT NC project

（1） 利用编程手段让轴自己运动，这里先在 motion 新建虚轴模拟实际电机使用。

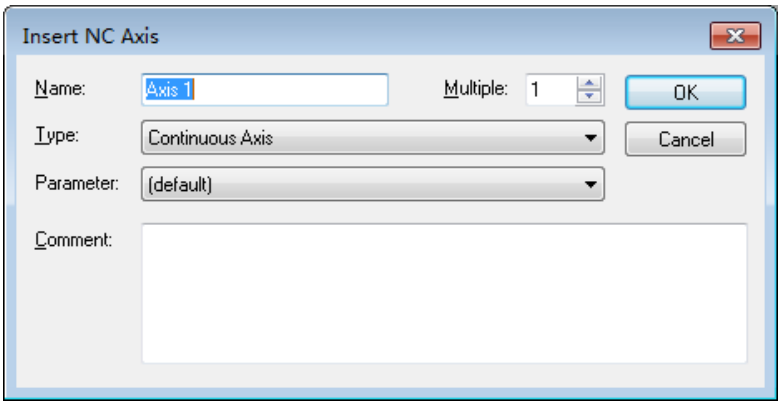
右键 motion 出现对话框，选择 NC 确定。



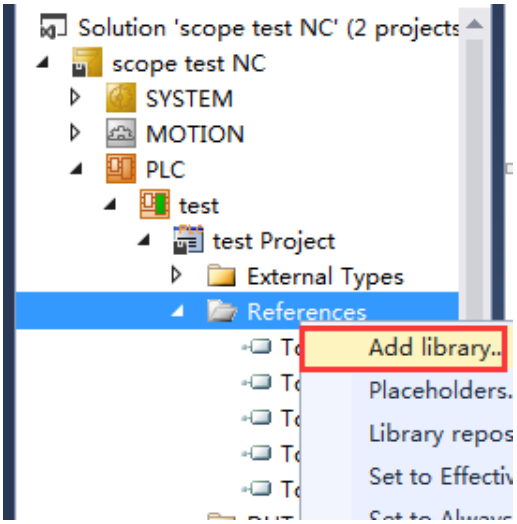
右键 axes，选择 new item。



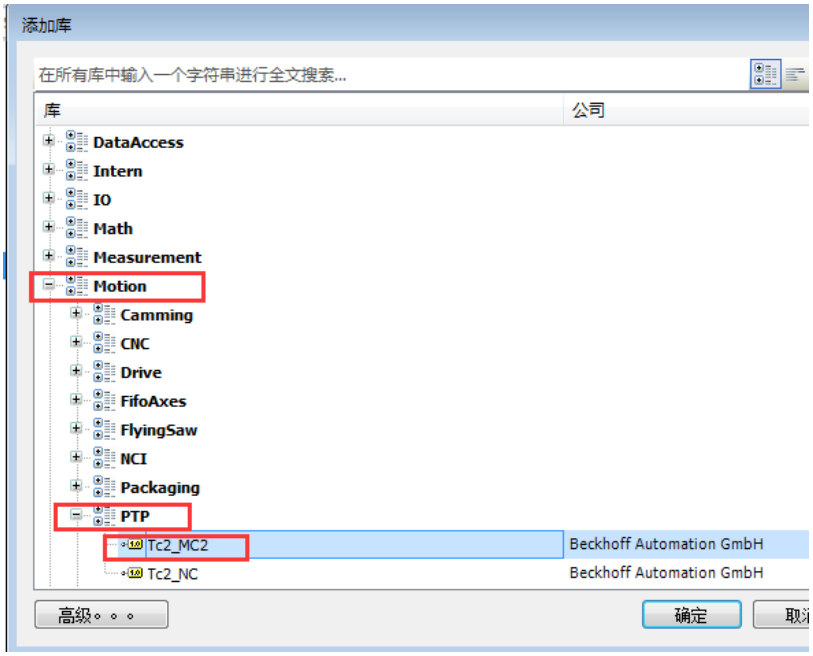
确定新建一个 axis1 轴。



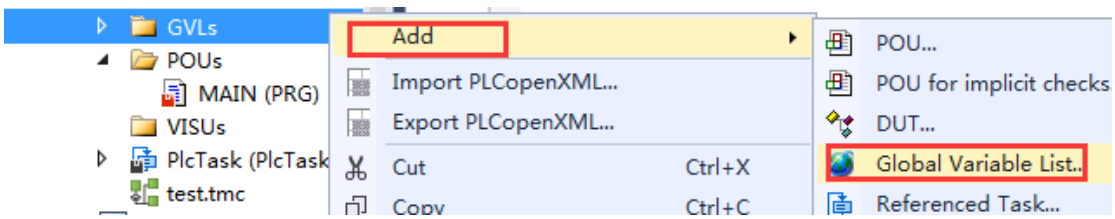
（2） 然后在 PLC 里新建项目。然后添加 Tc2_MC2 库。



点击 motion>PTP>Tc2_MC2，确定。



在 GVL 下添加一个 global variable list，

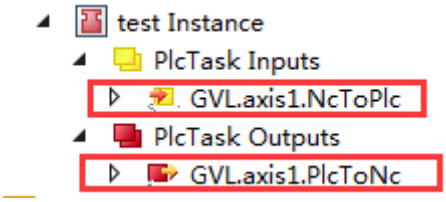


在这里定义一个全局变量 axis1，用来和 motion 的 axis 连接起到用程序控制

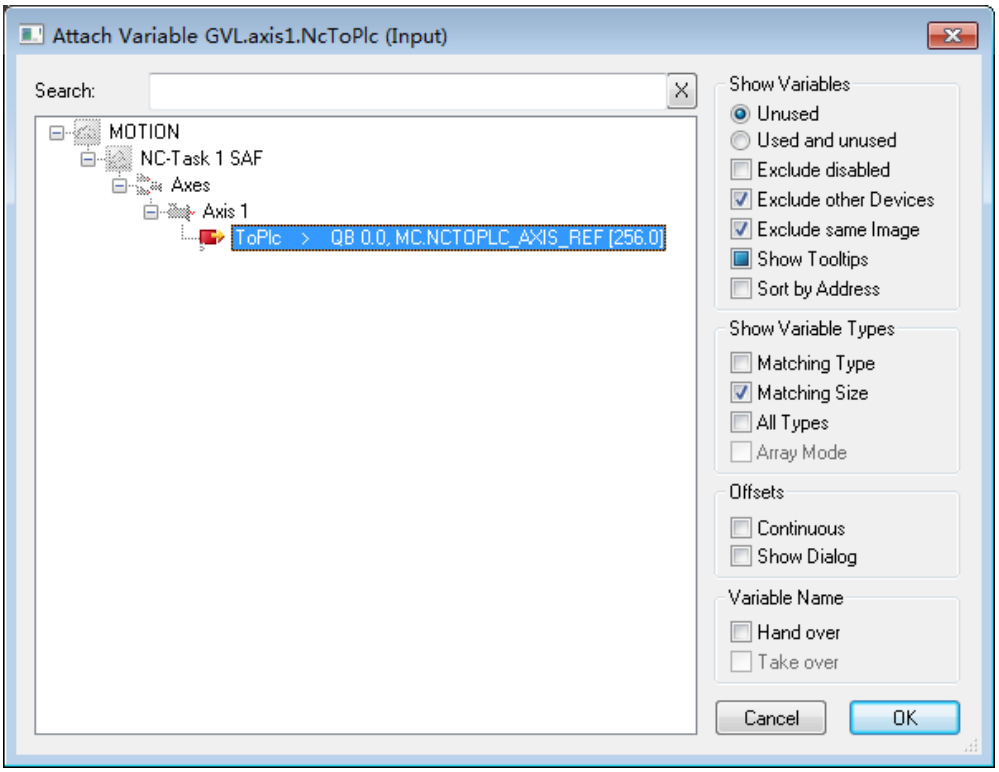
轴的作用。

```
VAR_GLOBAL
    axis1:AXIS_REF;

END_VAR
```

在 project 展开的项目中找到 project Instance ，展开它后会有一个 Inputs 和一个 Outputs，都是 Plc Task 的。



双击第一个 input 下的.NcToPlc 文件，选择刚才新建的虚轴。output 下的.NcToPlc 文件同理。



（3） 在 PLC 主程序部分，写入电机使能和简单运动的功能块，使轴能够以一定的速度前进。

Scope YT NC Project | MAIN | scope test NC | GVL

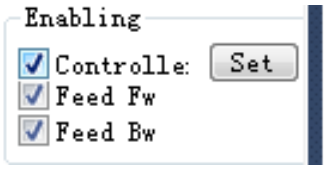
```
PROGRAM MAIN
VAR
    axis1_en:MC_Power;
    axis1_en_cmd: BOOL;
    axis1_jog:MC_Jog;
    axis1_jog_f: BOOL;
    axis1_jog_b: BOOL;
```

```
axis1_en(
    Axis:= axis1,
    Enable:=axis1_en_cmd ,
    Enable_Positive:= TRUE,
    Enable_Negative:= TRUE,
    Override:= 100,
    BufferMode:= ,
    Options:= ,
    Status=> ,
    Busy=> ,
    Active=> ,
    Error=> ,
    ErrorID=> );
axis1_jog(
    Axis:=axis1 ,
    JogForward:=axis1_jog_f ,
    JogBackwards:= axis1_jog_b,
    Mode:= MC_JOGMODE_CONTINOUS,
    Position:= 1000,
    Velocity:= 60,
    Acceleration:= ,
```

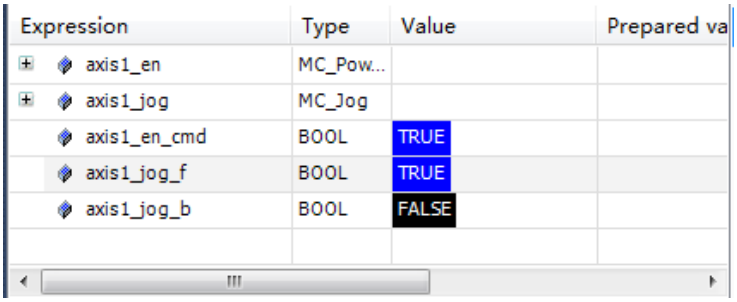
（4） 分别点击 login 和 start，测试使能程序和 jog 程序。

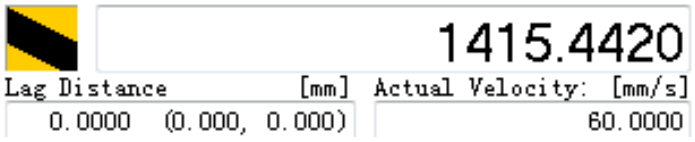
当 axis1_en_cmd 变为 true 时，axis1 被使能。

Expression	Type	Value	Prepared valu
axis1_en	MC_Pow...		
axis1_jog	MC_Jog		
axis1_en_cmd	BOOL	TRUE	
axis1_jog_f	BOOL	FALSE	
axis1_jog_b	BOOL	FALSE	

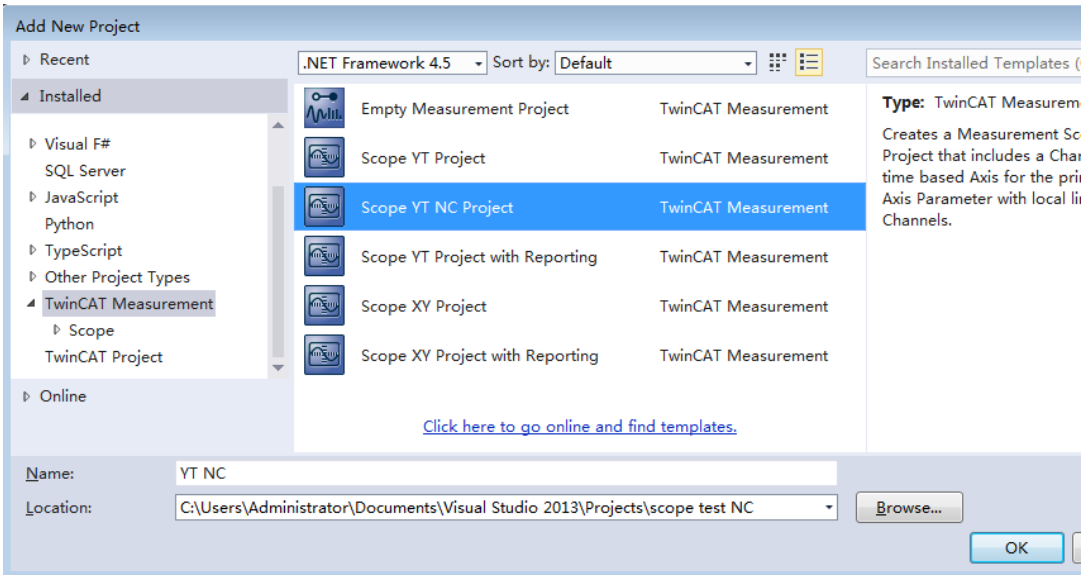


将 axis1_jog_f 置为 true 然后看 axis1 的参数变化。Axis1 轴以设定的速度运动。

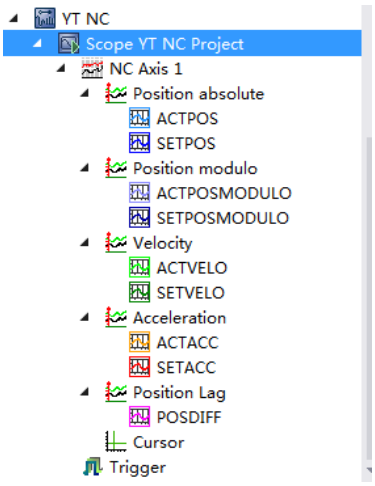




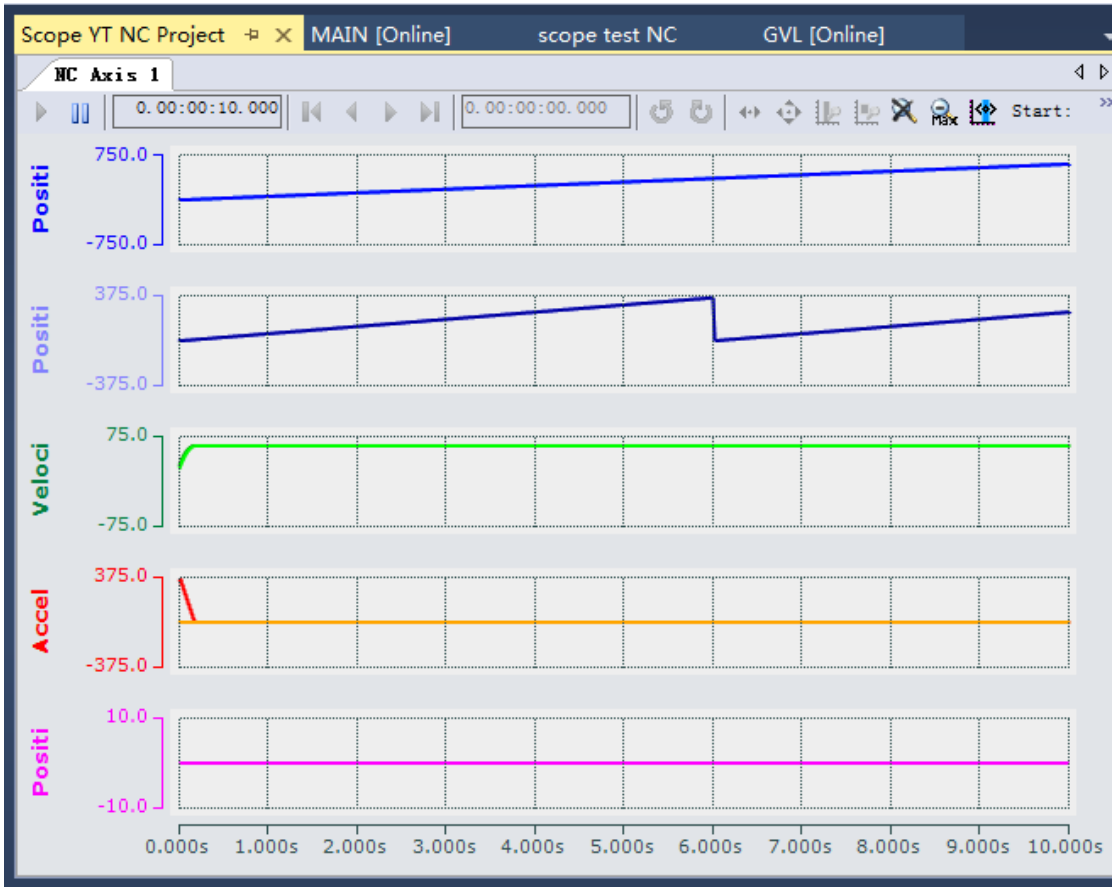
（5） 新建 YT NC Project。



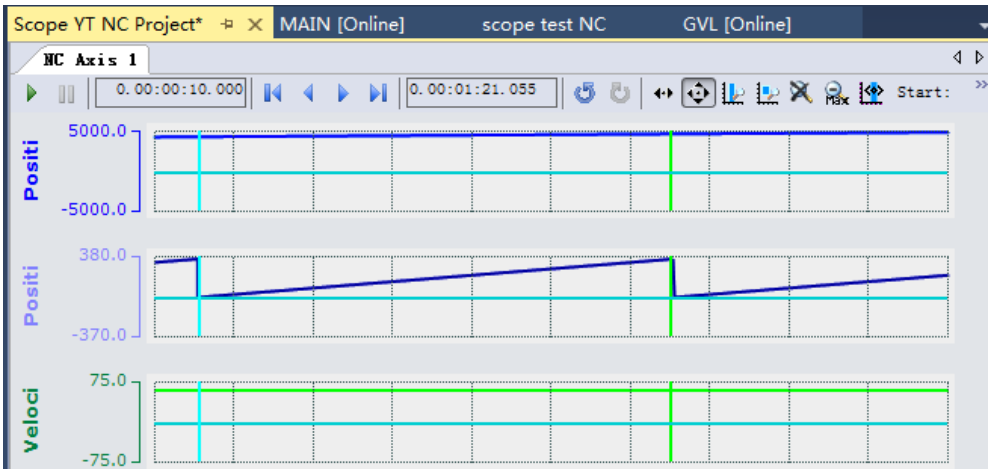
Twincat3 会根据 PLC 里 axis1 的各个参数自动生成各个 axis 轴，将参数以图像形式表现出来。



（6） 运行 PLC，并且手动打开电机使能和 jog 使能，点击 scope 的 record 按钮。可以在 scope project 的窗口中看到如图变化。

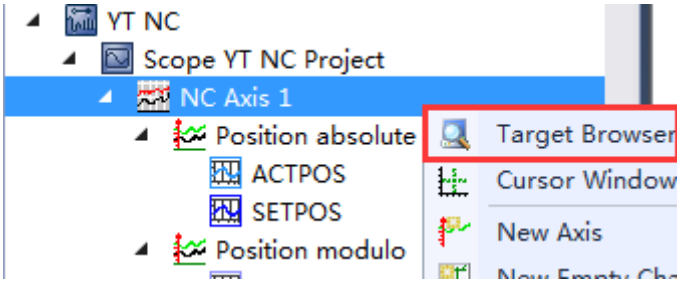


（7） 接下来就可以点击 stop display 暂停显示，通过 cursor 游标等方法对图表进行测量等工作。

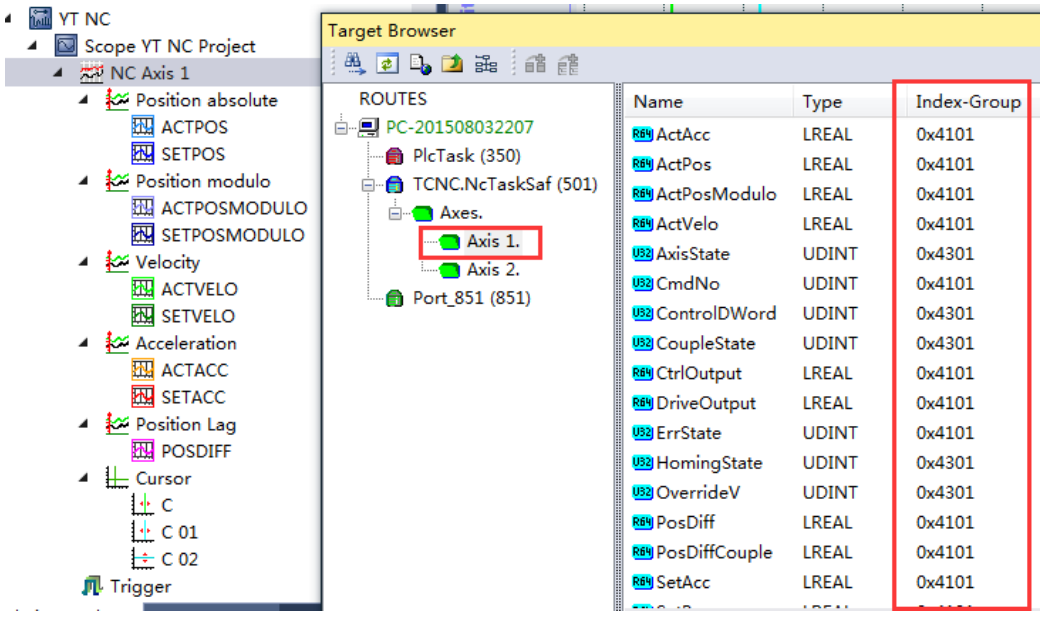


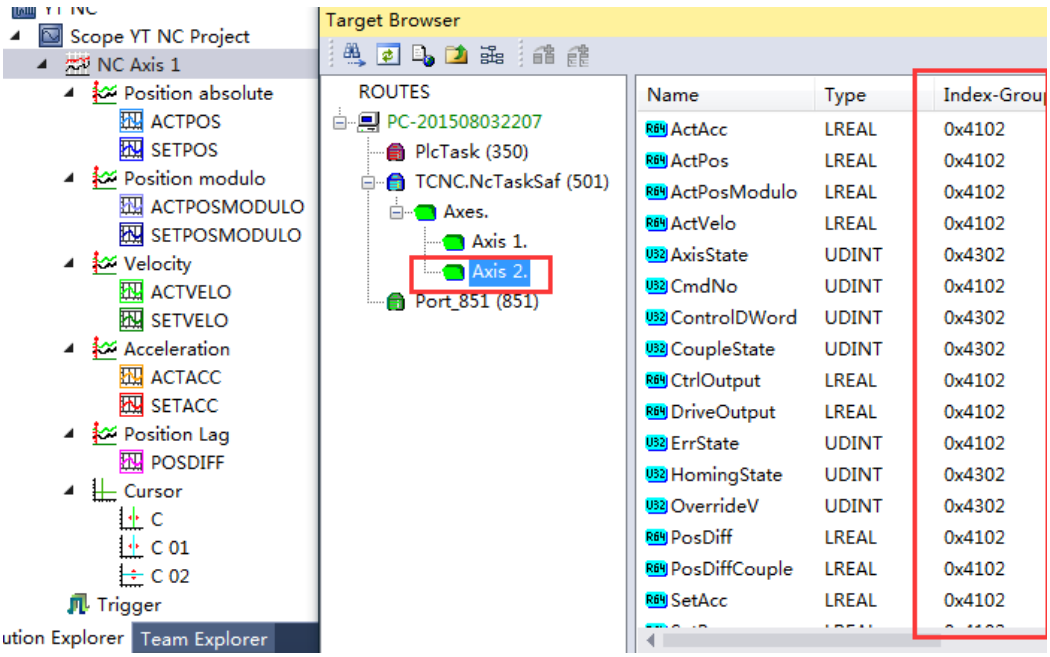
（8） 以上是一个 NC 轴的情况，如果在 NC 里再加一个 axis2 呢？实际情况下 NC 不仅仅只有一个 axis1 轴，但 TwinCat3 还是默认根据 axis1 来建立 scope view 窗口，于是我们需要做一些操作让新建的 scope project 对应到 NC 的 axis2 上。

- 进入 target browser。

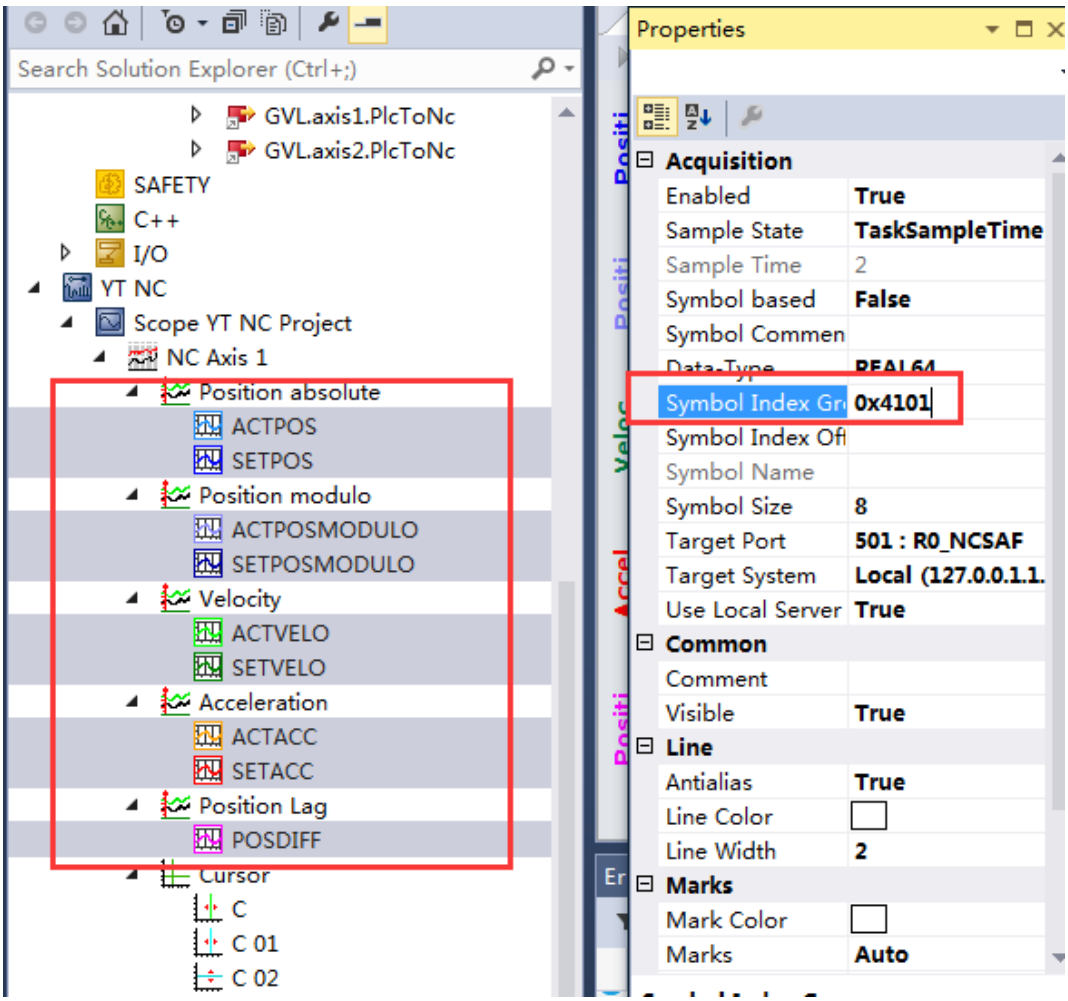


- 找到 axis1 和 axis2，对比发现 index-group 值不同，axis1 在 scope 里引用到的变量对应 0x4101，axis2 的对应 0x4102。于是我们可以统一更改变量属性来实现目的。

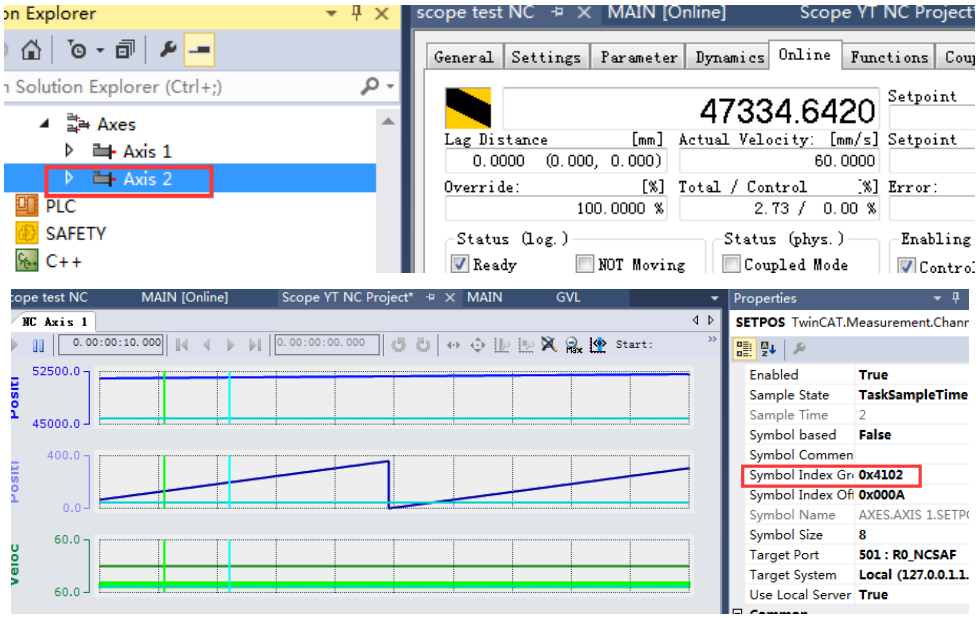




- 按住 Alt 键双击任意变量进行多选，properties 选项卡里就会显示统一的属性，将 symbol index group 的 0x4101 改为 0x4102 即可。



- 开始记录测试，NC 的 axis2 值变化，scope 显示 axis2 的变量值，更改成功。



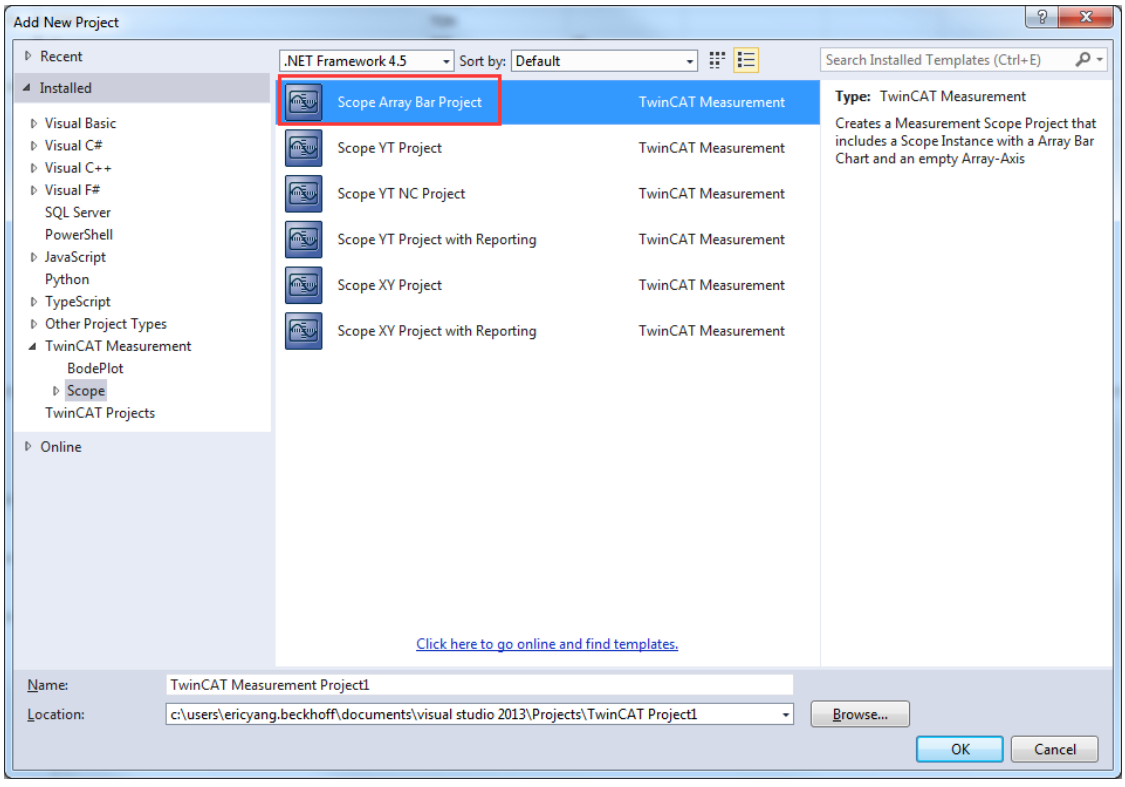
1.3 最后使用 Scope Array Bar Project

（1）首先在程序中写一段，产生一个随机数，随后依次赋值给一个数组中的元素中

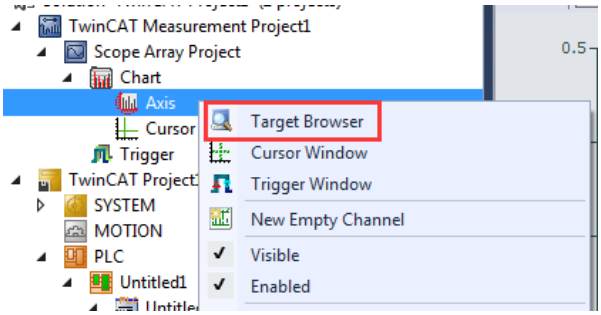
```
PROGRAM MAIN
VAR
    DRAND1:DRAND;
    TON1:TON;
    a: INT;
    real_data: LREAL;
    array10: ARRAY[0..9] OF LREAL;
    m: INT;
END_VAR
```

```
DRAND1(Seed:= a, Num=>real_data );

TON1(IN:=NOT ton1.q , PT:=T#10MS , Q=> , ET=> );
IF ton1.Q THEN
    array10[m]:=real_data;
    IF m<10 THEN
        m:=m+1;
    ELSE
        m:=1;
    END_IF
END IF
```

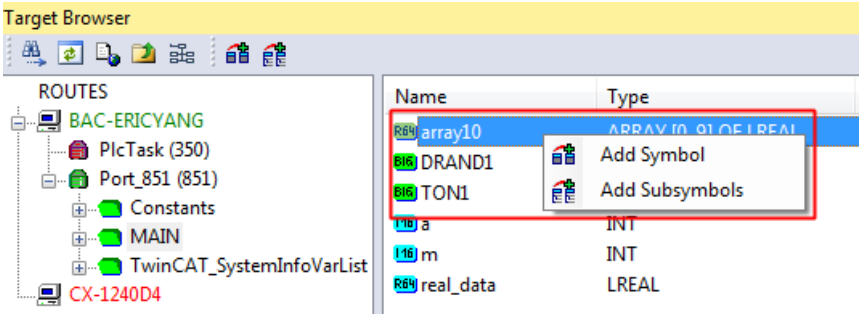
（2）程序运行起来后新建 scope 对这个数组进行观察，选择 Scope Array Bar Project 进行添加

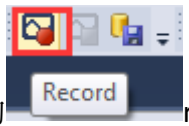


（3）右键 Axis 找到 Target Browser，开始查找变量

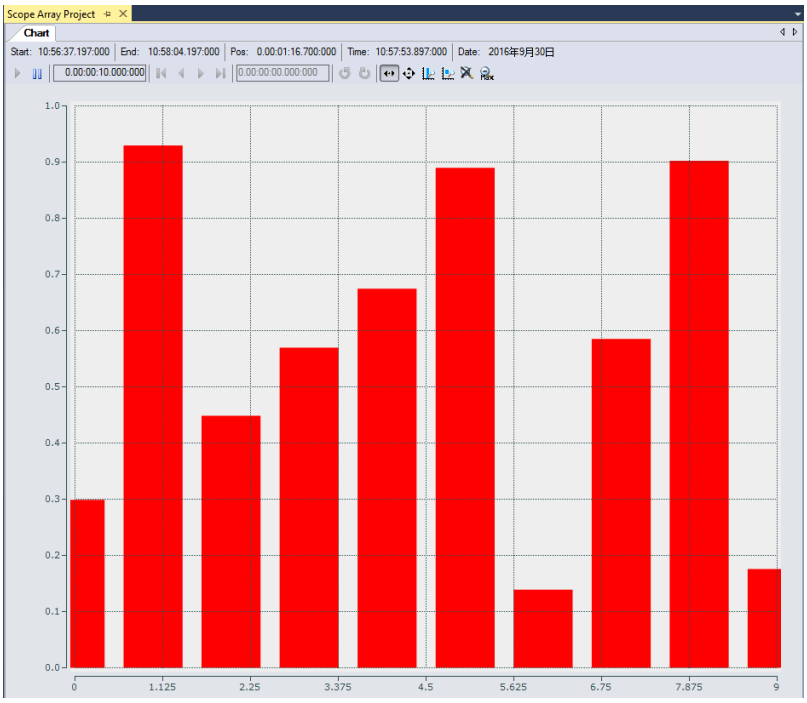


（4）在对于 runtime 中找到 array10 这一数组后，直接右键进行添加

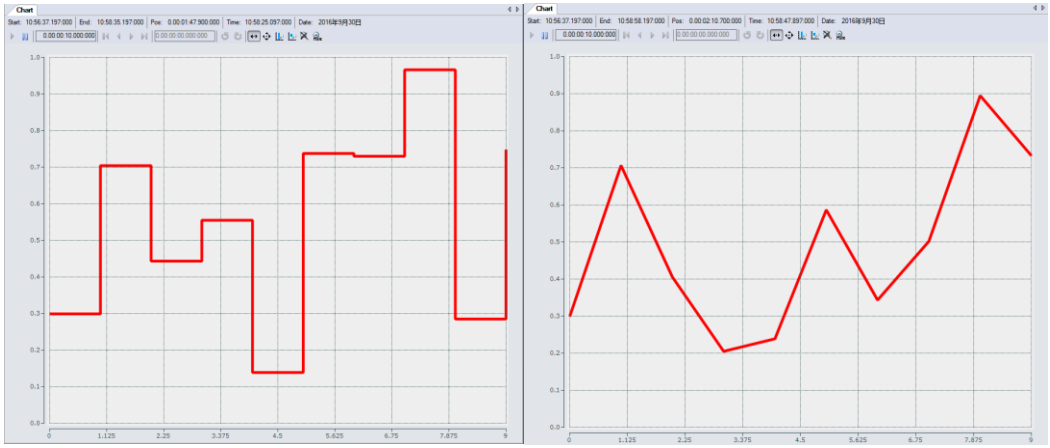


（5）点击工具栏中的 

record 开始采集数据，可以发现 10 个元素的数组产生的随机数实时采集并在 scope 中通过柱状图的方式显示



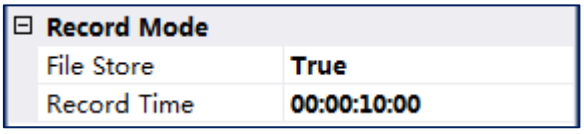
（6）当然除了柱状图的方式，还有阶梯图和折线图的模式可供选择



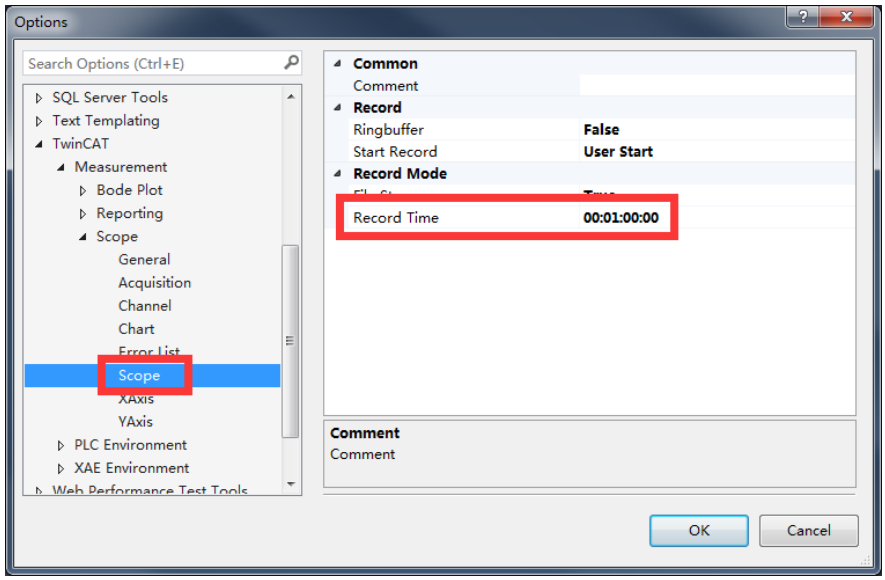
1.4 Scope View 记录时长、坐标轴 T/X 的轴长、显示时长的分别以及如何设置 Scope 的独立采样周期

- Scope 记录时长：Scope 连续记录的最长时长，在不购买 License 的情况下最长是一小时

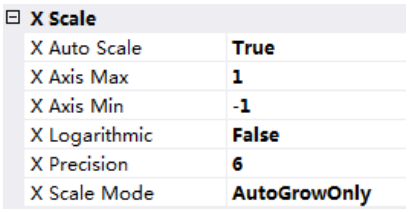
默认设置是 10 分钟。选中这个 Project，在 Properties 里选择 Record Mode→Record Time



也可以在菜单栏 Tools 下选择 Option 进行设置



- X 的轴长：单是坐标轴长，在 YT 轴里它和显示时间是一样的；但是在 XY 轴里，这就不是一样东西了，X 的变化范围可以很大，改显示时间可以只是取 XY 变化中的一小段。



X 轴的长短可以选中 Chart 下面的 Axis，在 Properties 的 X Scale 下可以修改（这部分属性仅在 XY 轴里才会出现，YT 轴没有的）。

X Auto Scale：X 轴长是否跟随 X 变量自动缩放。True 则自动缩放；False 不自动缩放，由 X Axis Max 和 X Axis Min 决定。

X Axis Max：X 轴最大值（X Auto Scale 为 False 才可修改）

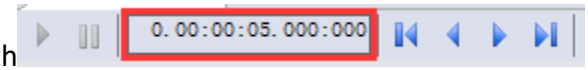
X Axis Min：X 轴最小值（X Auto Scale 为 False 才可修改）

X Logarithmic：对轴取对数显示

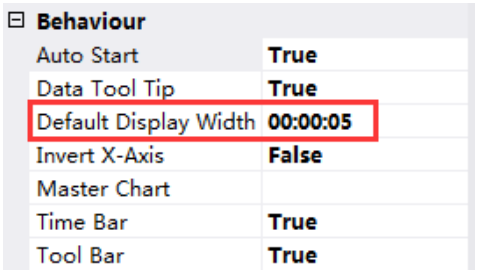
X Precision：

X Scale Mode：两种模式切换。AutoGrowOnly 是 X 轴随 X 变量变化，但是最终显示 X 所达到过的历史最大最小值；AutoGrowNShrink 则是 X 轴的实时显示，就是当前 X 达到的最大最小值，实际上就是对应到显示时间了。

- 显示时长：可以修改的地方有两处。

法一：直接在工具栏里改 Display Width 

法二：选中这个 Chart，在 Properties 里面修改 Behavior→Default Display Width，



- **Scope 独立采样周期的设置**

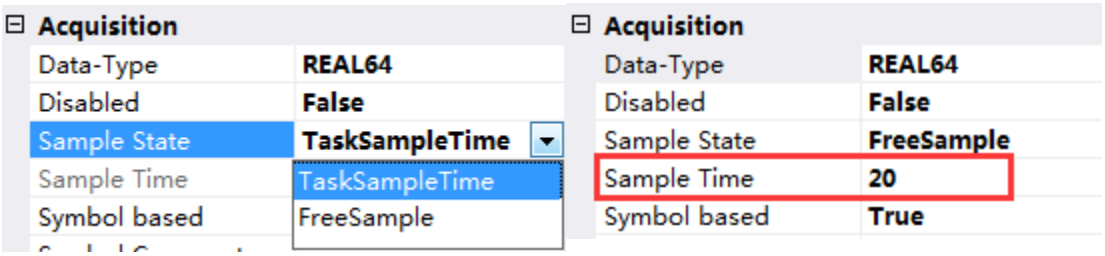
每一个独立的变量都可以设置单独的采样周期，但是这个值要大于 PLC 的周期。

选中轴下面一个单独的变量，在 Properties 里面选择 Sample State，有两个模式。

TaskSampleTime：Scope 的采样周期和 PLC 周期一样。

FreeSample：Scope 的采样周期可以定义为任意大于 PLC 周期的数。

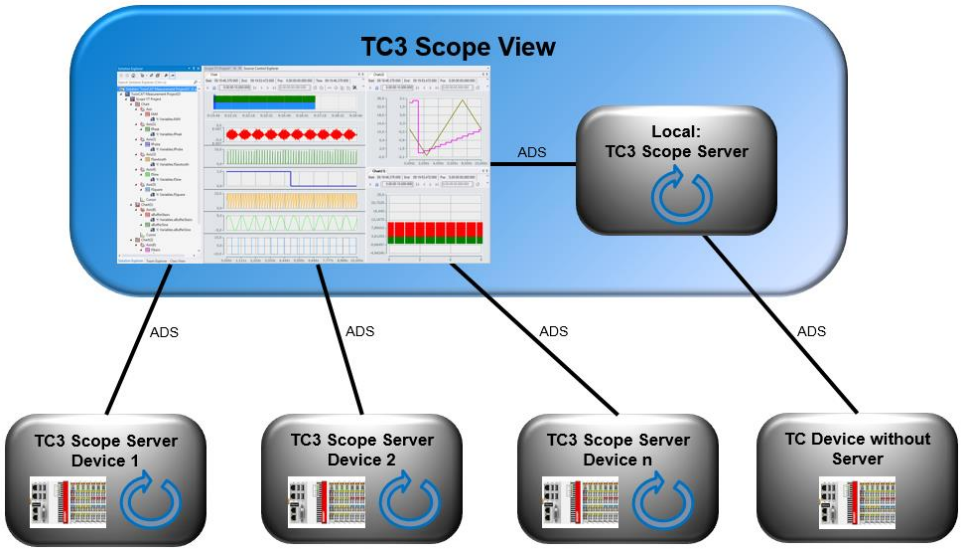
只有选择 FreeSample 模式的时候，Sample Time 才可更改。



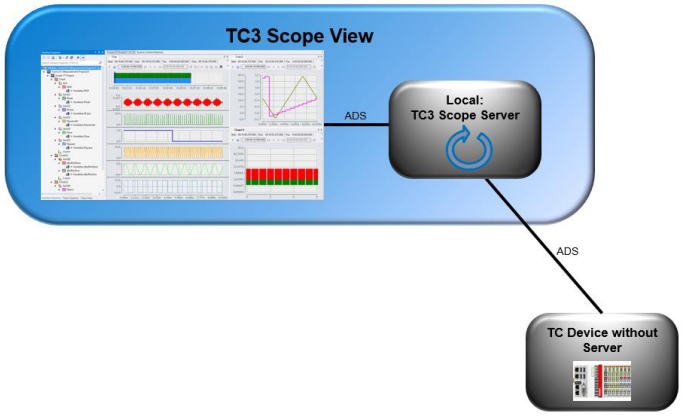
如果把 Scope 的采样周期设置的比 PLC 周期小也可以，程序并没有报错，这是为

什么？
TC3 里面这样设置是不会报错的，但是没有意义，比方说本身 PLC 的扫描周期就是 10ms 了，那么在这样一个周期内最终扫描出来的某个变量数值都是一样的，所以即便我们设置的 Scope 采样周期 5ms，还是无法在一个 PLC 周期里获取两次数值的。

1.5 TwinCAT 3-Scope View 运行机制
实际 Scope 分为三个等级，Scope Base、Scope Server 和 Scope View Professional，常用的 Scope Base 已经整合在 TwinCAT3 的安装包中，但底层数据的传递和通讯还是通过 ADS 通讯来进行的，我们的 TwinCAT3 PLC 将数据发送出去以后，Scope View 再通过 ADS 获取变量和变量值。
但是在实际应用中，经常会本地控制器带多个从站和远程 IO，这就涉及到远程数据采集的问题了。这里可以提供两种解决方案。

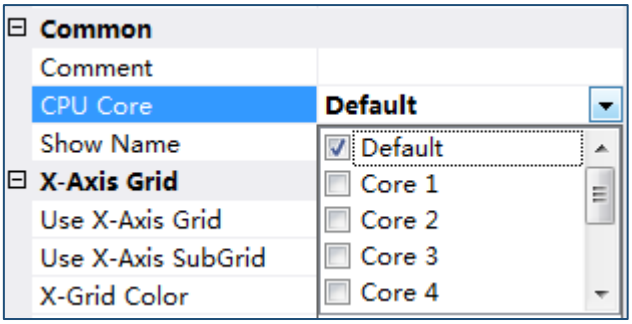


- 远程的 TC 或控制器不加装 TC Scope Server，这样所有的 Scope 采集工作都由 Local 的控制器来完成，也可以同时采集多个远程控制器的数据，通讯都通过 ADS 通讯方式来完成。但是这种情况下，势必会造成本地控制器的 CPU 占用率较高。

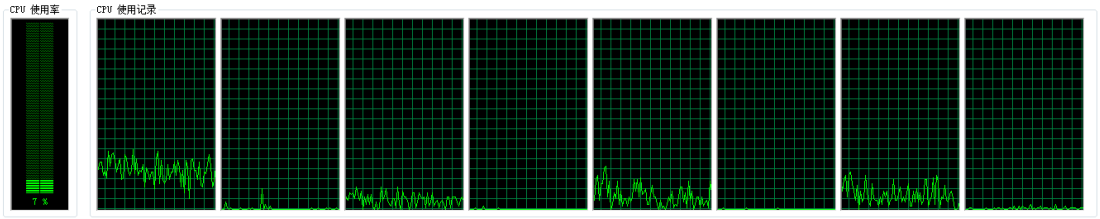


我们可以利用多核的优势，将这些任务分摊到其他核上去。

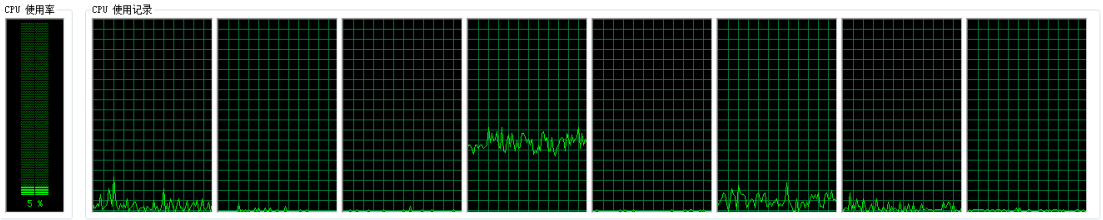
选中单独的一个 Chart，在 Properties 选项卡下修改 CPU Core。



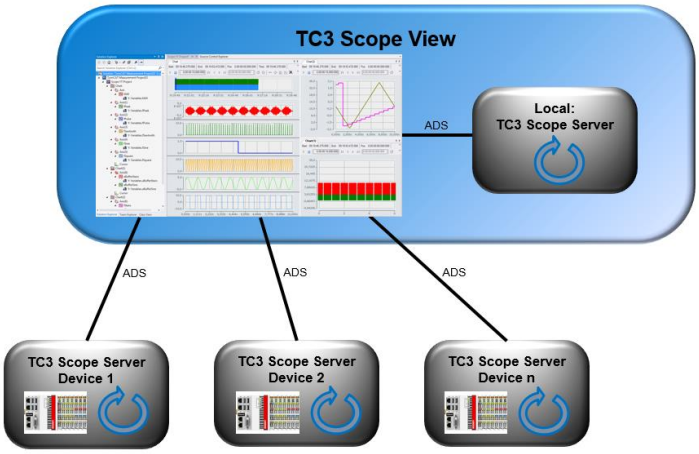
不分配的情况下的资源占用显示，主要分布在核 1、5、7 中，



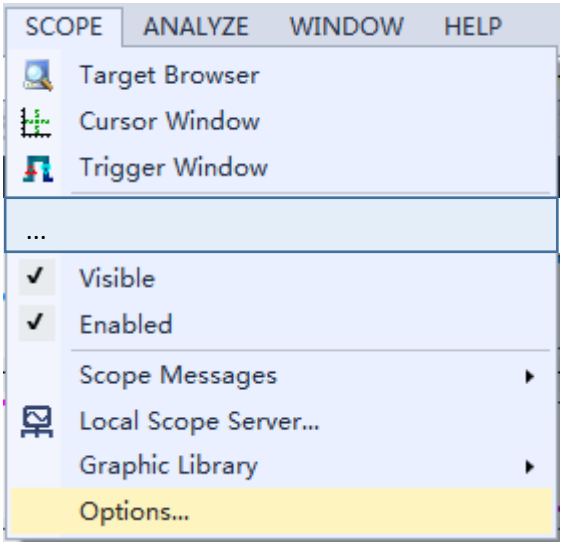
将 Chart1 和 Chart2 分配到 Core4 和 Core6 当中去后的资源占用显示。可以明显的看到核 4 和核 6 的 CPU 占用率提高了，而核 1 和核 5 的 CPU 占用率降低了。

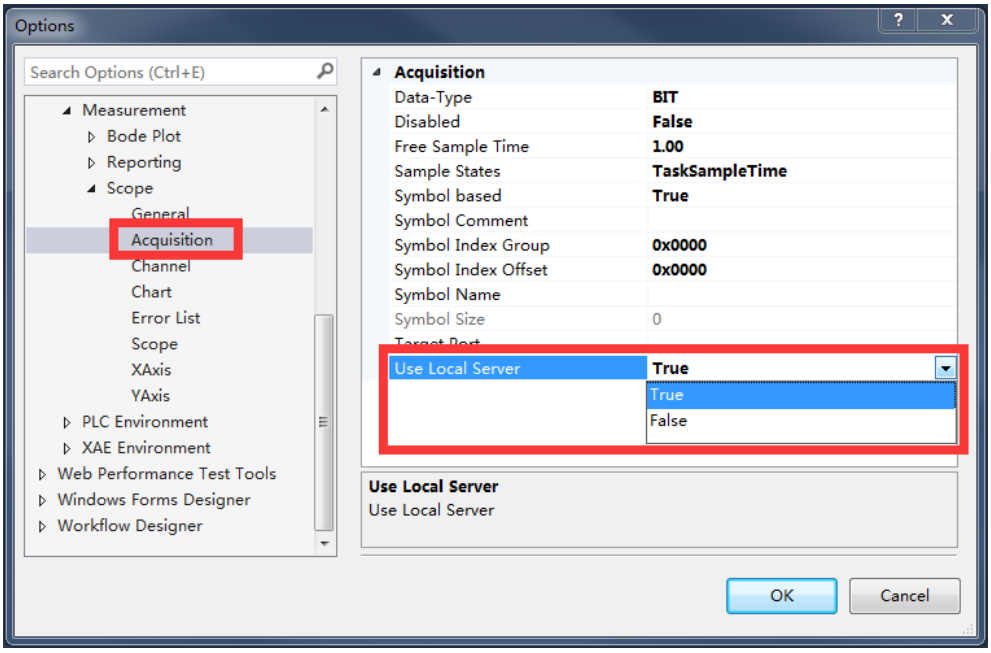


- 远程的 TC 或控制器都加装 TC Scope Server，这样所有的 Scope 采集工作都由各自的控制器来完成，通讯都通过 ADS 通讯方式来完成。这样可以让从站自行完成采集和绘制的过程。需要选择远程的 Server 来完成。



选择菜单栏 SCOPE→Options...，弹出对话框里选择 Scope→Acquisition→Use Local Server，如果选 True 就是用本地的 Server，选 False 就是用远程的 Server。



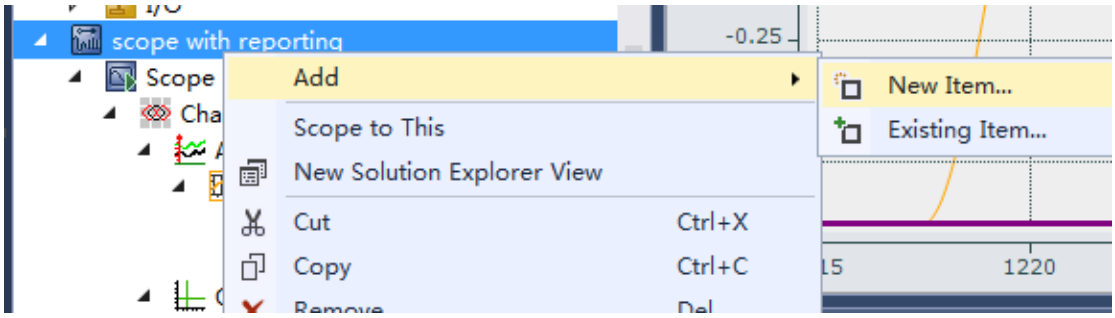


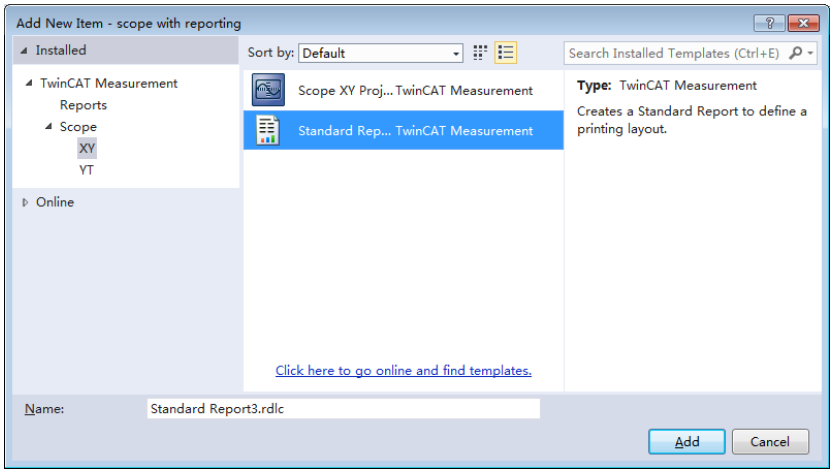
2. TwinCAT 3-Scope View 的 reporting 功能（TC3.1.4022）

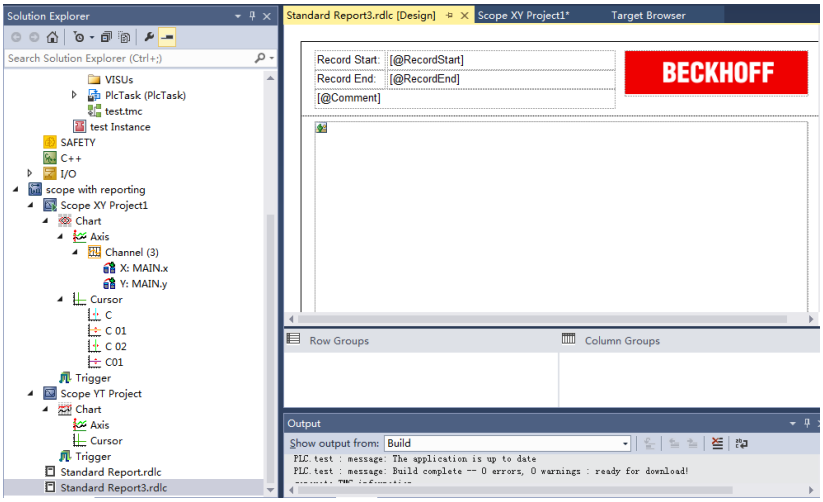
在新建工程项目的时候我们也看到有 Scope YT project with Reporting 和 Scope XY project with Reporting 选项，可以直接新建带 reporting 报告的 project。当然，我们也可以在已经建立好的工程项目中添加 reporting。

2.1 TwinCAT3-Scope View 的 Reporting 功能使用

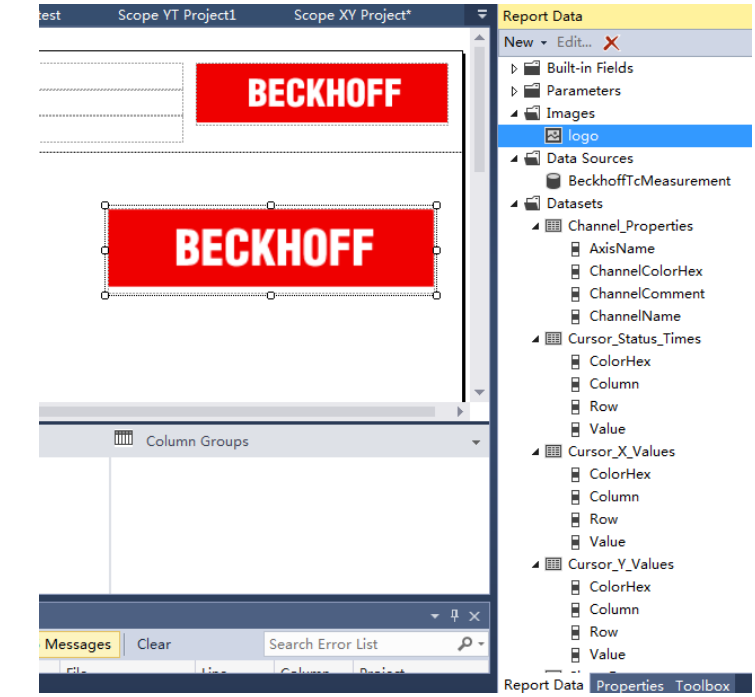
（1） 以刚才的 XY scope 程序为例，做一个 reporting。右键 scope project 项目，选择 add，点 new item，选择 standard report 新建标准的报告文件，如图所示。双击.rdlc 后缀文件即可看到 reporting 的模板框架。

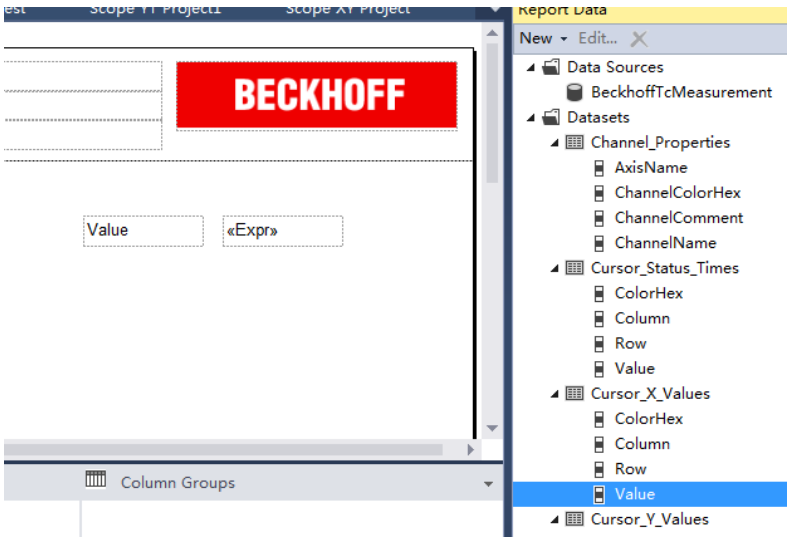




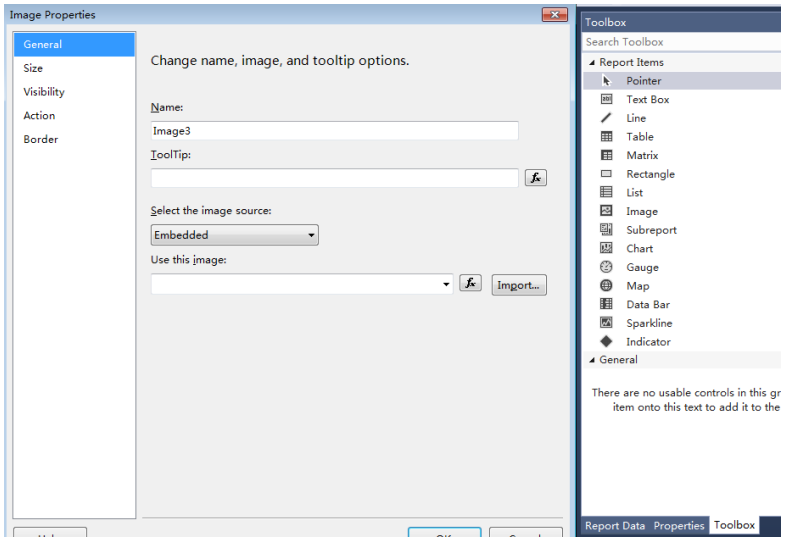


（2） 现在我们可以对这个模板进行修改，可以在 report data 选项卡看到可以添加进 reporting 的数据插件，类似于网页设计，可以加入 logo，也可以加入数据框、游标值、图表值等等。

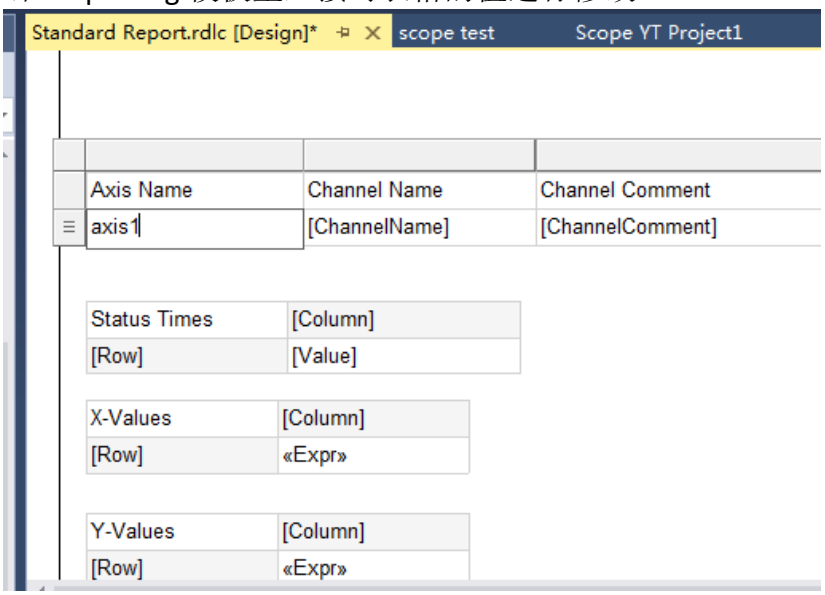




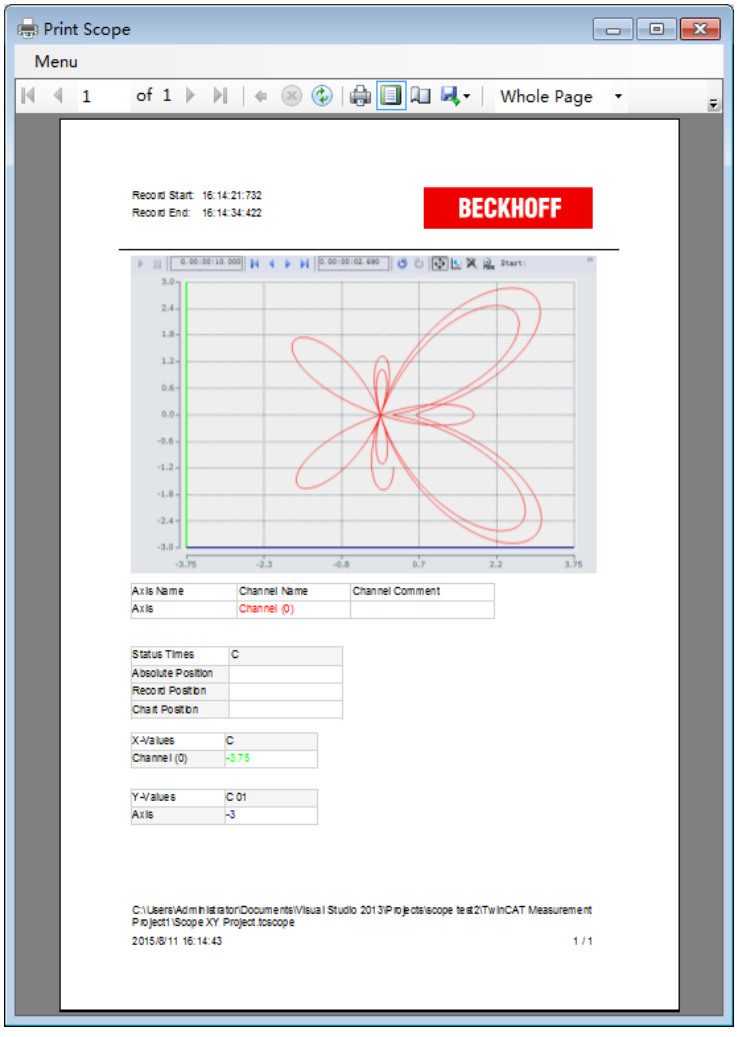
（3） 也可以用 toolbox 添加常规的插件，比如文本框或者图片。



（4） 可以在 reporting 模板里直接对表格的值进行修改。



（5） 点击 file，点击 print 打印，就可以看到 reporting 的预览了，或者按快捷键 ctrl+P。

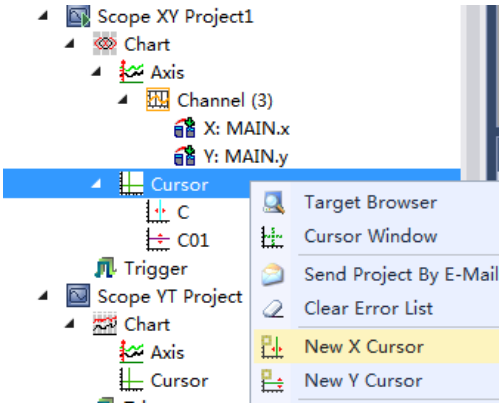


（6） 可能在上面第一点双击.rdlc 后缀的文件后显示的是 xml 源文件，原因是报

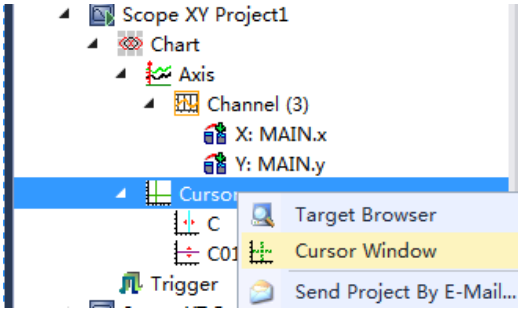
表编辑功能是需要 Report Designer(Visual Studio)支持的，可以补装一个 Microsoft SQL Server Data Tools 来完善此功能。

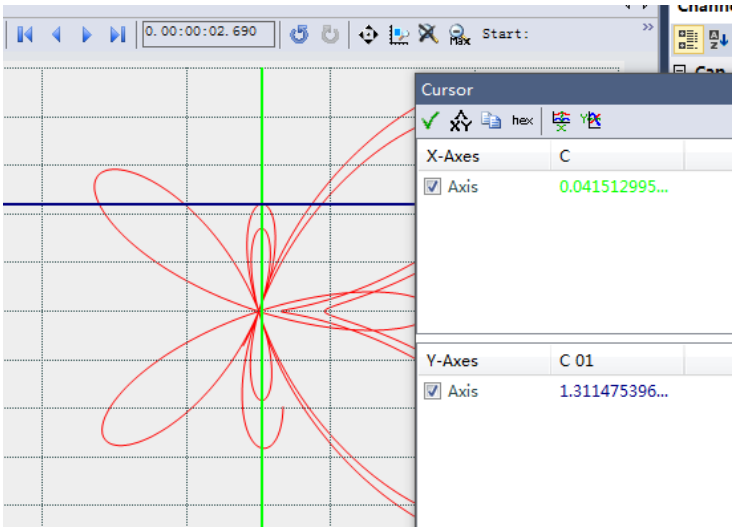
3. TwinCAT 3-scope view 的 cursors 功能

（1） 接下来我们使用 cursors 游标功能，以刚才的 XY scope 程序为例。右键 cursor，分别点击 new X cursor 和 new Y cursor，可以分别显示经过该线的 X 值和 Y 值，如图所示。



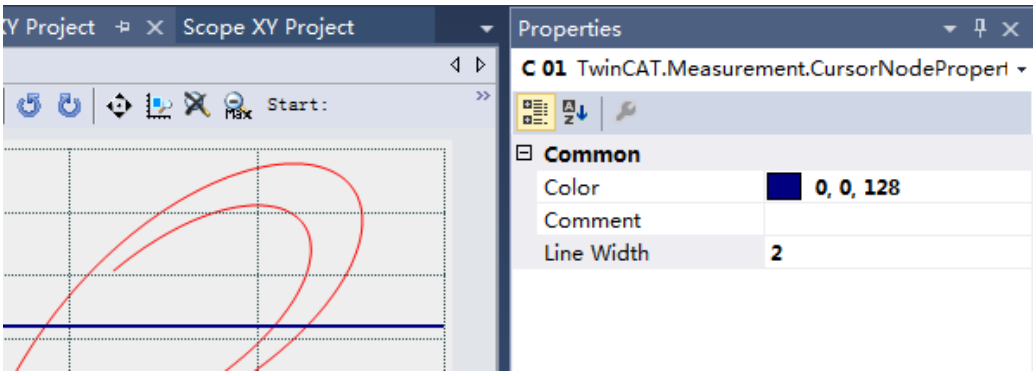
（2） 右键 cursor 点击 cursor window，可以看到游标值的窗口，显示图上游标处的 X 和 Y 的值，拉动 scope 图中的两个游标线，可以观察游标线与实线交点的 X 和 Y 值。每个游标线都有各自的颜色，和 cursor 窗口的值颜色对应。



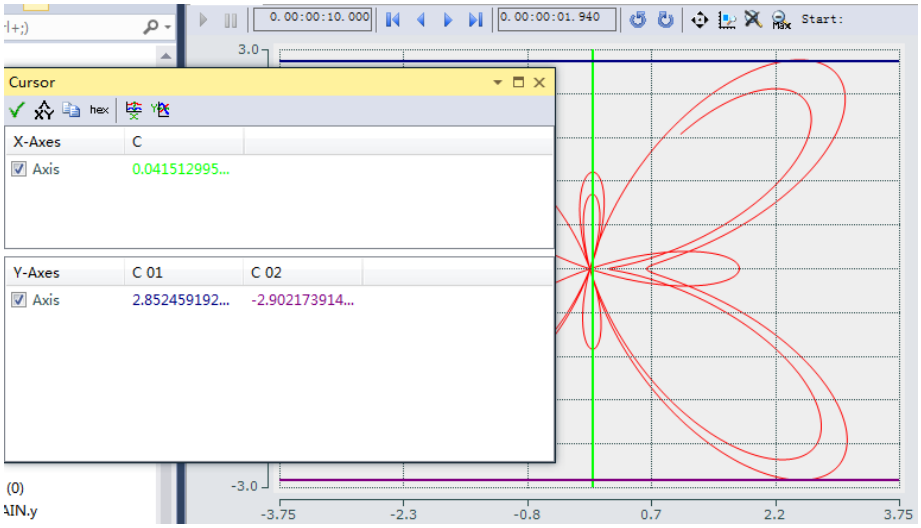


（3） 在每个 X、Y cursor 的 properties 里可以更改游标线的颜色和粗细，上图线

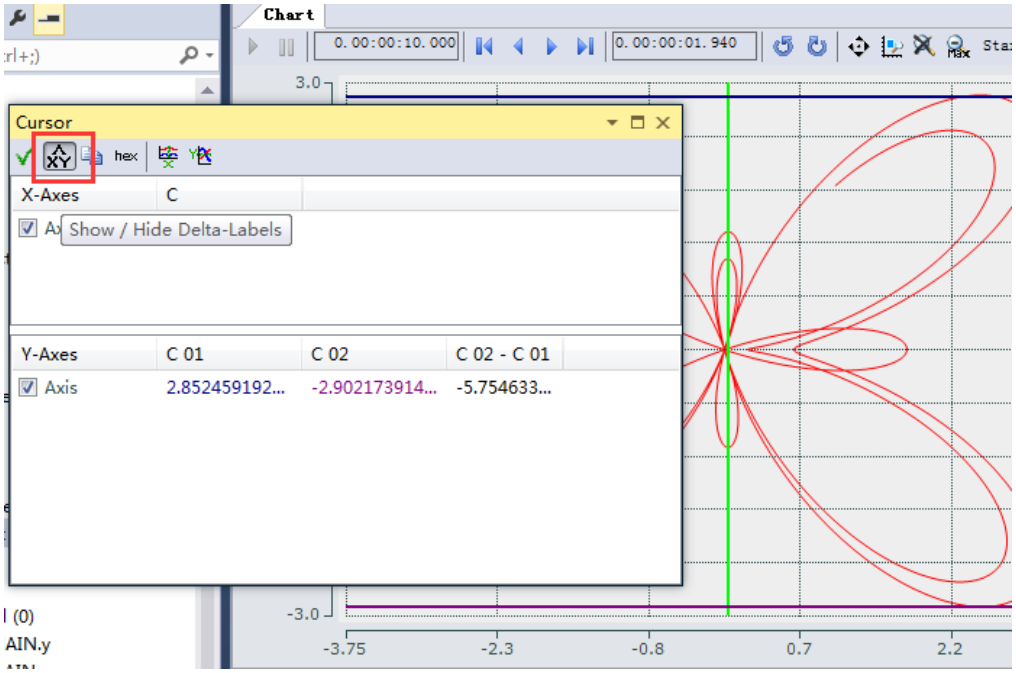
宽为 1，不方便观察，这里改为 3 。



（4） 可以再新建一个 Y cursor，拉动两个水平游标线至 Y 最大值处和最小值处，可以在 cursor 窗口看到两个游标处的值。



（5） 点击 cursor 窗口第二个按钮，将显示 X 或者 Y 两个游标的差值，比如刚才的最大值最小值之间的差值，就会自动计算出来显示在窗口上。

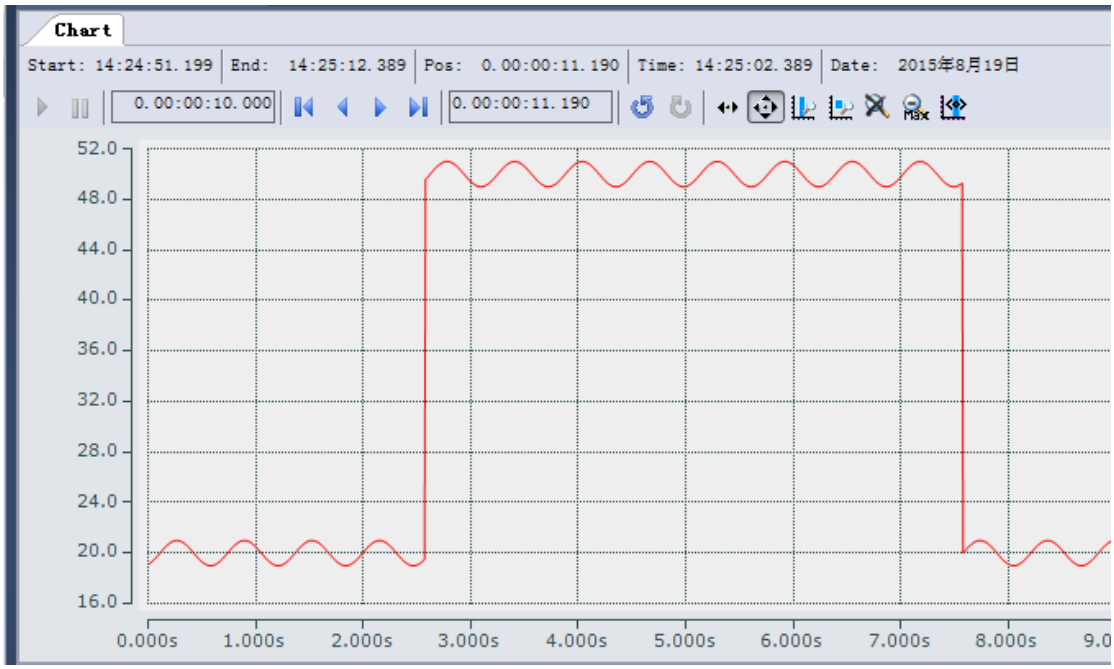


4. TwinCAT 3-scope view 的 trigger 功能

使用 Trigger 模块为 Scope 配置添加各种触发器功能。我们以一个 YT project 为例，PLC 的程序如图所示。把 y 变量添加到新建的 YT project。

```
PROGRAM MAIN
VAR
    x: LREAL;
    y: LREAL;
END_VAR
```

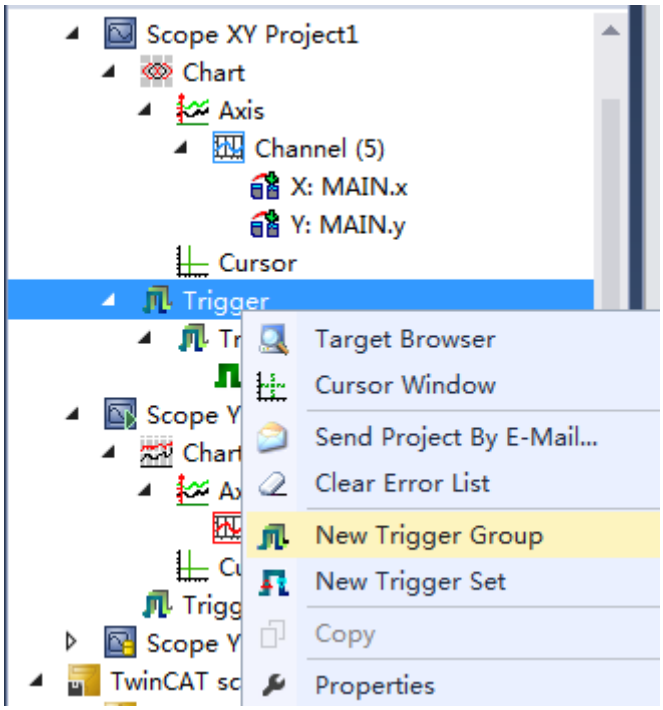
```
IF x<0 THEN
    x:=0;
    y:=SIN(x)+20;
    ELSIF x<100 THEN
    y:=SIN(x)+20;
    ELSIF x<150 THEN
    y:=SIN(x)+50;
    ELSE
    x:=0;
    y:=SIN(x)+20;
END_IF
x:=x+0.1;
```



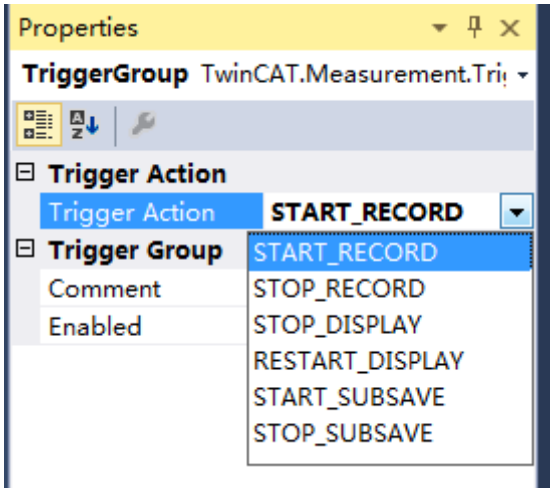
以 y 为 Y 轴变量，目的是使 y 值先在 20 附近震荡，过了一段周期上升到 50 附近震荡变化，持续一段时间再恢复。以该程序模拟监测水温的实例，y 为监测到的水温，20 为正常，超过 40 异常，我们需要把超过 40 的部分记录下来分析

异常原因，那么就可以用触发器自动控制从异常点开始记录，到退出异常点停止记录，或者把这段异常的部分显示出来，或者自动控制把这一段的记录存储下来。

在 scope project 中右键 trigger，选择 new trigger Set，新建一个触发器设置，会自动生成在一个新的 trigger group 下。



右键 trigger group，选择 properties 配置，可以在属性选项卡里看到触发器的配置信息。



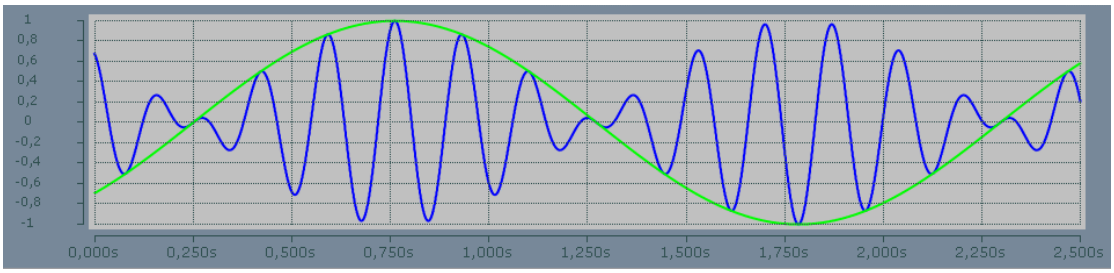
Trigger action: 触发器触发的行为区，在下拉选项里可以看到，定义了当整个触发器组满足条件后执行那些动作：

StartRecord 开始记录:要使用这个选项应该将 Scope 设定配置成触发器启动“trigger start”。一旦记录按钮被点击，Scope 将连接至相关的 Sever 并检查触发器条件而不马上开始记录。

StopRecord 停止记录: 这个选项里的设置要在开始记录的情况下才会生效。使用前置/后置触发器（Pre- and/or Post-Trigger）来定义触发事件前后的时间范围。

StopDisplay 停止显示: 这个动作停止所有运行模式下已链接的图表。使用触发器定位以显示宽度的百分比来改变触发器事件在图表中的位置。（举例如下：

设置 10 即绿色曲线在 2.5 秒的 10%=0.25 秒处其上升沿穿越零点）如果触发器条件被再次触发，显示将调至新触发事件处，以避免使用图表工具条中的中断按钮再次触发。



RestartDisplay 重启显示:这个触发器重启所有由“停止显示触发器”-"Stop-Display"-Trigger 暂停的所有图表。

StartSubsave 开始后台保存: 一旦这个触发器事件在后台发生，使用活动配置。后台保存始终在环形缓冲模式中运行（即使在基本配置中没有设定）。可以同时运行最多 5 个后台保存（由一个或多个触发器事件触发）。后台保存的记录长度可以在记录时间框“'Record-Time”中编辑。可以在基本配置中选择比使用的更长的时间（在这个情况下基本配置也应设定成环形缓冲模式‘Ringbuffer-Mode’）。

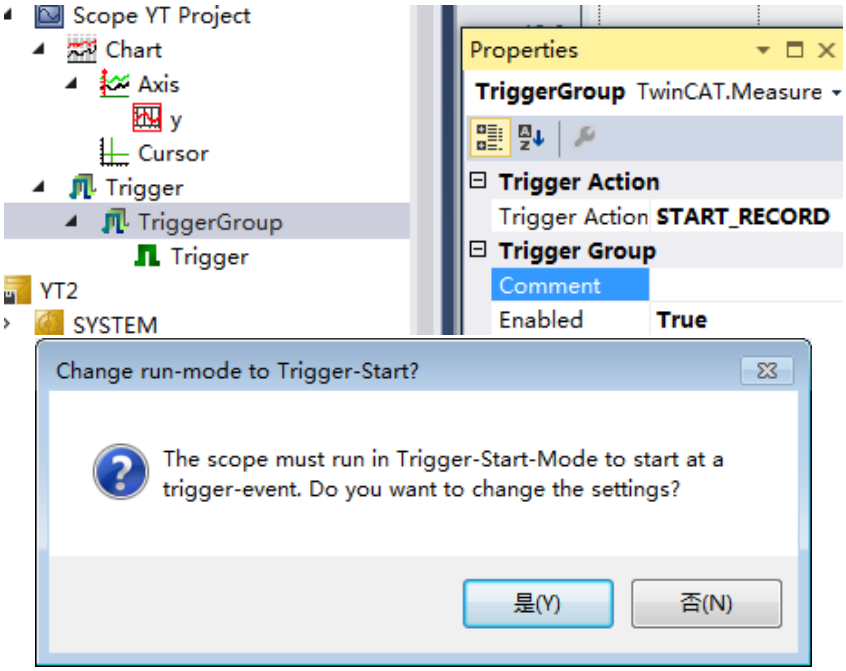
StopSubsave 停止后台保存: 这个事件触发将停止之前的存储，将之前存储的数据保存在一个.svd 文件中。存储路径可以更改为任意有效位置。存储文件名用基本配置中 Scope 的名称及一串数据时间编码建立。

Comment: 此处可以写该触发器的注释内容。

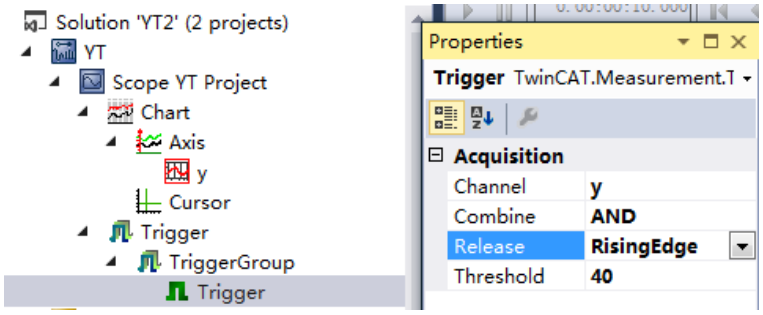
Enabled: 触发器的使能，默认为 true，可以双击改为 false 手动关闭触发器。

4.1 触发器控制记录开始和停止

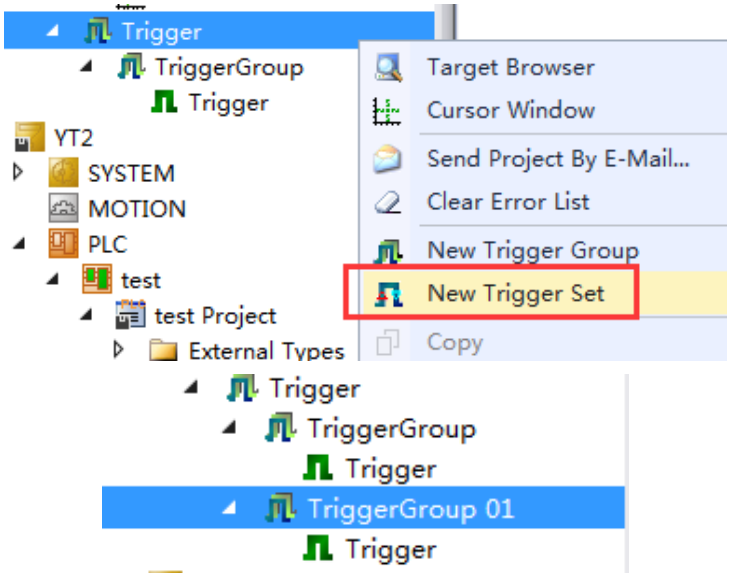
（1） 在刚才的 trigger group 的属性选项卡里，选择 START RECORD，跳出提示框，提示 scope view 需要在 trigger 引导模式下才能启动触发器事件，点击是确定，然后在 enabled 使能一栏选择 true。



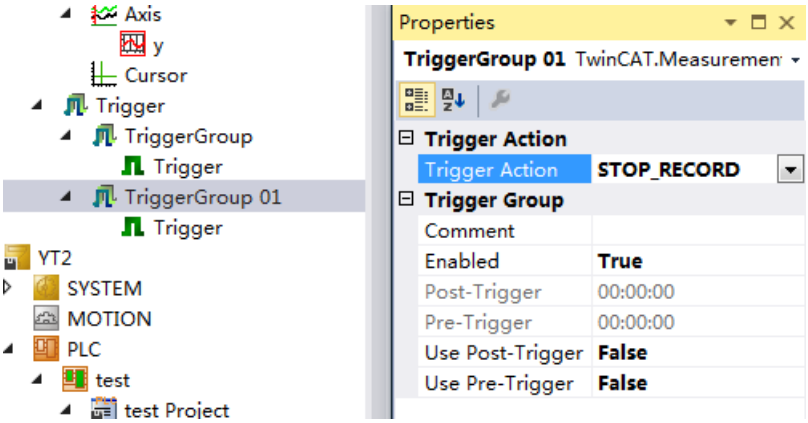
（2） 点击 group 下的 trigger，在 trigger 的属性选项卡里进行设置，channel 通道选择 y 变量，combine 逻辑选择 and 默认即可，这个选项在一个 group 下多个 trigger set 的时候生效，release 选择 risingedge 上升沿触发，threshold 值输入 40。这样，trigger 触发开始记录的功能部分设置完成，trigger 会在 y 值上升经过 40 这个值的时候触发 START RECORD，开始记录接下来的 y 值变化。



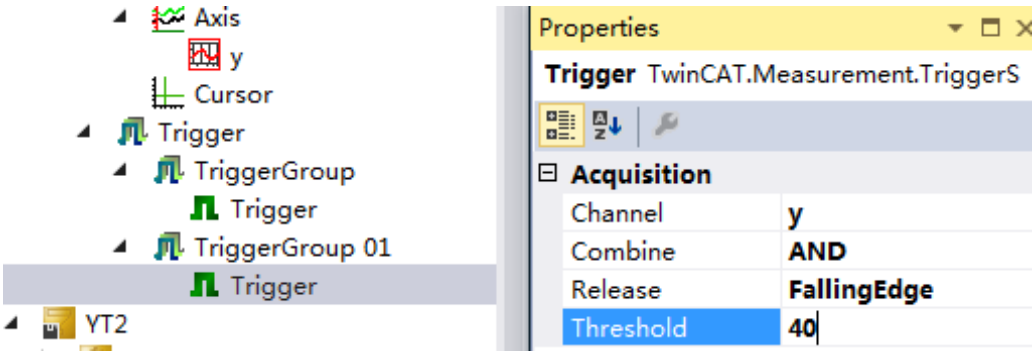
（3） 右键第一个 triggergroup 上方的 trigger，点击 new trigger set，新建一个 trigger 设置，并且自动生成在一个新的 triggergroup01 组下，接下来将这个 trigger 设置为条件触发停止记录。



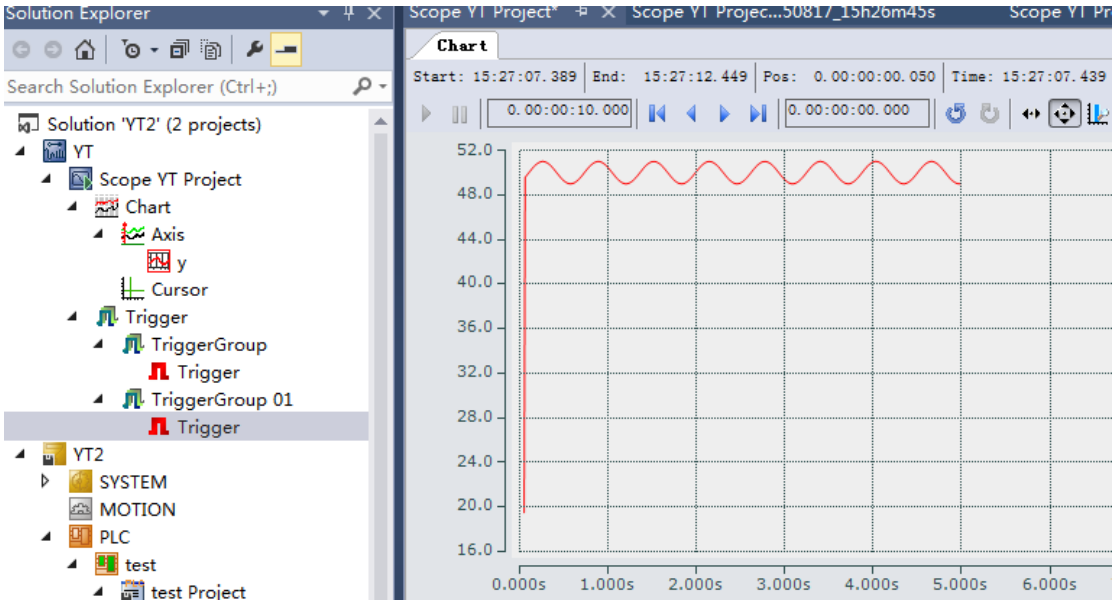
（4） 在 TriggerGroup01 选项卡里，action 选择 STOP_RECORD，触发行为停止记录，这时候会跳出提示框，提示 scope view 需要运行在环形缓冲模式下才能触发停止，点击是确认，然后 enabled 使能设为 true。



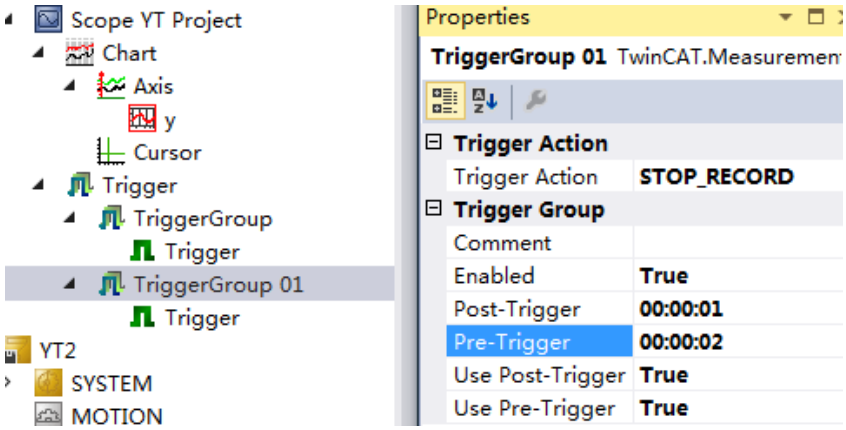
（5） 点击 triggergroup01 下的 trigger，在 trigger 的属性选项卡里进行设置，channel 通道选择 y 变量，combine 逻辑选择 and 默认即可， release 选择 fallingedge 下降沿触发，threshold 值输入 40。这样，trigger 触发停止记录的功能部分设置完成，trigger 会在 y 值下降经过 40 这个值的时候触发 STOP_RECORD，停止记录 y 值变化。



（6） 这样 trigger 设置完成了，接下来在 PLC 运行的时候，点击工具栏的 start record 按钮，然后观察效果。可以看到，如果触发器成功被触发，触发器的图标绿色  会变成红色  。数据记录是从 y 值跃升超过 40 的点开始，第一个触发器变红，到下降过 40 结束，第二个触发器变红。这样异常数据段的数据就自动记录下来了。



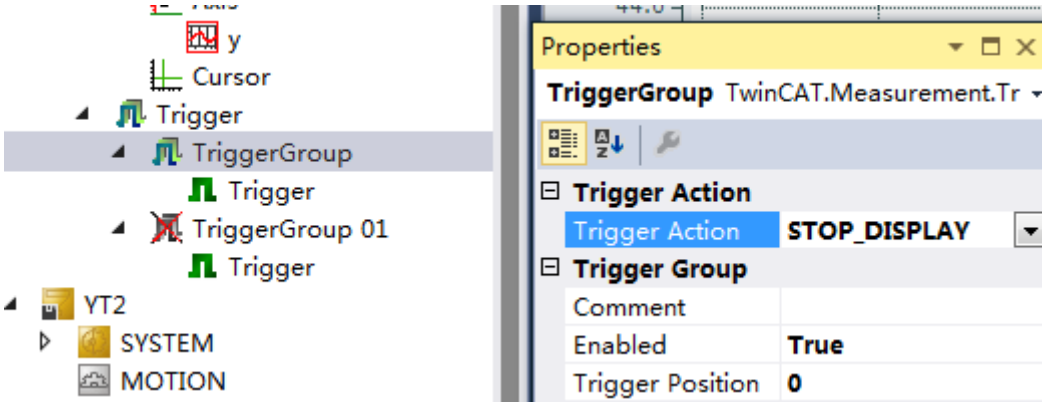
（7） 刚才在 TriggerGroup01 的属性选项卡里可以看到，有 post-trigger 和 pre-trigger 两个选项，定义了触发事件前后的时间范围。先将 Use post-trigger 和 Use pre-trigger 置 true，然后就可以设置时间。可以如图设置，在这里，post-trigger 1 秒指的是条件满足后 1 秒触发停止记录，pre-trigger 2 秒指的是条件满足触发之前要有 2 秒的正常记录，如果开始记录只有不到 2 秒就满足条件，触发不成功。



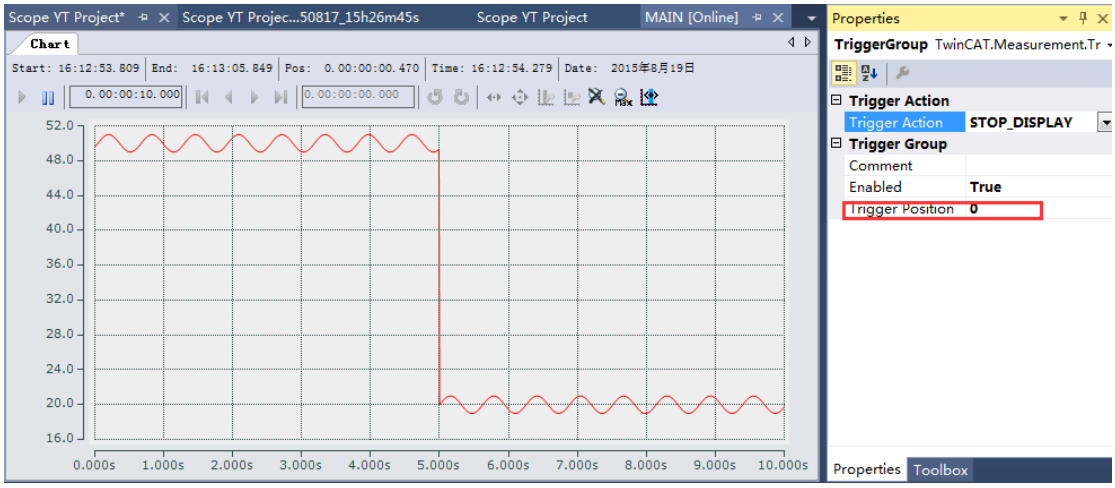
（8） 点击 start record 按钮开始记录，观察图表。如图效果。在下降沿过 40 满足条件后 1 秒才停止记录。

4.2 触发器控制显示停止和重新显示

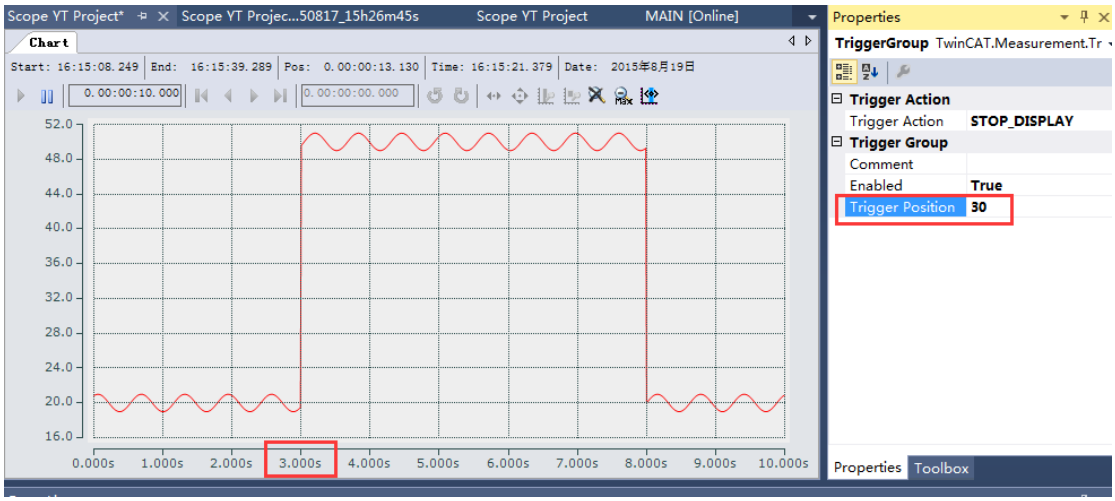
（1） 将 TriggerGroup01 使能置 false，TriggerGroup 的 action 改为 STOP_DISPLAY，trigger position 指的是图表停止时触发点在图表位子的百分比，比如这里先设置为 0，即 0%。trigger set 同上。



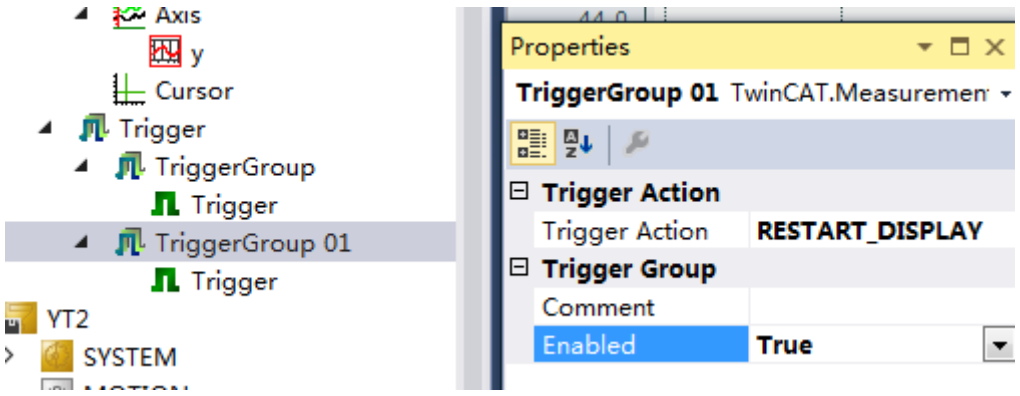
（2） 在 PLC 运行的时候点击工具栏 start record 开始记录，观察效果。图表显示自动立即开始，这时候的触发效果是，停止显示时即 y 值刚刚上升跃过 40 值的触发点会停留在图表 0%处，如图所示。



（3） 将 trigger position 改为 30，即 30%处，再记录一次数据，观察效果。如图所示，图表会在触发点到达图表 30%处停止显示。

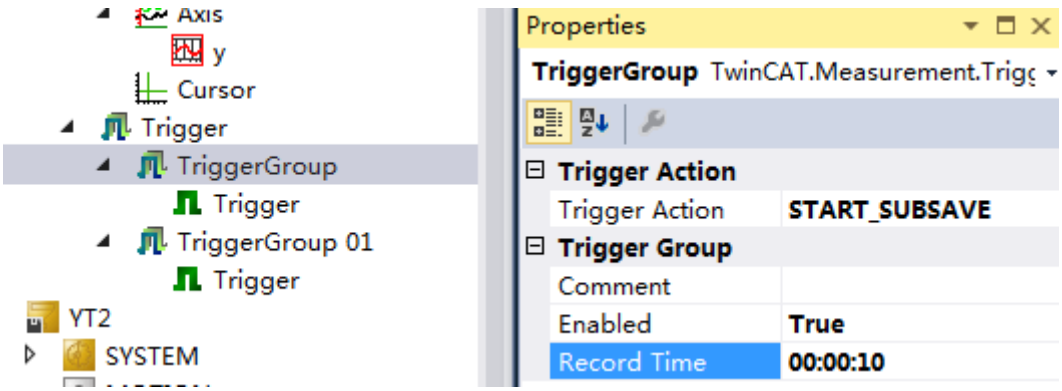


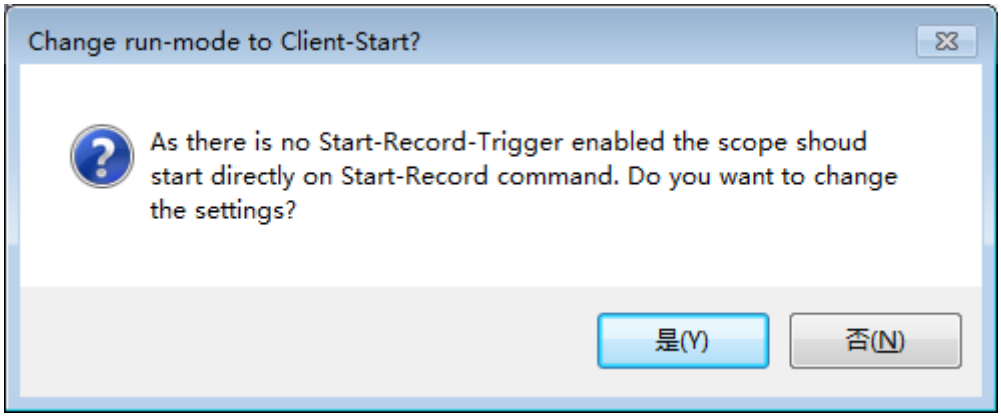
（4） 接下来将 triggergroup01 使能置 true，trigger action 改为 RESTART_DISPLAY，trigger 的设置同上不变，下降沿过 40 触发。那么这时候的效果是，y 值上升，触发显示停止，异常值记录大约在图表中间处停止，y 值下降沿过 40 恢复正常后重新开始显示。如图所示。



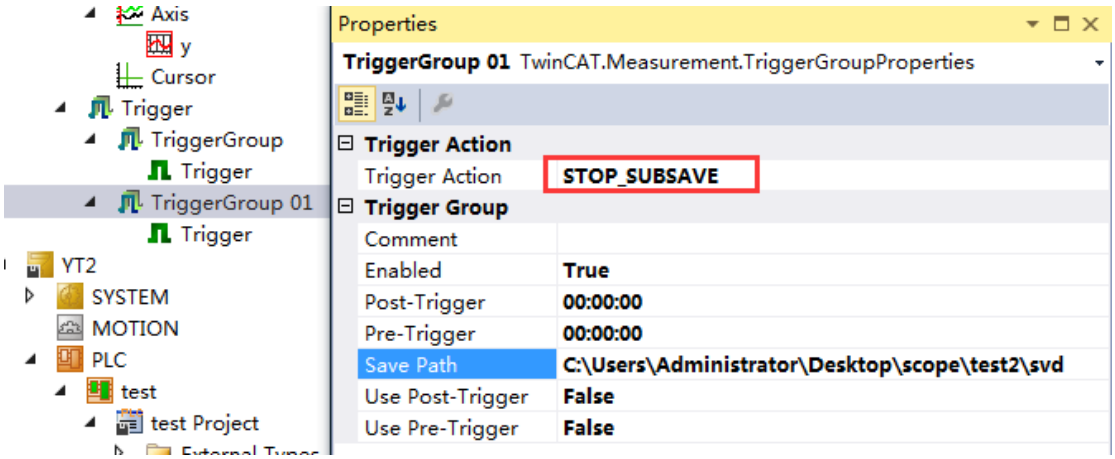
4.3 触发器控制自动存储数据

（1） 点击 triggergroup，在属性选项卡中将 action 改为 START_SUBSAVE，跳出提示框，点击是确定，设置 enabled 使能为 true，record time 指的是开始存储数据的时间长度，这里设置为 5 秒。



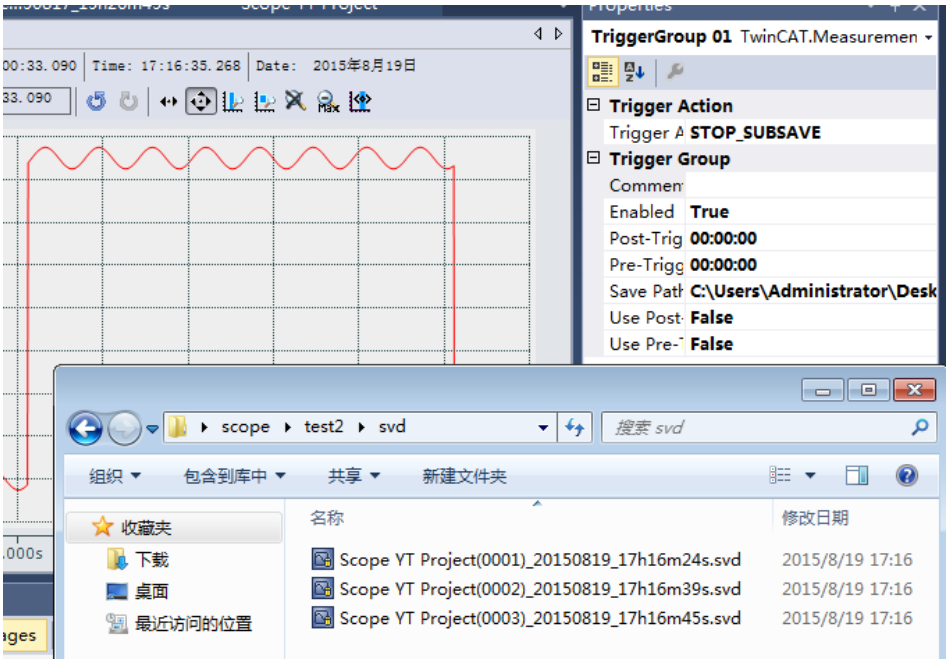


（2） 点击 triggergroup01，在其属性选项卡中设置 trigger action 为 STOP_SUBSAVE，触发将停止之前的存储，将之前存储的数据保存在一个.svd 文件中。存储路径可以更改为任意有效位置，在 save path 一栏输入地址即可。存储文件名用基本配置中 Scope 的名称及一串数据时间编码建立。Enabled 使能置 true。Post-trigger 和 pre-trigger 功能效果同上，这里不设置，置 false。

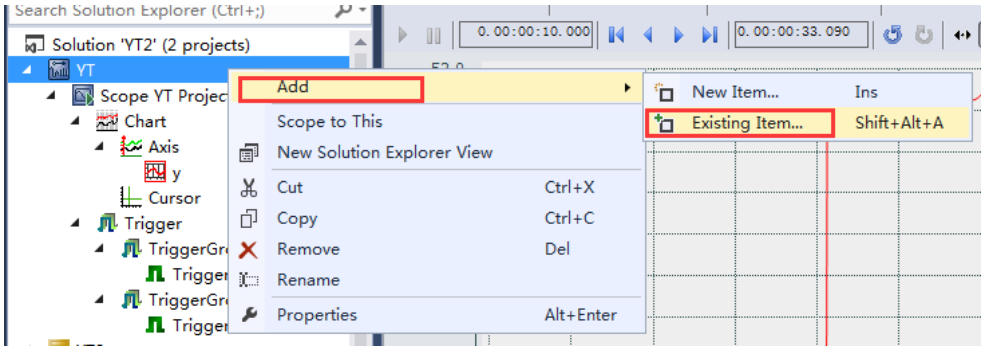


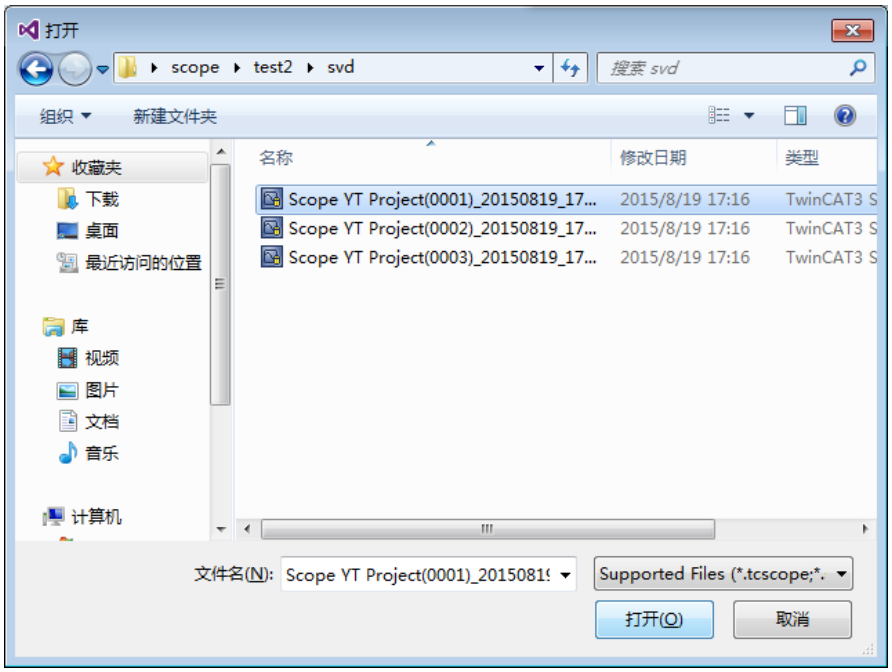
（3） 点击 start record 开始记录，观察效果。如图，上升沿过 40 触发数据自动存储，下降沿过 40 触发存储的数据保存为一个.svd 格式文件，方便调用分析，可以看到，在存储路径下，每次触发都会自动生成一个.svd 格式文

件。这里情形类似于监测水温异常，将异常段数据自动存储下来。



（4） 在树形目录里右键 scope 项目 YT，点击 add，点击 existing item，添加外部文件，添加刚才存储的.svd 文件进行查看。



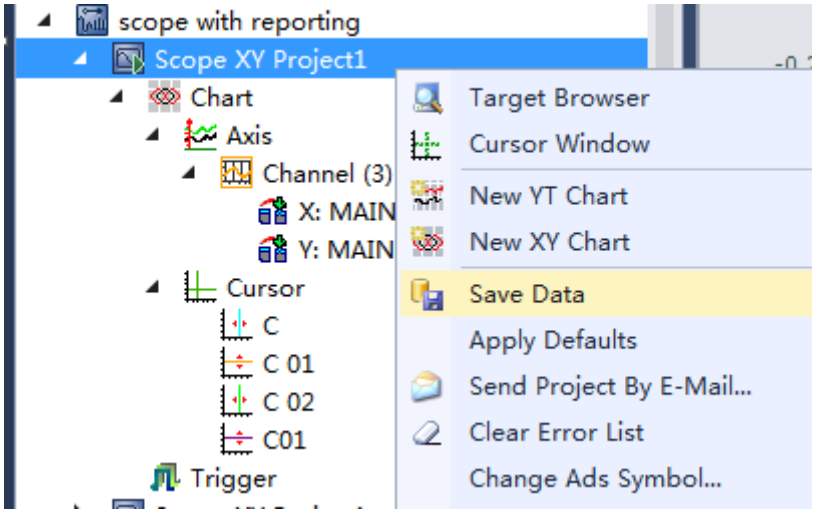


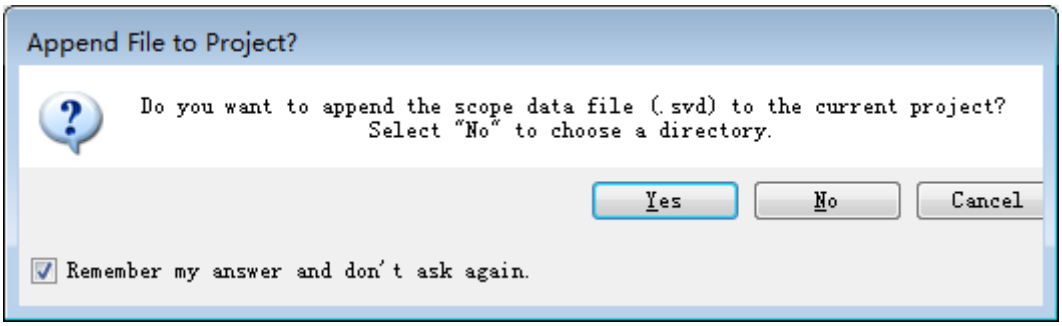
（5） 添加完成可以在树形目录下看到.svd 文件，点击就能看到记录下来的图表数据。如图所示，数据从上一次开始存储持续 10 秒长度。之后，我们就可以对这些图表进行编辑或者分析操作了。

5. TwinCAT 3-Scope View 的 Saving and Exporting data (TC3.1.4022 版本)

TwinCat3 可以将 scope 记录下的数据以 .svd 的格式保存下来，通过这个数据文件我们可以用 TwinCAT-3 Scope View 再次查看记录。

（6） 在 scope 工程右键选择 Save Data，在弹出的对话框中选择 yes 即可保存.svd。

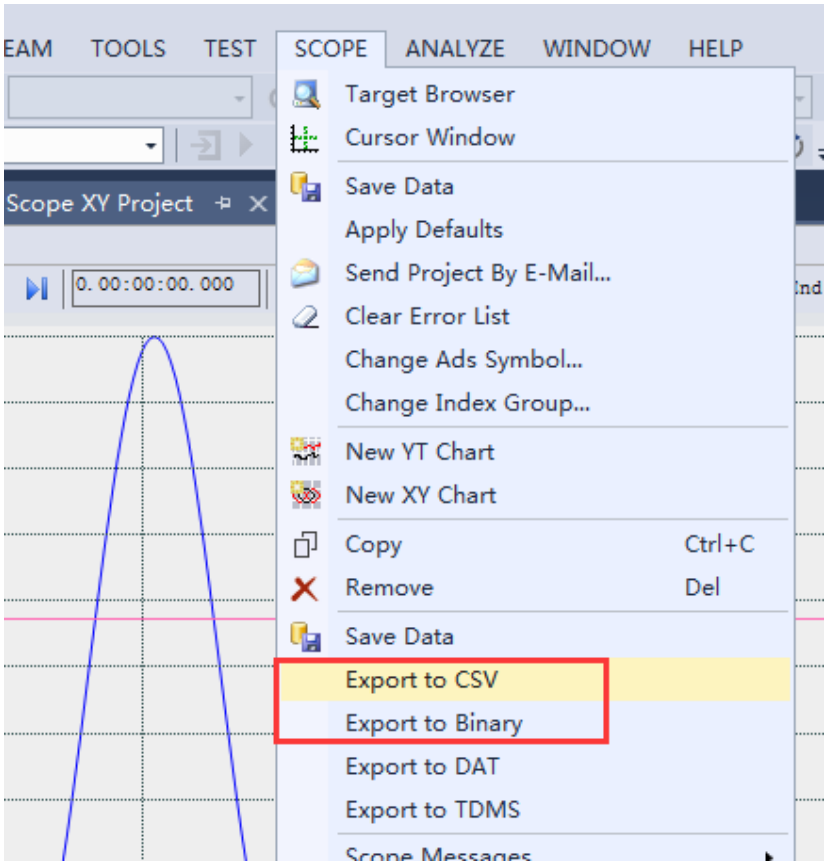


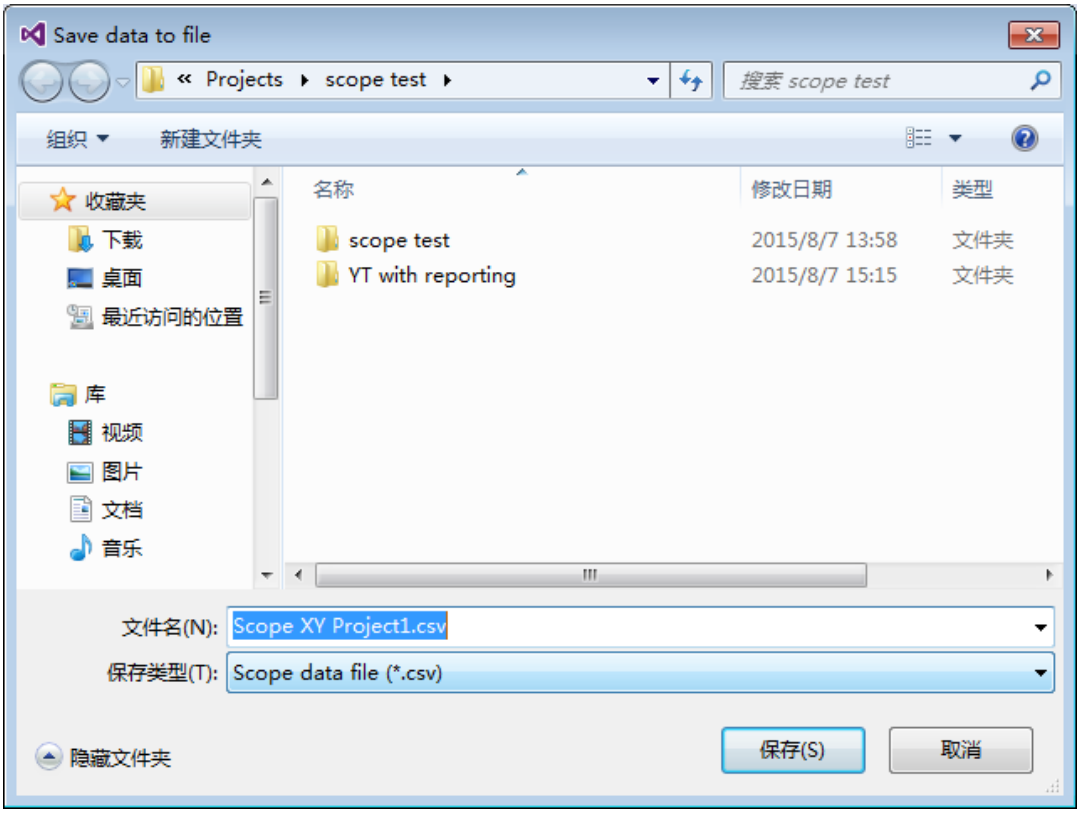


（7） 在停止 scope 当前的数据记录后，scope 的数据也可以保存为.csv、.txt 或者 binary 格式。

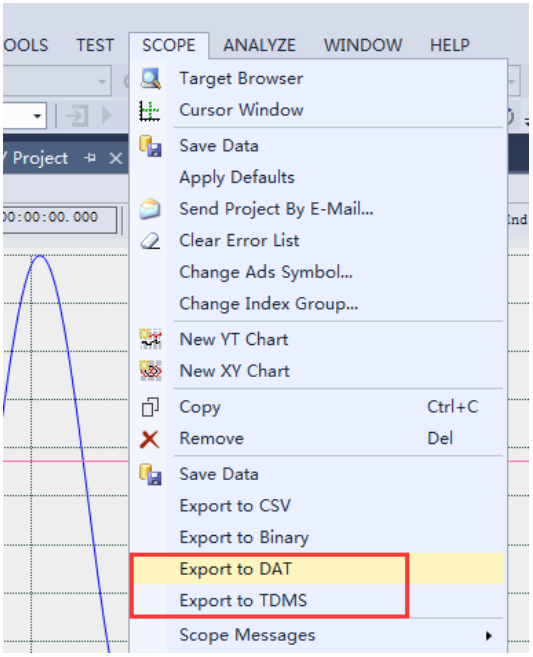
选中 scope project，在 scope 菜单下可以看到存储这些数据格式的按钮。

选 Export to CSV，然后命名新文件保存即可。





（8） 在完成了 scope 数据记录后，数据也可以保存为.tdms 和.dat 格式。这些格式广泛运用在第三方的测量软件，不过 Binary/DAT/TDMS 这三种导出只有在正式版授权的 Scope View Professional 上才可以用。



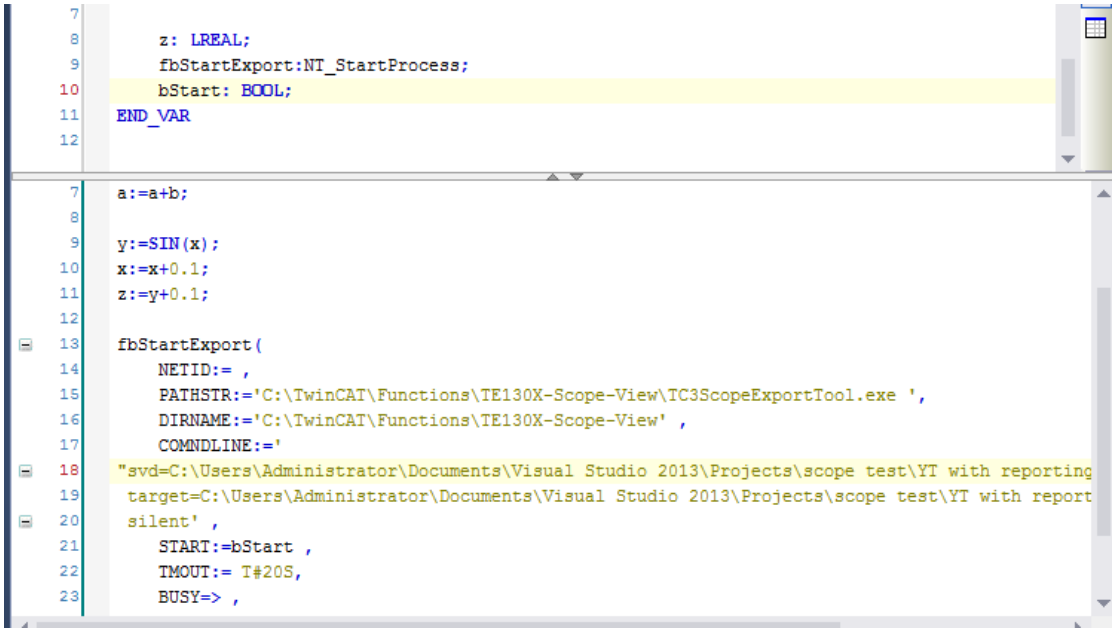
（9） 接下来讲 Automated export 功能，可以自动将程序保存下来的.svd 文件另

存为.csv 文件。

需要在 PLC 的 main 程序里添加 NT_StartProcess 功能块，如下图加入。

将在 20S 后打开 The TC3ScopeExport tool.exe 程序，在默认的路径下生成 csv 文件。

PATHSTR 处输入的是 The TC3ScopeExport tool.exe 程序路径，DIRNAME 处输入该程序所在文件夹，COMNDLINE 处输入.svd 所在路径和要保存的.csv 文件路径，加上 silent。

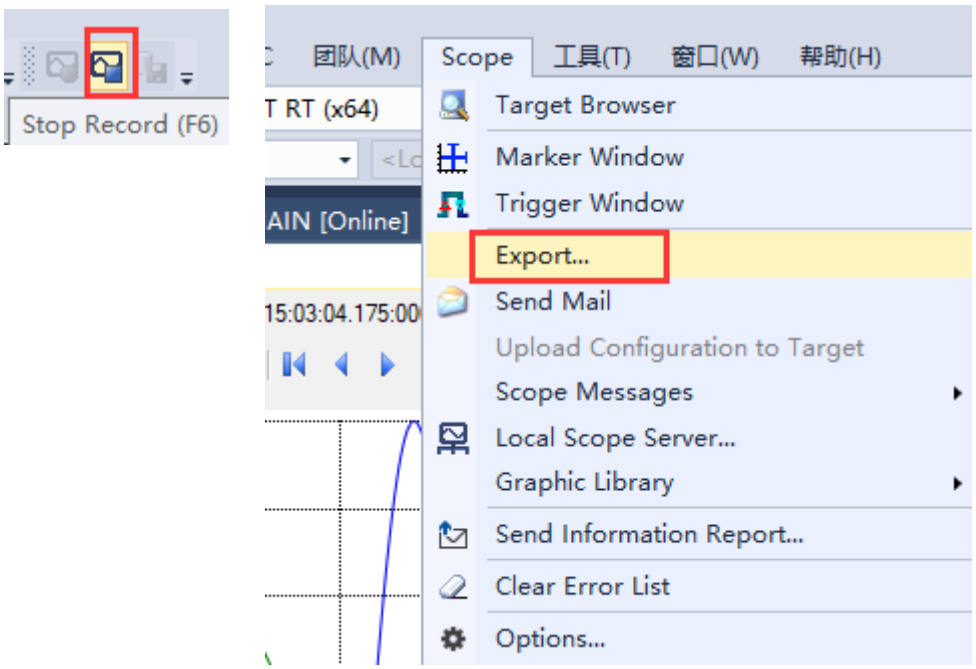


```
    z: LREAL;
    fbStartExport:NT_StartProcess;
    bStart: BOOL;
END_VAR
```

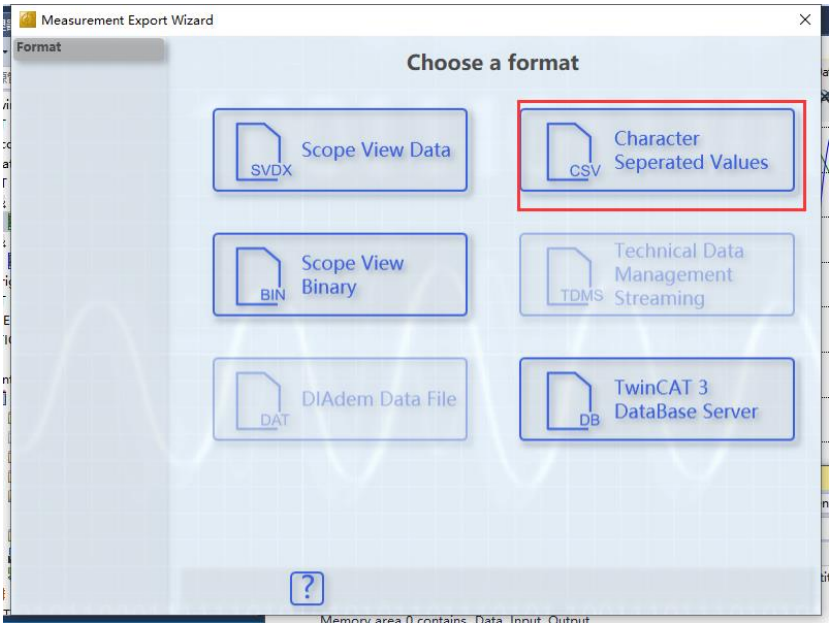
```
a:=a+b;

y:=SIN(x);
x:=x+0.1;
z:=y+0.1;

fbStartExport(
    NETID:= ,
    PATHSTR:='C:\TwinCAT\Functions\TE130X-Scope-View\TC3ScopeExportTool.exe ',
    DIRNAME:='C:\TwinCAT\Functions\TE130X-Scope-View' ,
    COMNDLINE:='
"svd=C:\Users\Administrator\Documents\Visual Studio 2013\Projects\scope test\YT with reporting
 target=C:\Users\Administrator\Documents\Visual Studio 2013\Projects\scope test\YT with report
 silent' ,
    START:=bStart ,
    TMOUT:= T#20S,
    BUSY=> ,
```

6、TwinCAT 3-Scope View 的 Saving and Exporting data （TC3.1.4024 版本）

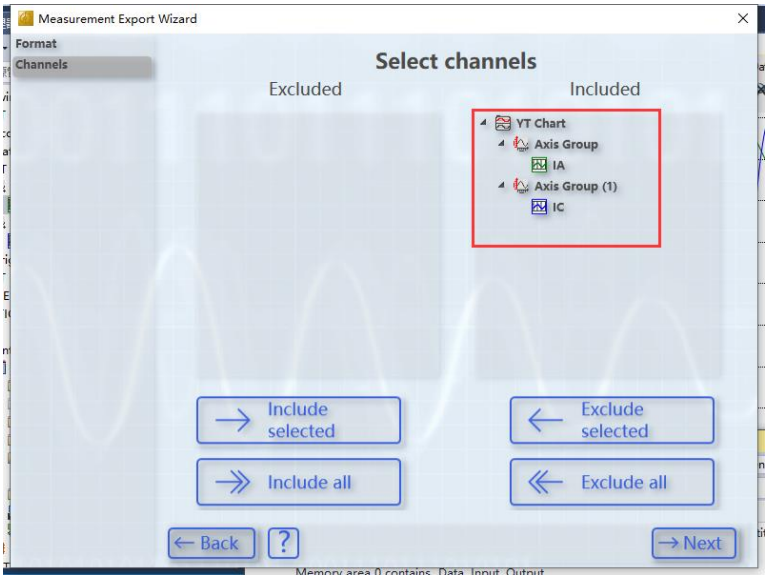
在 TC3.1.4024 版本中停止数据记录后选择工具栏的 scope，展开后选择 Export…



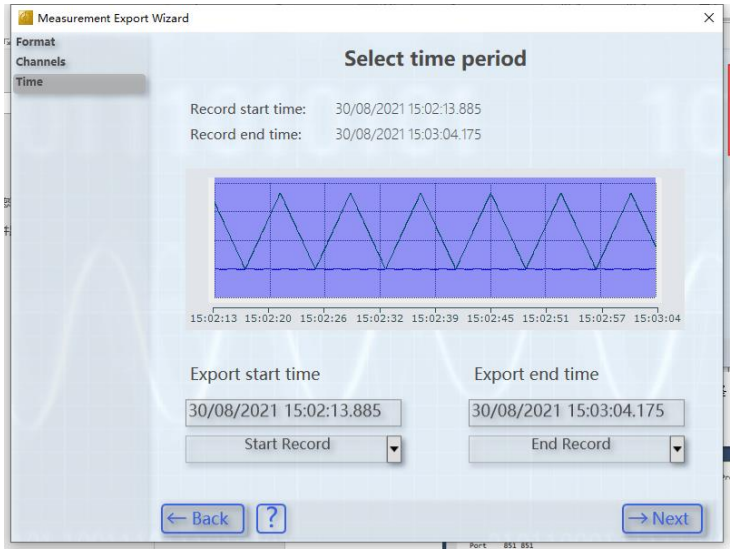
弹出选项卡后选择保存为 CSV 格式



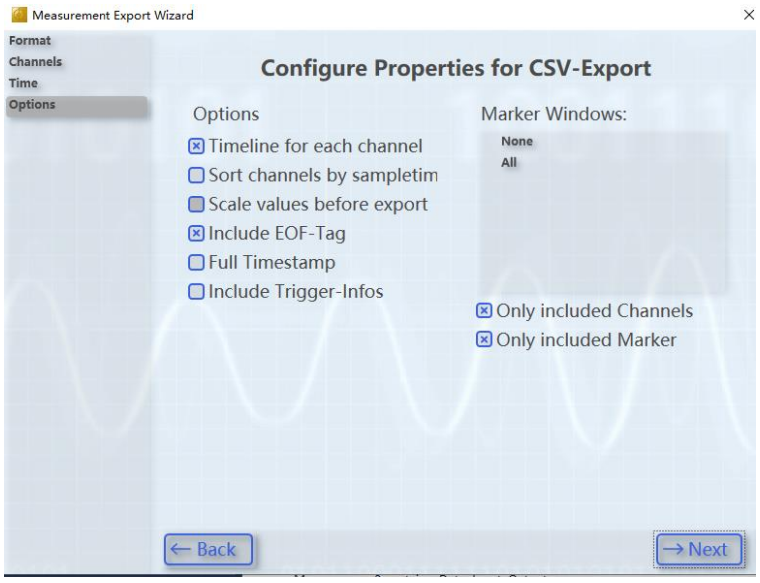
选择需要保存的变量，对不需要的变量可以选择 Exclude selected

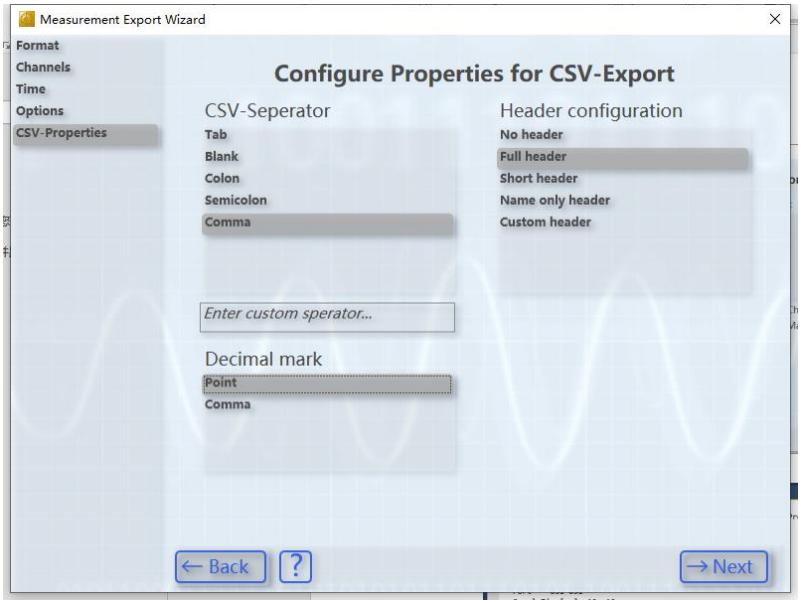


选择保存变量的时间范围

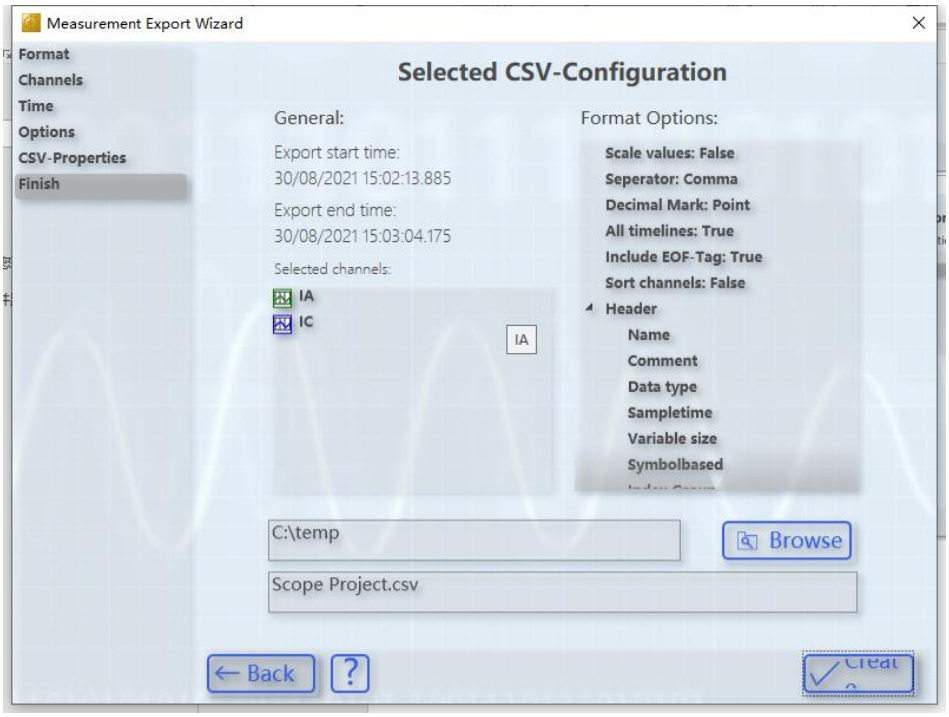


对 csv 数据的格式进行配置。

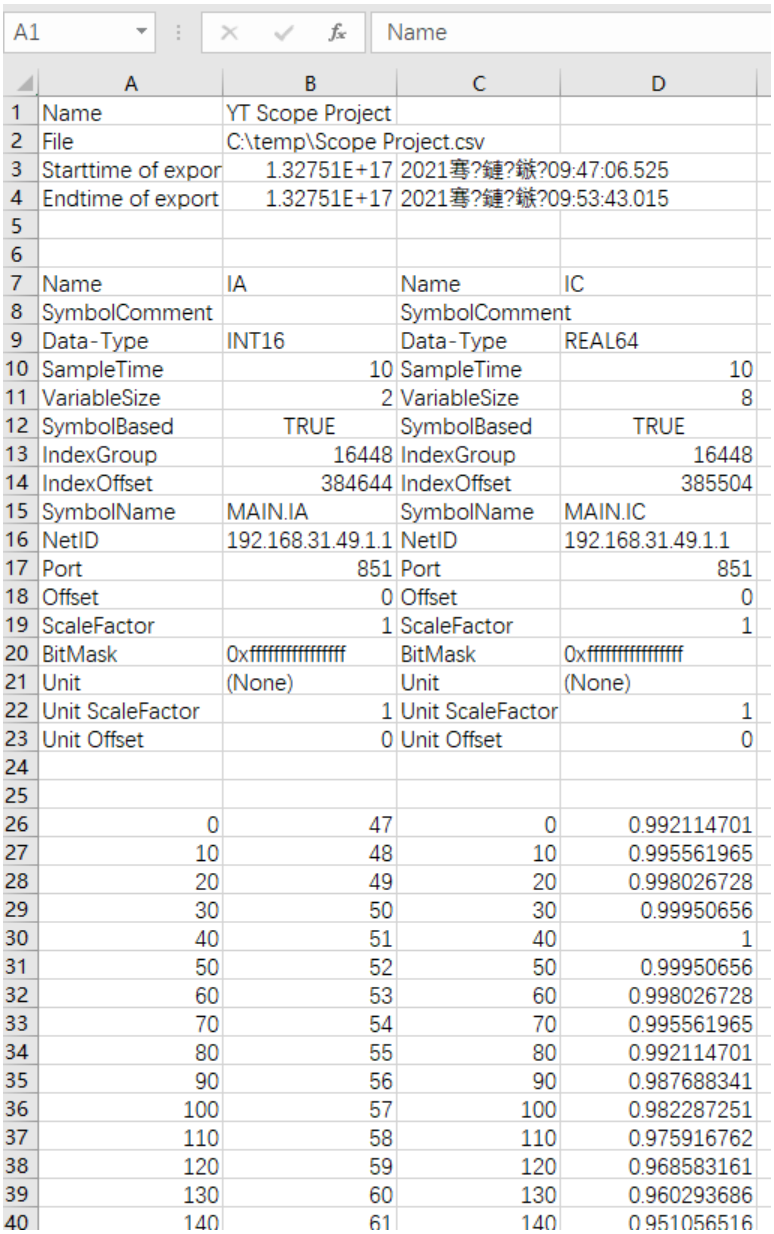




选择保存位置。

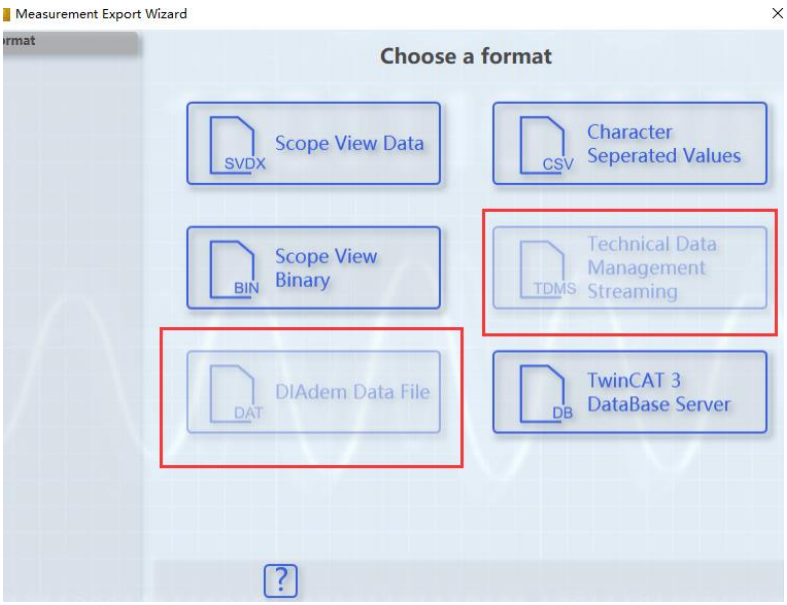


打开.csv 文件如图所示。



	A	B	C	D
1	Name	YT Scope Project		
2	File	C:\temp\Scope Project.csv		
3	Starttime of expor	1.32751E+17	2021骞?鏈?鏃?09:47:06.525	
4	Endtime of export	1.32751E+17	2021骞?鏈?鏃?09:53:43.015	
5				
6				
7	Name	IA	Name	IC
8	SymbolComment		SymbolComment	
9	Data-Type	INT16	Data-Type	REAL64
10	SampleTime	10	SampleTime	10
11	VariableSize	2	VariableSize	8
12	SymbolBased	TRUE	SymbolBased	TRUE
13	IndexGroup	16448	IndexGroup	16448
14	IndexOffset	384644	IndexOffset	385504
15	SymbolName	MAIN.IA	SymbolName	MAIN.IC
16	NetID	192.168.31.49.1.1	NetID	192.168.31.49.1.1
17	Port	851	Port	851
18	Offset	0	Offset	0
19	ScaleFactor	1	ScaleFactor	1
20	BitMask	0xffffffffffffffff	BitMask	0xffffffffffffffff
21	Unit	(None)	Unit	(None)
22	Unit ScaleFactor	1	Unit ScaleFactor	1
23	Unit Offset	0	Unit Offset	0
24				
25				
26	0	47	0	0.992114701
27	10	48	10	0.995561965
28	20	49	20	0.998026728
29	30	50	30	0.99950656
30	40	51	40	1
31	50	52	50	0.99950656
32	60	53	60	0.998026728
33	70	54	70	0.995561965
34	80	55	80	0.992114701
35	90	56	90	0.987688341
36	100	57	100	0.982287251
37	110	58	110	0.975916762
38	120	59	120	0.968583161
39	130	60	130	0.960293686
40	140	61	140	0.951056516

（8）在完成了 scope 数据记录后，数据也可以保存为.tdms 和.dat 格式。这些格式广泛运用在第三方的测量软件，不过 Binary/DAT/TDMS 这三种导出只有在正式版授权的 Scope View Professional 上才可以用。



（9）接下来讲 Automated export 功能，可以自动将程序保存下来的.svd 文件另存为.csv 文件。

需要在 PLC 的 main 程序里添加 NT_StartProcess 功能块，如下图加入。

将在 20S 后打开 The TC3ScopeExport tool.exe 程序，在默认的路径下生成 csv 文件。

PATHSTR 处输入的是 The TC3ScopeExport tool.exe 程序路径，DIRNAME 处输入该程序所在文件夹，COMNDLINE 处输入.svd 所在路径和要保存的.csv 文件路径，加上 silent。

```
    z: LREAL;
    fbStartExport:NT_StartProcess;
    bStart: BOOL;
END_VAR
```

```
a:=a+b;

y:=SIN(x);
x:=x+0.1;
z:=y+0.1;

fbStartExport(
    NETID:= ,
    PATHSTR:='C:\TwinCAT\Functions\TE130X-Scope-View\TC3ScopeExportTool.exe ',
    DIRNAME:='C:\TwinCAT\Functions\TE130X-Scope-View' ,
    COMNDLINE:='
"svd=C:\Users\Administrator\Documents\Visual Studio 2013\Projects\scope test\YT with reporting
 target=C:\Users\Administrator\Documents\Visual Studio 2013\Projects\scope test\YT with report
 silent' ,
    START:=bStart ,
    TMOUT:= T#20S,
    BUSY=> ,
```

八、 TwinCAT3 库管理

TwinCAT3 中库管理方式与 TwinCAT2 不同，更精细，更安全，通过引用的方式来调用库的占位符，使得库的调用更灵活，给库添加版本，公司名称等属性使得库的管理更便捷。因此首先我们来熟悉下 TwinCAT3 中库如何调用和管理，这里提到几个关键词：

Reference：在 TC3 中不是单纯的加载一个库，而是引用这个库的占位符。

Version：每个库被创建的时候都有各自版本号。

Company：每个库被创建的时候也都有公司名称。

Placeholder：可以实现对于不同版本的库进行管理，加载库其实就是加载一个库的占位符，随后在通过占位符选项可以设置这个库当前的版本，当前的解决方案（也就是说可以映射到另一个库），或者也可以看得到所有的内敛库，并且也可以切换这些内敛库的版本，都无需重新编译。

Namespace：每个库都可以定义命名空间，这样在不同库中有相同功能块的时候就可以利用命名空间进行区分。

Install：所有的库都需要安装才可以在 PLC 项目中被调用，也就是才可以支持加载到 library repository 中，而不是单纯的复制粘贴。

1. TwinCAT3 中的库管理

（1）首先新建 PLC 项目，右键 reference 可以有 5 个选项：

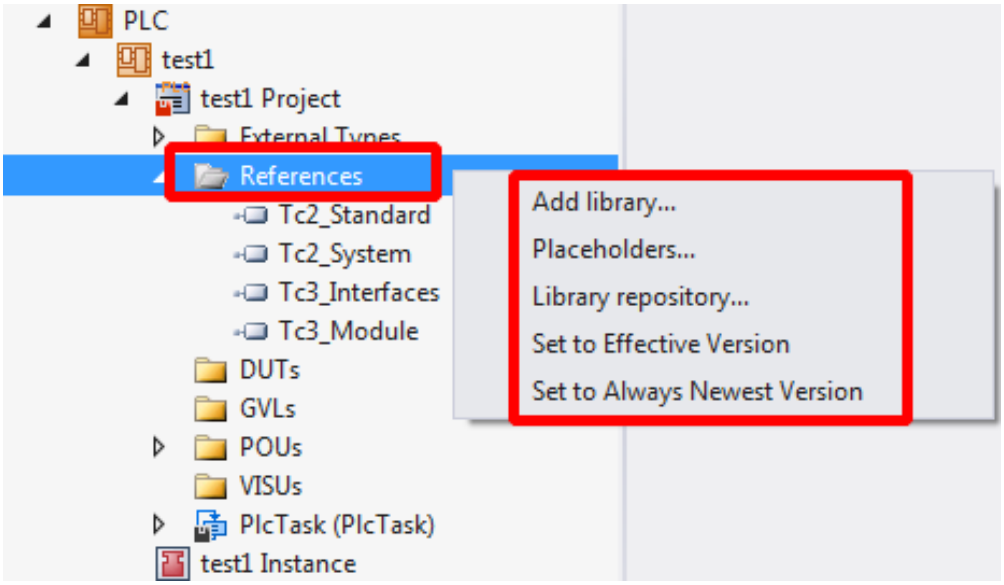
Add library：添加库。

Placeholders：查看已添加库的占位符。

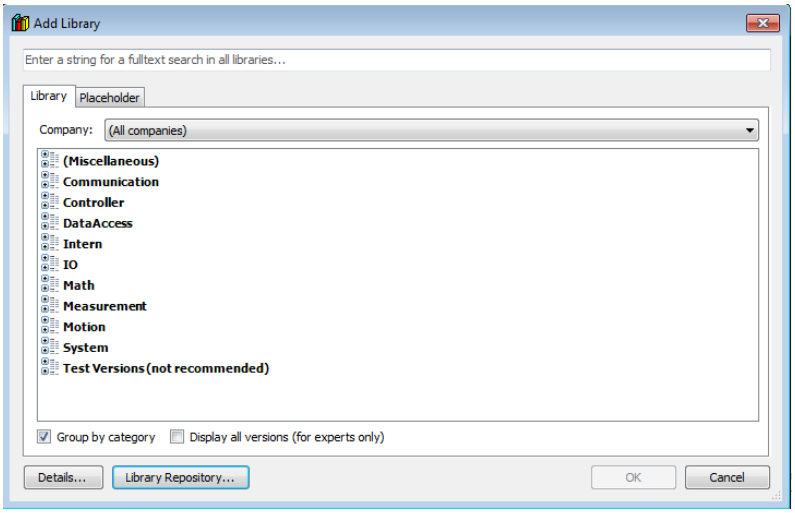
Library repository：可以在其中编辑所引用库的地址，安装、卸载库和查找相应的 FB/FC。

Set to effective version：把所有引用的占位符都设置为当前有效版本的库。

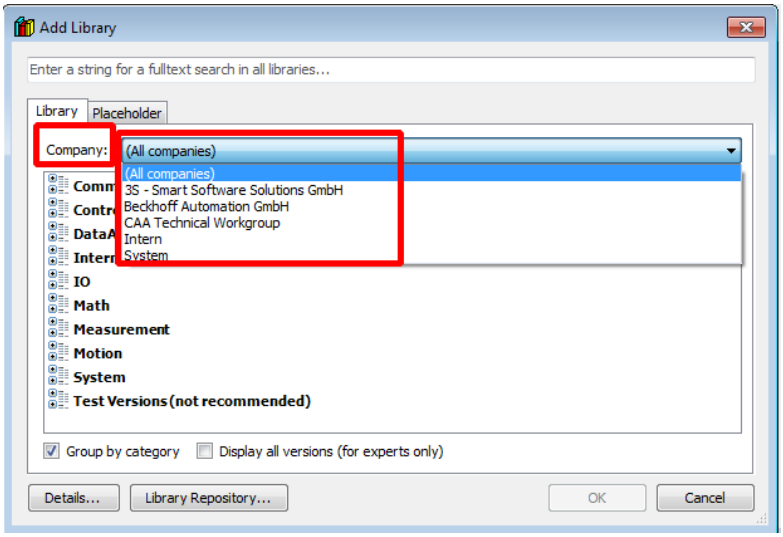
Set to always newest version：把所有引用的占位符都设置为当前最新版本的库。



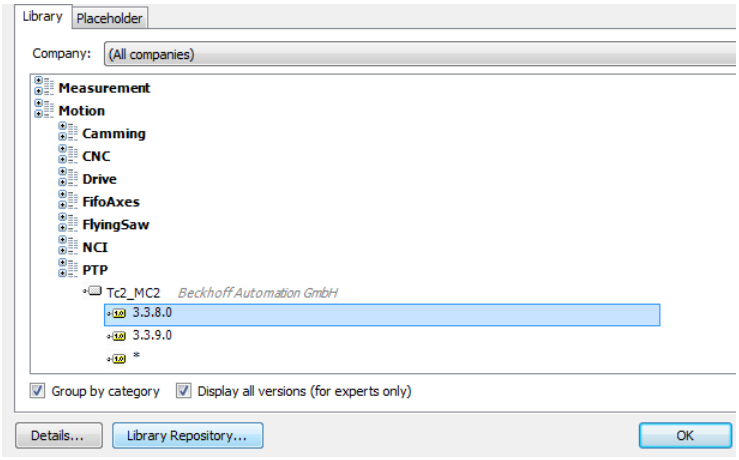
Add library 后可以看到所有库进行了分类，因此可以很方便得查找到所需要的库。



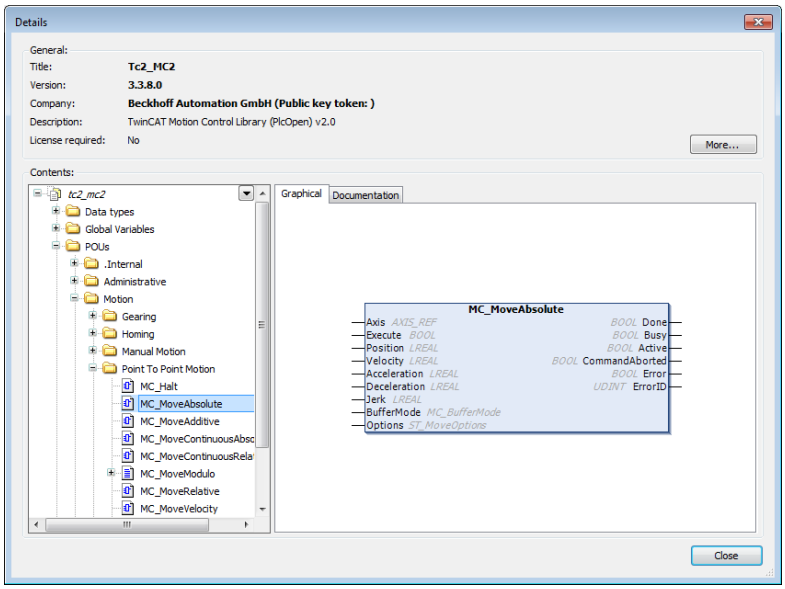
（2）在 company 中可以选择创建库的公司名称来找到相对应的库（当然大多数库都是 beckhoff automation gmbh 提供，如果客户有自己的库进行管理的时候这个功能会方便很多）。



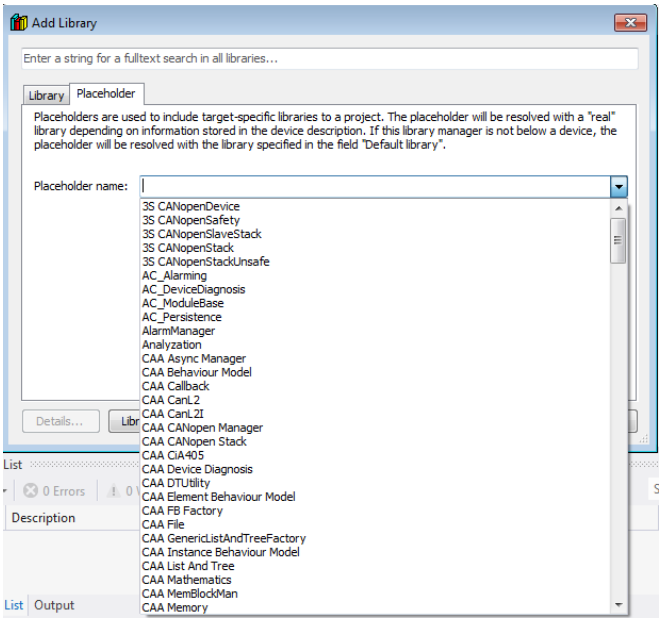
（3）如果勾选 display all versions 可以看到同一个占位符的库下面有不同的版本供选择。



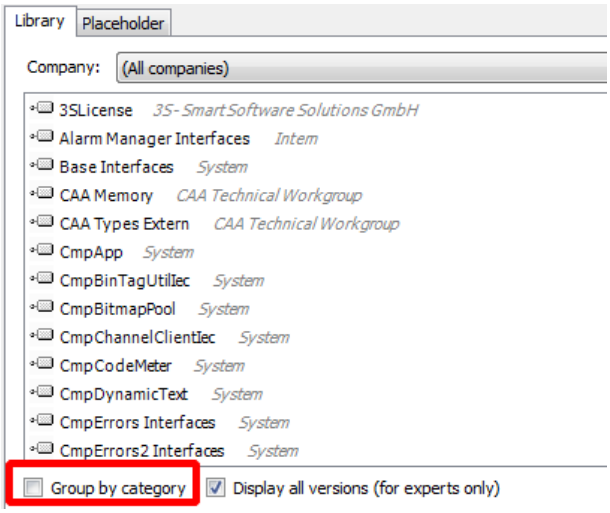
（4）点击 details 可以查看到库里面具体的内容。



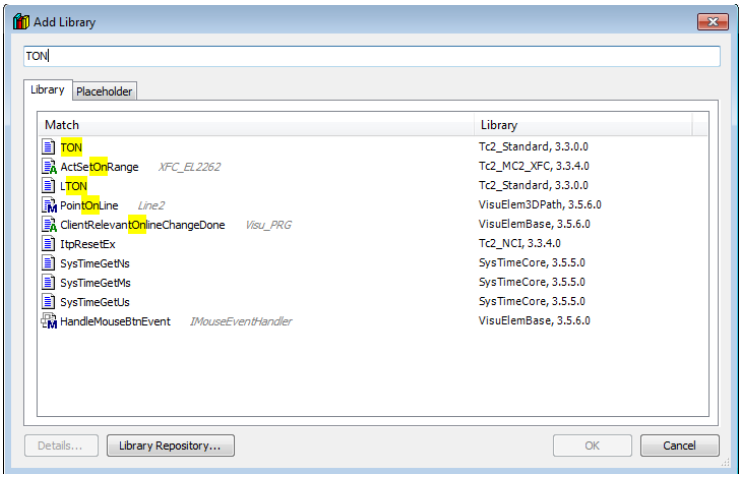
（5）也可以通过占位符来添加相应的库。



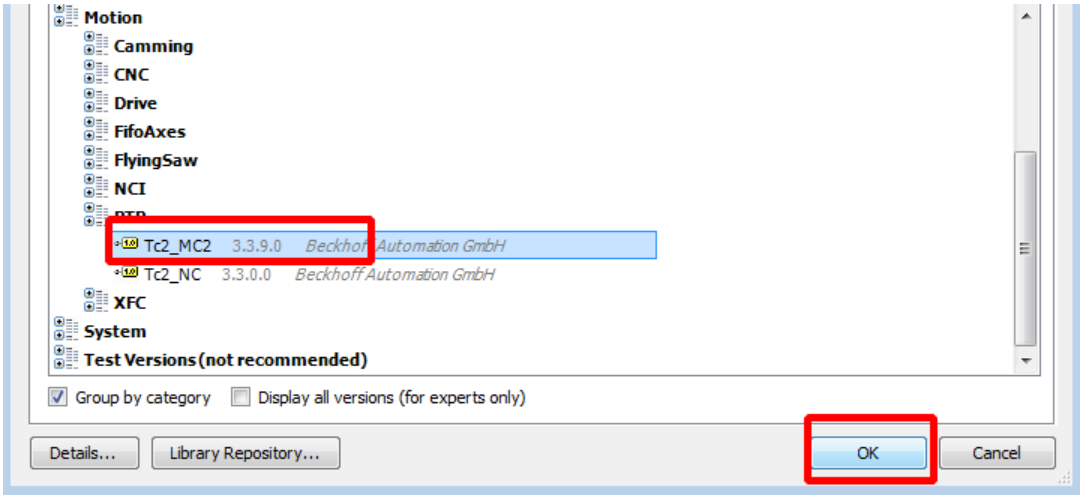
（6）当然不勾选 group by category 就可以以库字母排列来添加相对应的库。

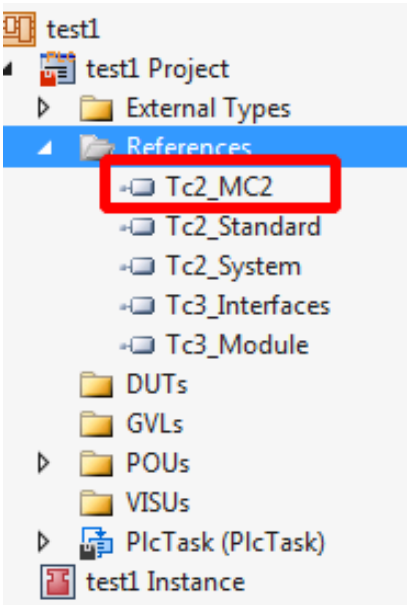


（7）在输入框可以直接输入所需要的库，甚至是输入一个功能块也可以对应到所在的库。

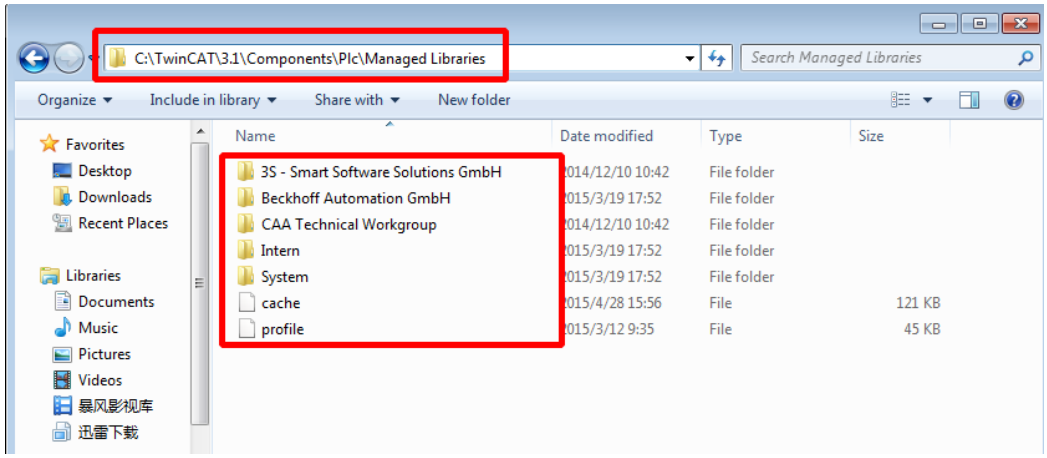


（5）不管以哪种方式，最后找到所需要的库后只需要点击 OK 就可以添加到 reference 中。



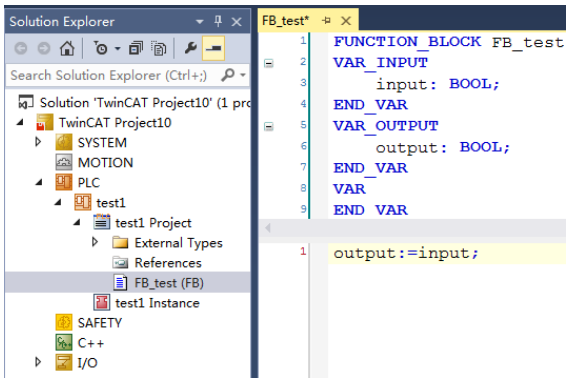


（6）当然所有库所在的路径默认都在 C:\TwinCAT\3.1\Components\Plc\Managed Libraries。

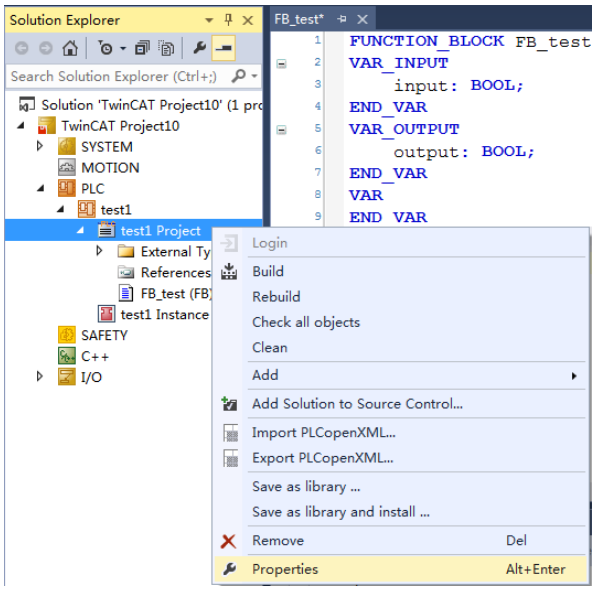


2. TwinCAT3 中新建和安装自己创建的库

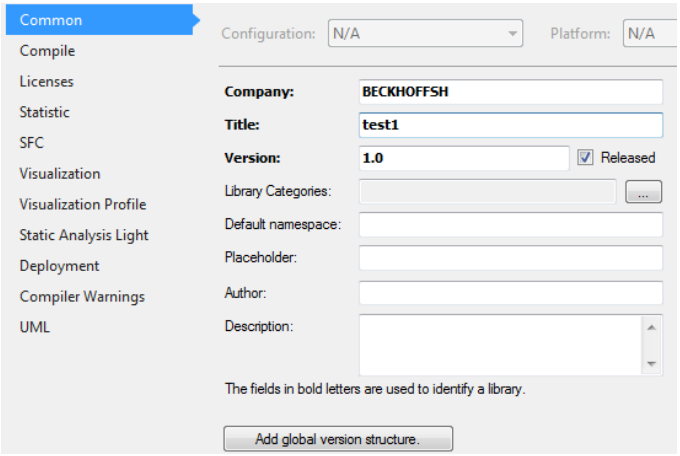
（1）首先创建一个 Empty PLC Project。然后创建简单的功能块，接下来我需要把这个功能块封装成一个库并且加载到 TwinCAT3 中进行管理。



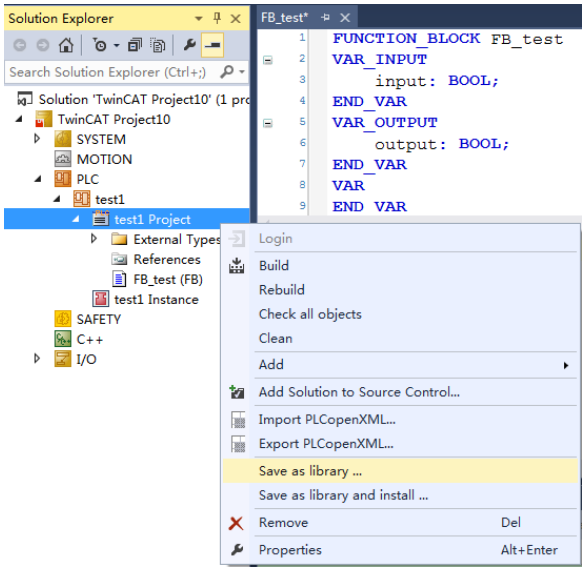
（2）安装库之前需要给这个库添加一些信息，因此右键 PLC 相对应的 project，选择 properties。



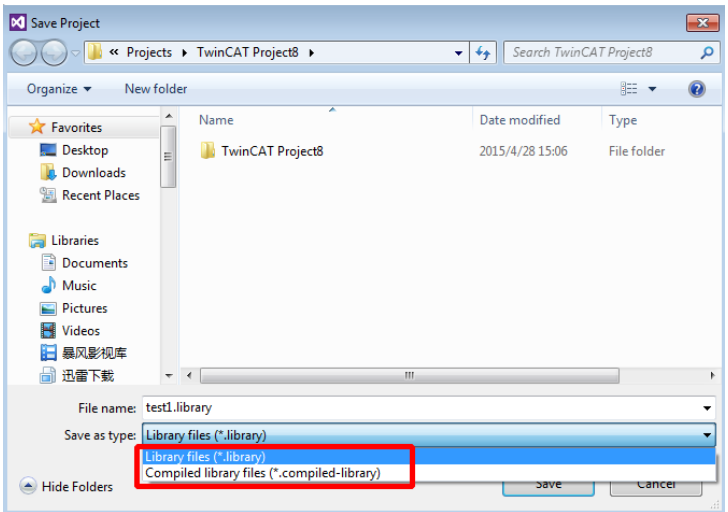
（3）随后弹出窗口，黑色粗体字选项是必填项，公司名称，库名，版本号（版本号必须是 1.0 或者 1.0.0 等等这种格式）。当然 namespace 和 placeholder 也可以单独命名，如果空着不填默认与 title 名字一致。



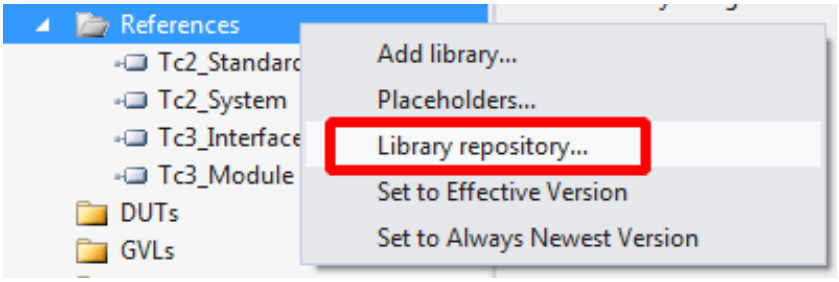
（4）随后右键 PLC 项目可以找到 2 个选项



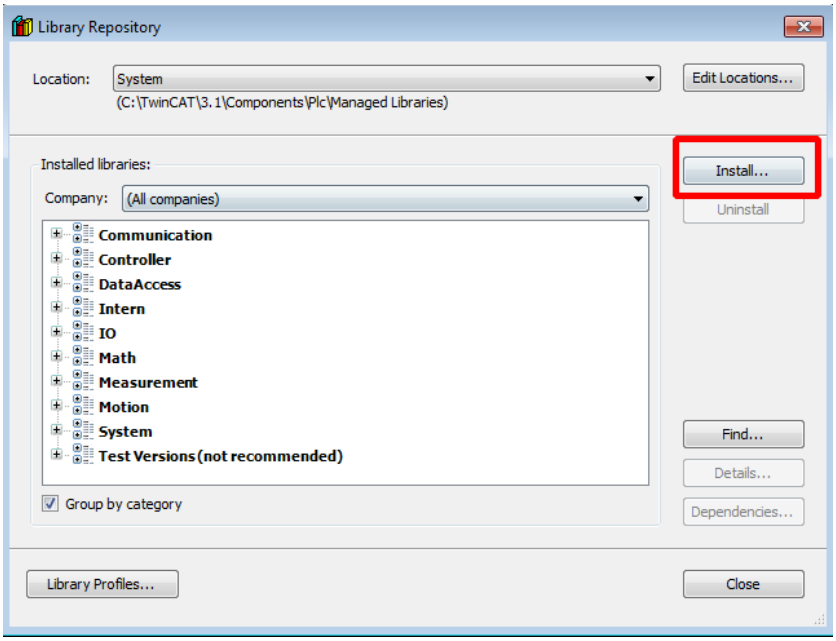
（5）save as library 是仅仅保存成库，但还没有安装到整个 library repository 中可以发现可以保存成两种后缀名的文件，.library 是可见源代码的，.compiled-library 是经过编译不可见源代码的。



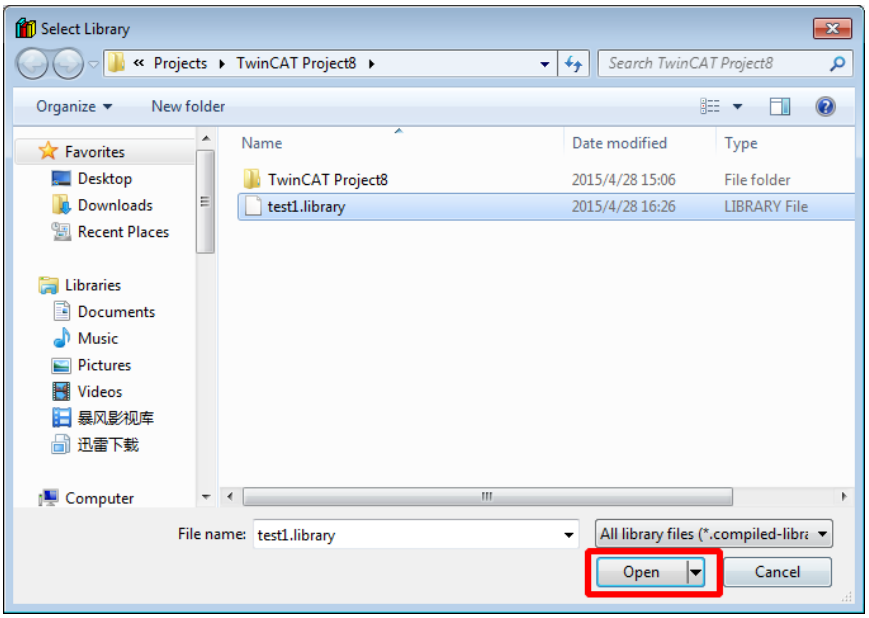
（6）一旦保存成库后，就可以把这个文件 copy 给同事或者客户进行安装，安装步骤也很简单，右键 reference 选择 library repository。



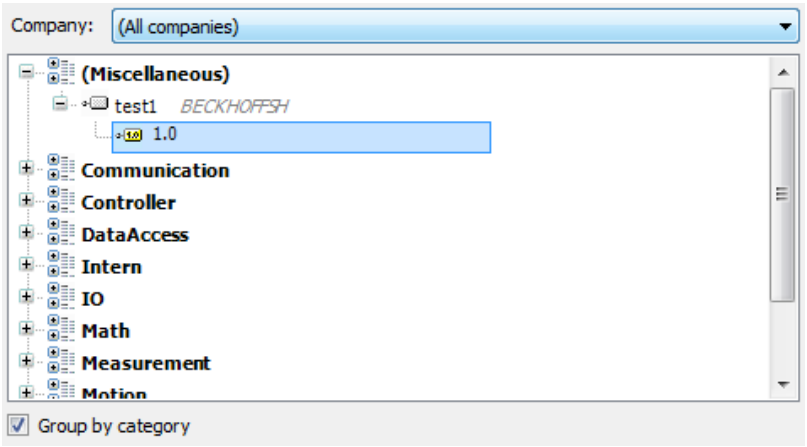
（7）随后点击 install。



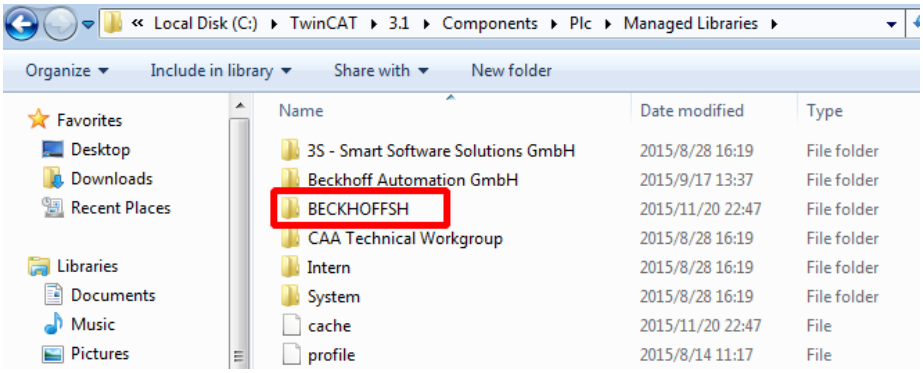
（8）选择刚才生成的库点击 open。

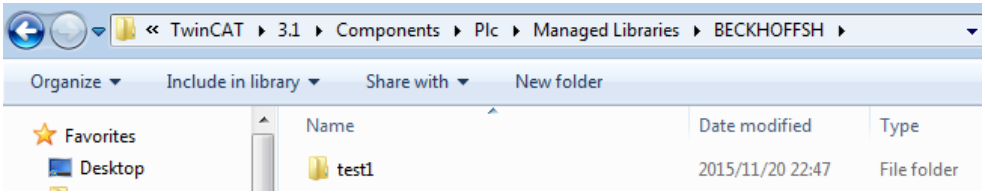


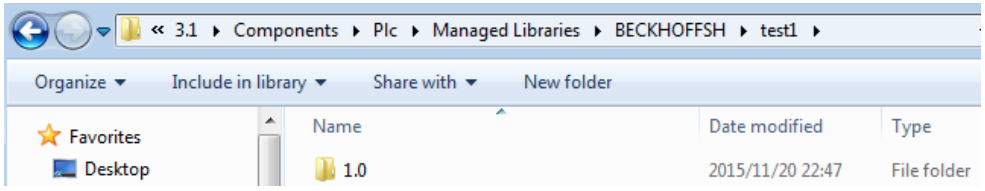
（9）那这个库就被成功安装到了整个 TwinCAT3 library 环境中。

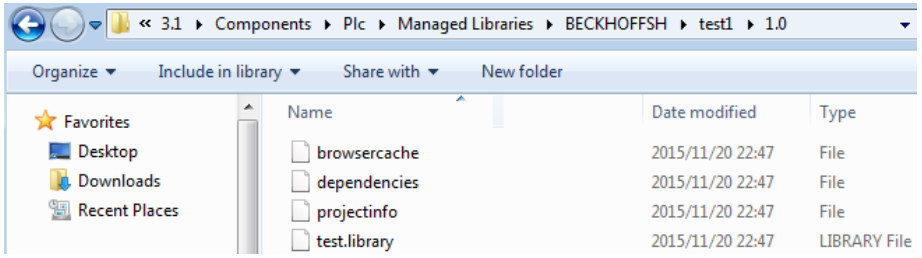


（10）我们也可以通过库所在的路径看到这个库。

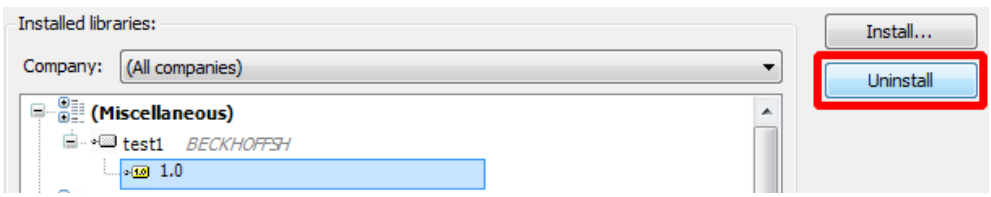




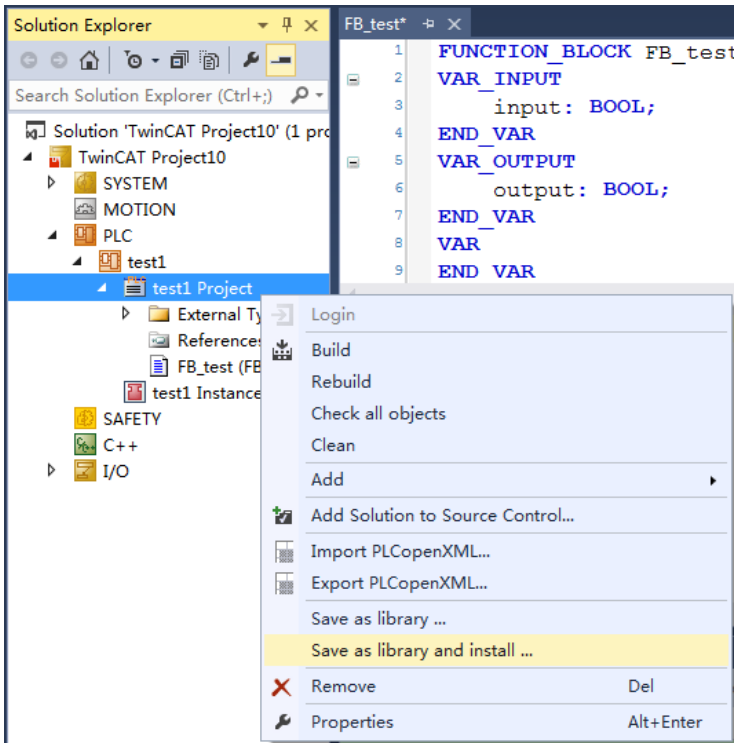




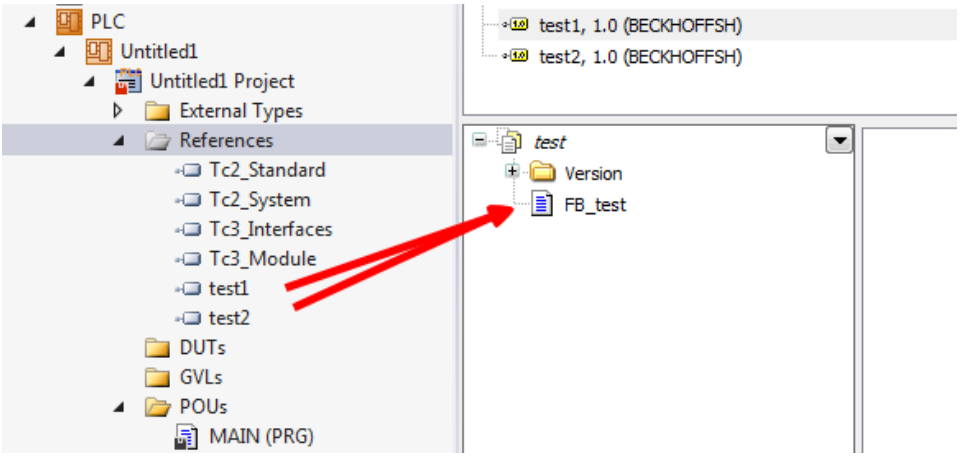
（11）当然也可以通过 uninstall 删除所安装的库。

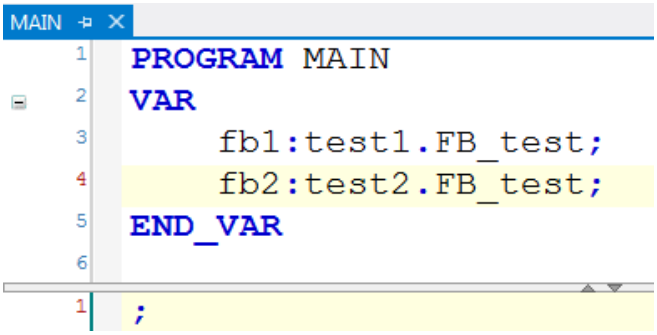


（12）Save as library and install 就一键封装成库并且安装在了本地电脑中，所以如果只是本地电脑中使用所创建的库可以一键操作方便很多。



（13）如果在一个项目中有包含多个个不同的库中，恰巧有相同名字的功能块，那在程序中给这个功能块申明的时候需要加上 namespace

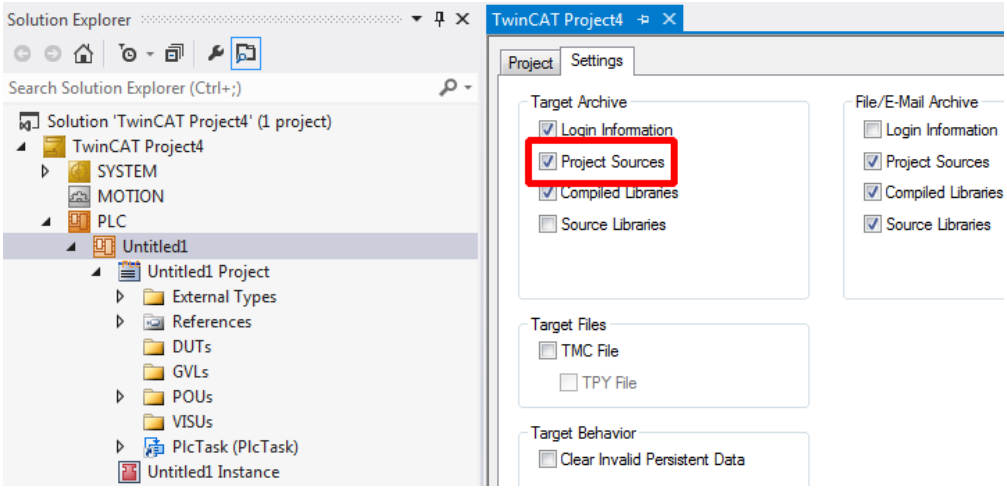




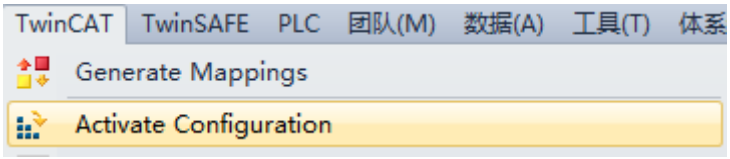
九、 TwinCAT3 代码管理

1. TwinCAT3 PLC 代码下载与上传

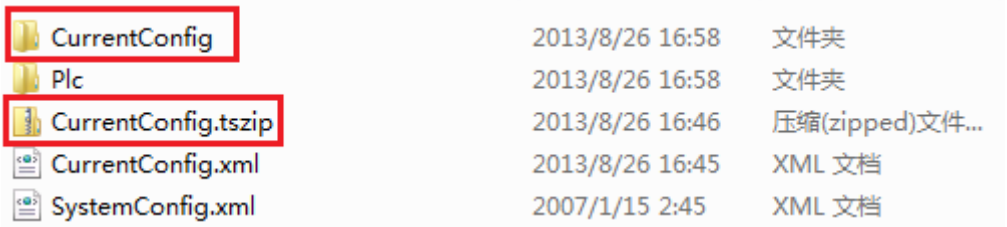
（1） 如果想把源代码下载到目标设备，方便其他人上载，可以通过以下两个步骤完成：双击 PLC 项目名，在 Settings 中把”Project Sources”打钩，如下图：（此操作就是 TC2 的 sourcecode download）。如果源代码只运行，不想被其他人上载，那就不要打钩。



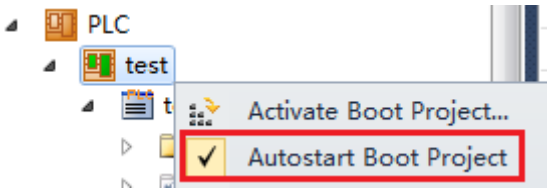
（2）通过 Activate Configuration 就可以把源代码和配置下载到目标控制器。



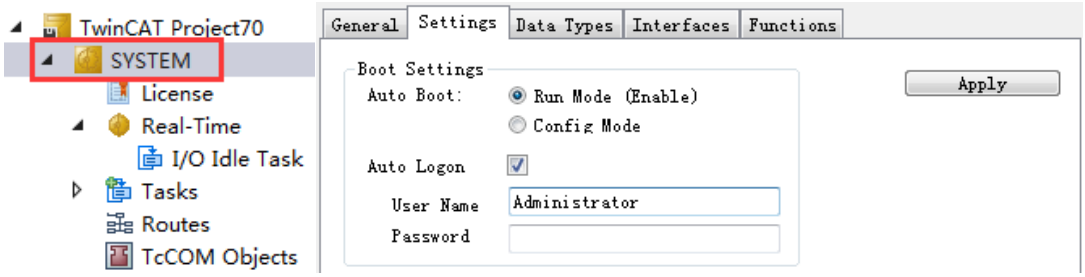
（3）完成操作后，在目标控制器中可以通过 Boot 文件夹可以发现，多出了两个文件，如下图红色方框：因此只要复制这 2 个文件到其他设备的 Boot 中就可以打开查看源代码。

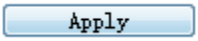


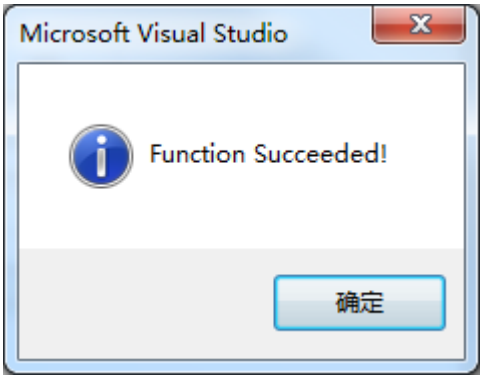
（4）如果需要开机自动运行程序，那需要做以下两个步骤，右键 PLC 项目，勾选 Autostart Boot Project。



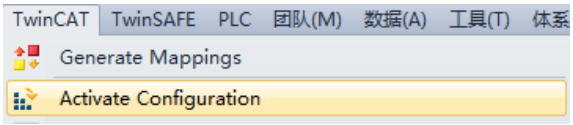
然后选择 SYSTEM 的 Settings 选项卡，在 Auto Boot 勾选 Run Mode（Enable），并且勾选 Auto Logon，在 User Name 和 Password 中填入控制器的对应用户名和密码。



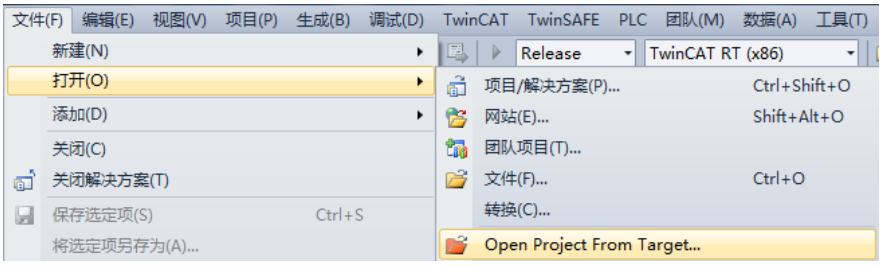
完成以后点击 Apply  ，会弹出 Function Succeeded 的弹窗，即表示设置成功。



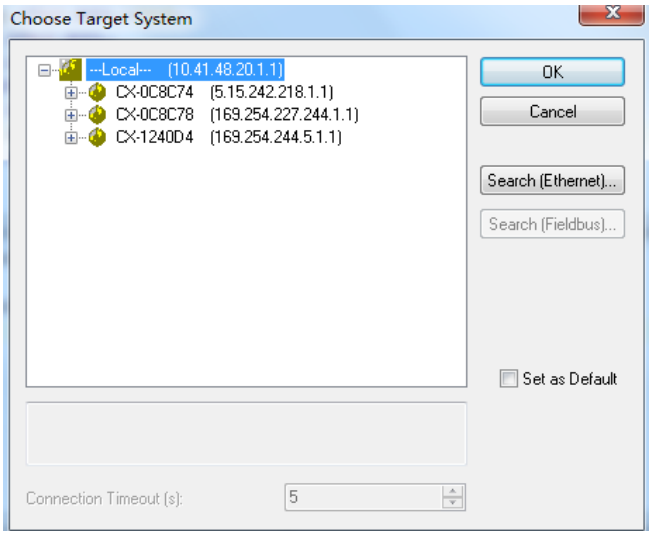
（5）Activate Boot Project：手动做 BOOT。
Autostart Boot Project：每次 Activate Configuration 的时候自动做 BOOT。
通过 Activate Configuration 就可以把源代码和配置下载到目标控制器。



（6）如果需要上载源代码，可以进行以下两个步骤，打开 TC3，选择文件→打开→Open Project From Target。



（7）随后弹出选择框，选择你需要读取源代码的目标控制器。

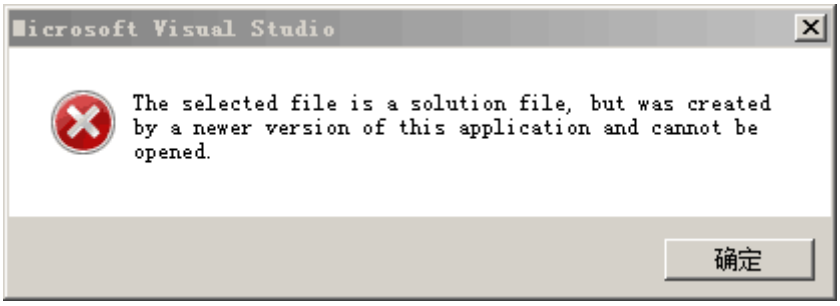


（8）如果此台目标控制器有源代码，则会继续弹出一个选择框，让你设置源代码上载后保存的路径。

（9）选择好路径后，点击确认，源代码就可以被上载上来了。

2. TwinCAT3 代码保存

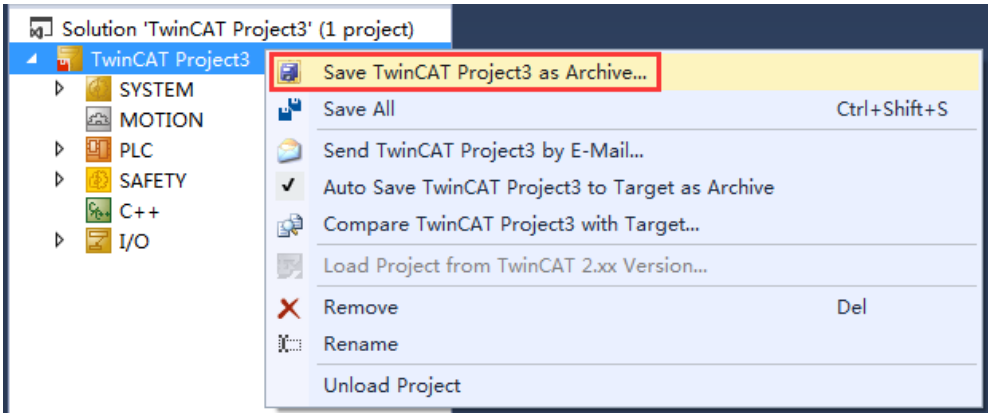
对于源代码的保存，通常是直接将项目保存，程序会自动生成一个文件夹包含所有配置和代码。但是当我们在拥有不同 VisualStudio 版本的电脑上打开项目时，可能会产生如下报错。

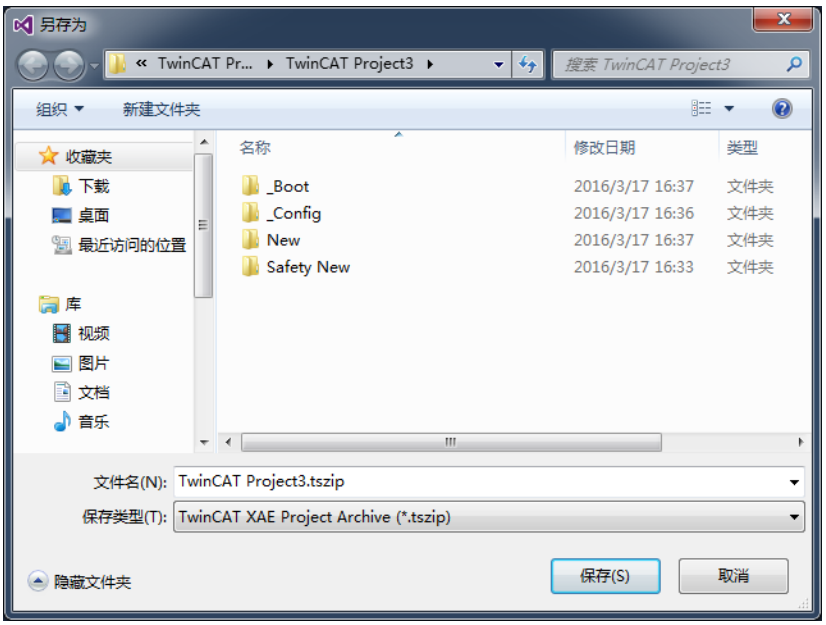


为了避免这种情况，我们可以选择保存成其他格式。

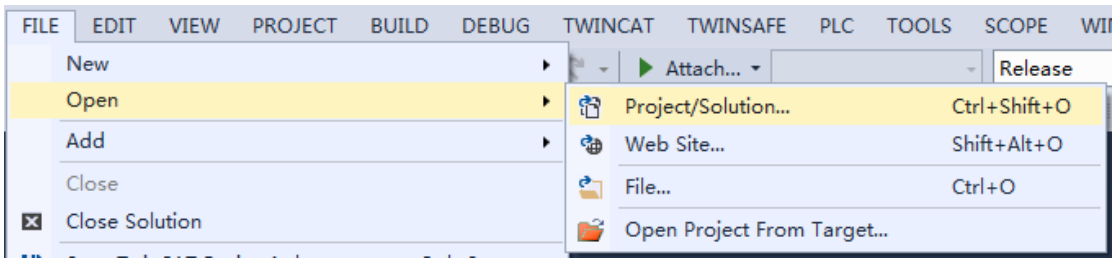
2.1 整个项目的保存与打开

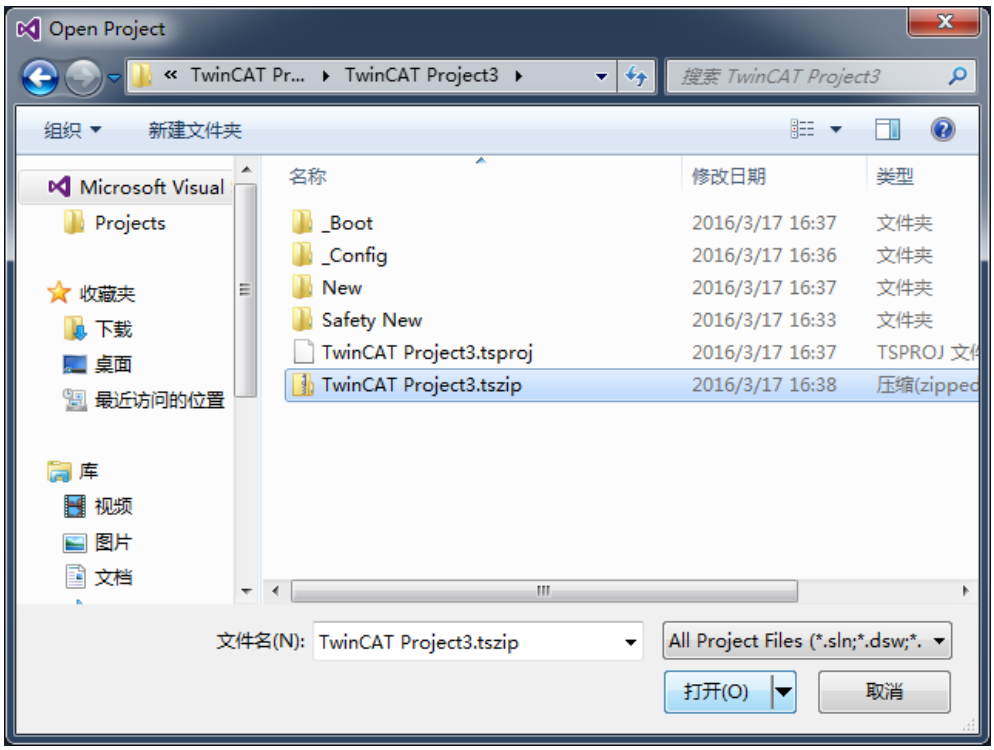
（1）保存项目：右侧资源管理窗口，右键 TwinCAT Project3，选择 Save TwinCAT Project3 as Archive...，在弹出的对话框中选择保存路径并将其另存为 tszip 的格式，点击保存。



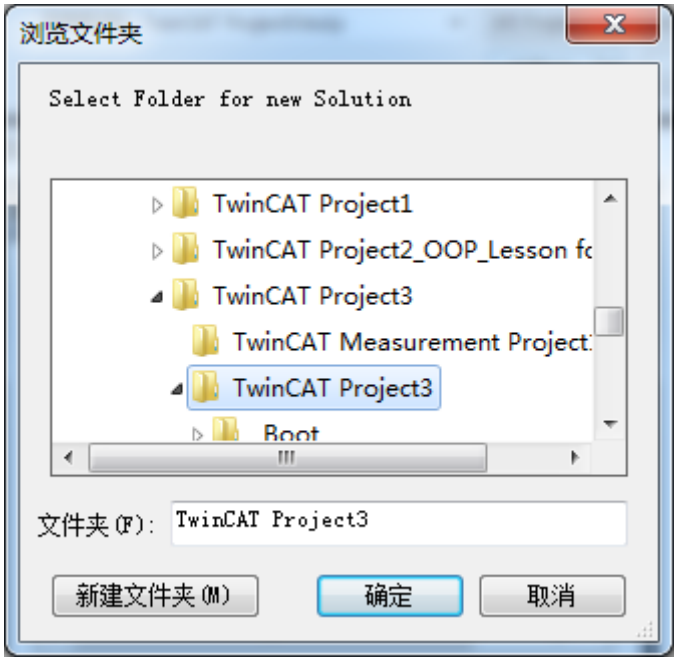


（2）打开项目：选择菜单栏 FILE，点击 Open Project/Solution…，在弹出的对话框中选择保存好的文件，单击打开。



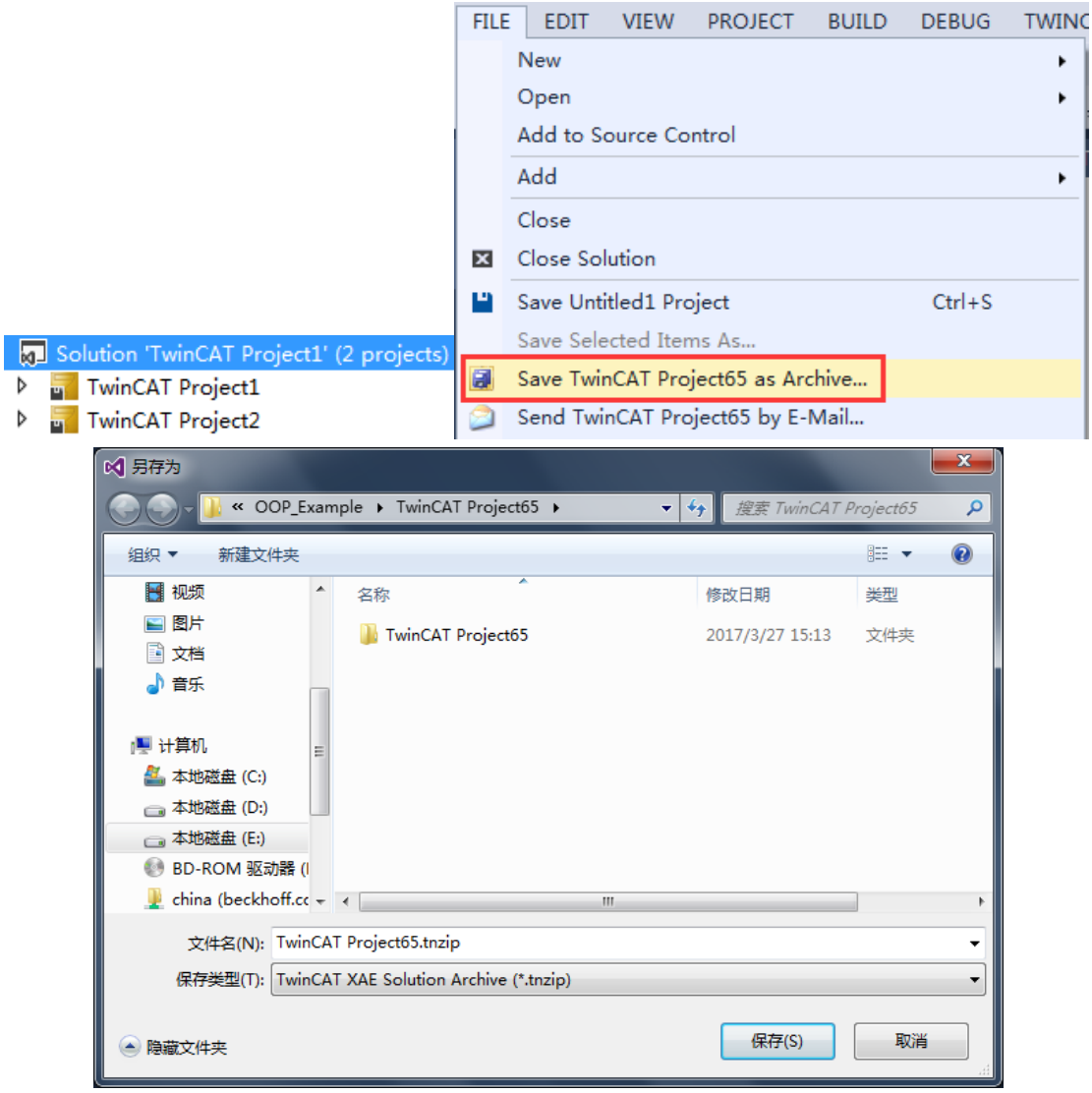


在弹出的对话框中，选择展开此项目的路径，点击确定。

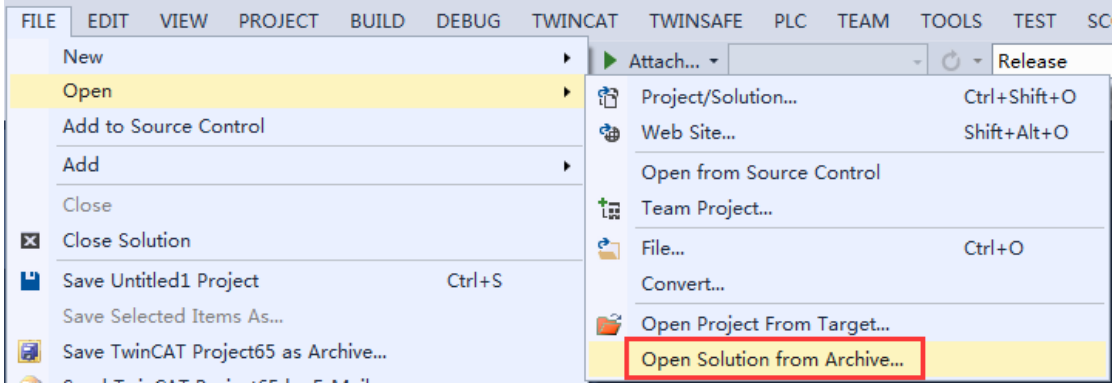


当然，整个工程也可以 tnzip 的格式保存。可以方便的将同一个 solution 下的多个 TwinCAT Project 都打包在一起。

（3）选择整个 Solution，选择菜单栏 FILE，点击 Save TwinCAT Project as Archive...，在弹出的对话框中选择所需保存的路径，并将其保存成 tnzip 的格式。

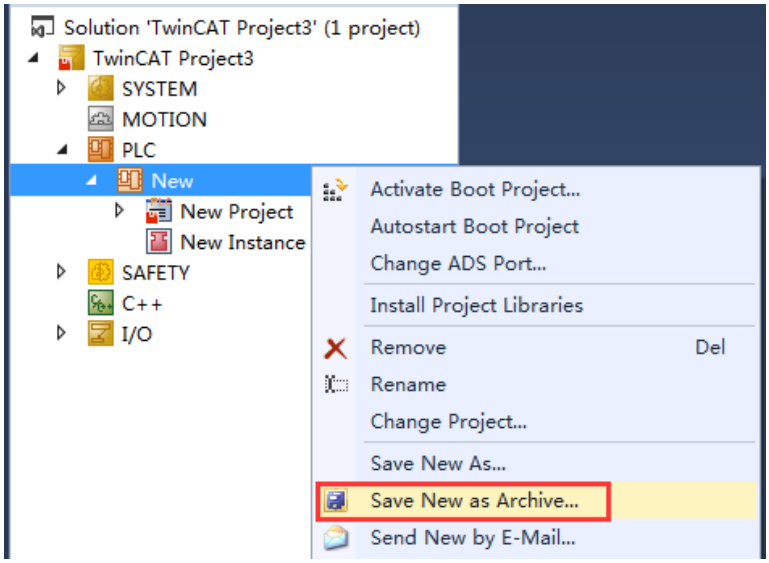


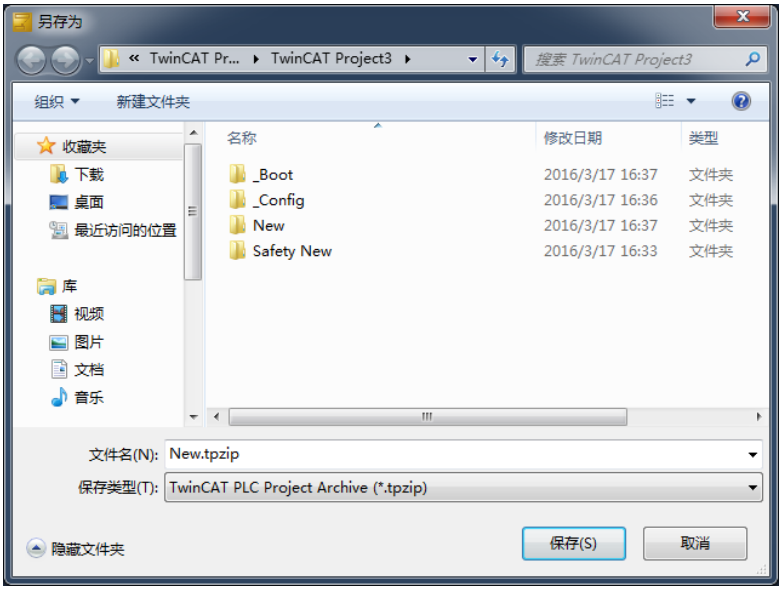
（2）打开项目：选择菜单栏 FILE，点击 Open Solution from Archive...，在弹出的对话框中选择保存好的文件，单击打开。选择展开此项目的路径，点击确定。



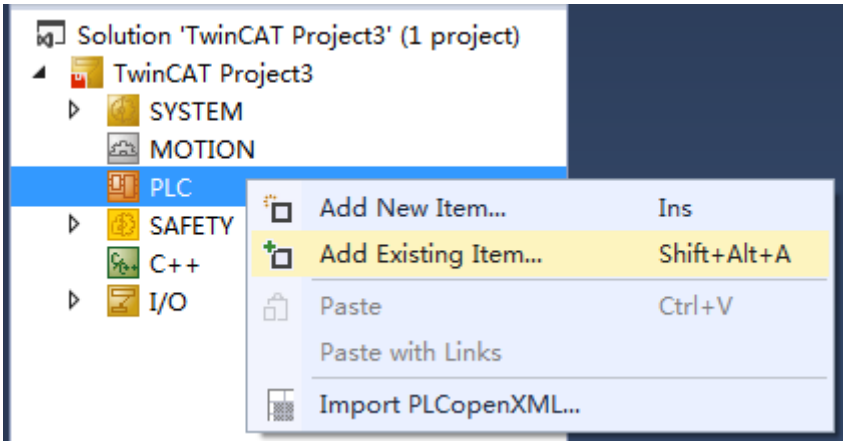
2.2 PLC 工程的保存与打开

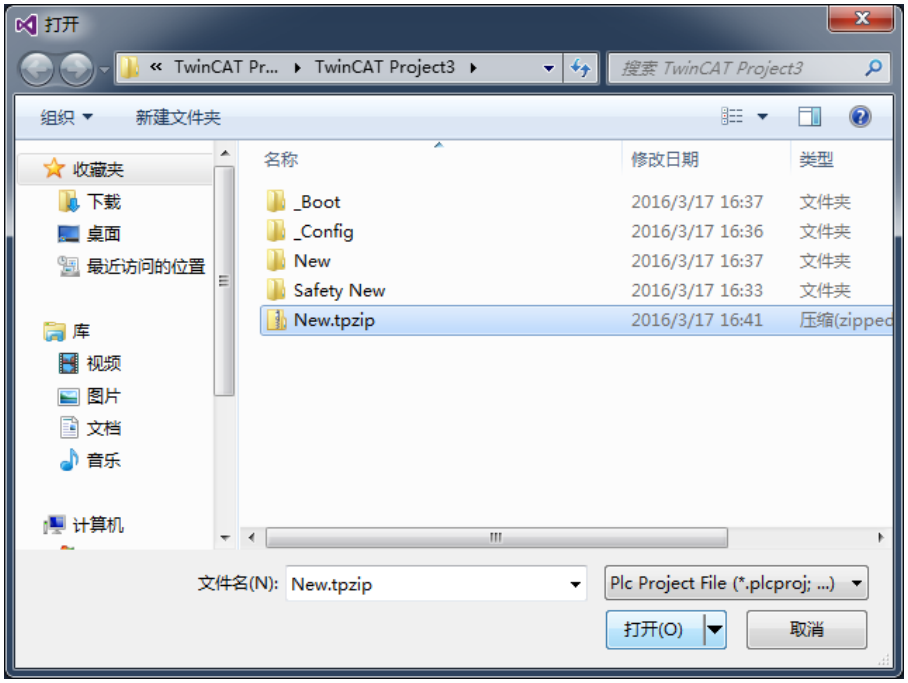
（1）保存工程：右键 PLC 下的 New，选择 Save New as Archive...，在弹出的对话框中选择保存路径并将其另存为 tpzip 的格式，点击保存。





（2）打开工程：右键 PLC，选择 Add Existing Item...，在弹出的对话框中选择保存好的文件，单击打开。

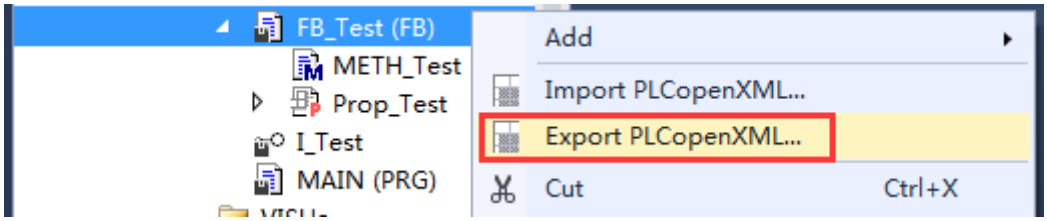


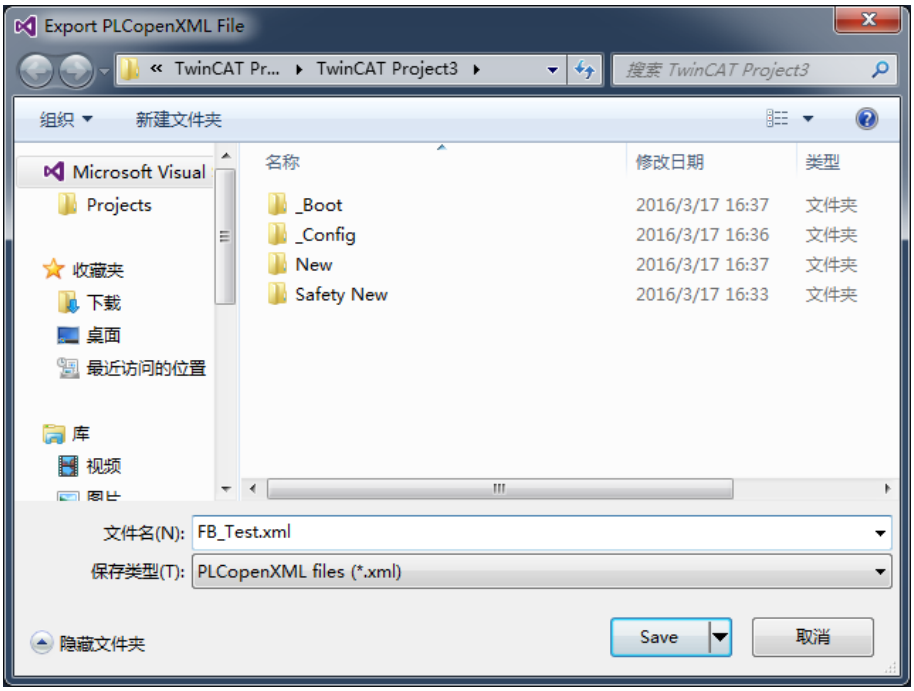


2.3 Program、Function Block、Function、interface、visualization 的保存与打开

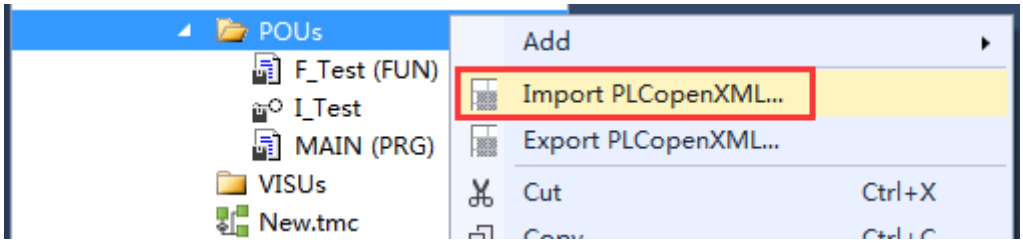
Program、Function Block、Function、interface 的保存与打开基本上都是一样的，这里以功能块和 visualization 为例。

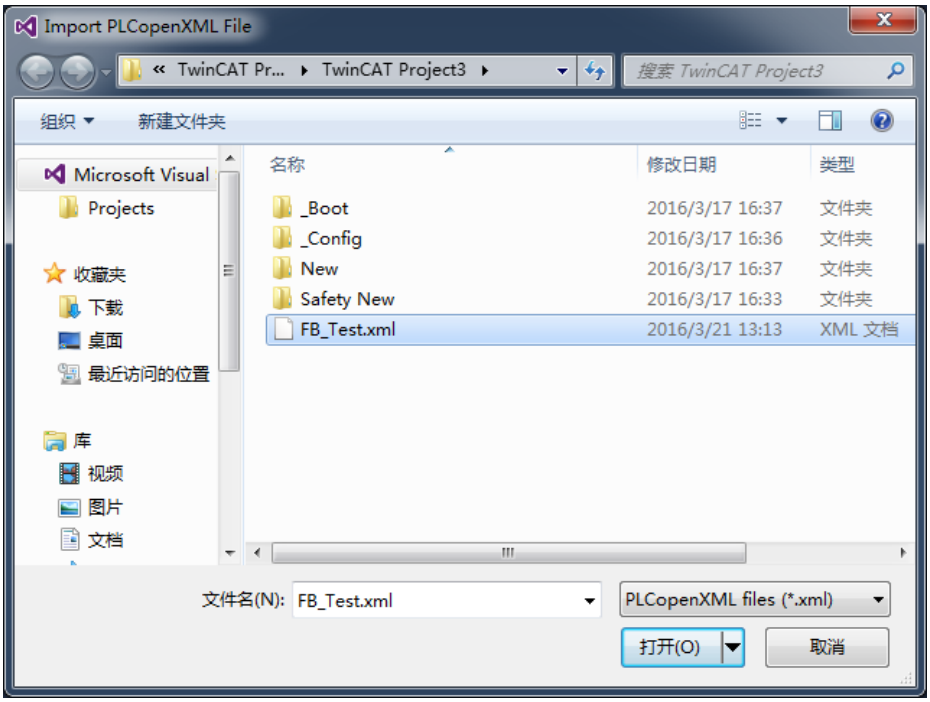
（1）保存功能块：右键 POUs 文件夹下的 New，选择 Export PLCopenXML...，在弹出的对话框中选择保存路径并将其另存为 xml 的格式，点击保存。



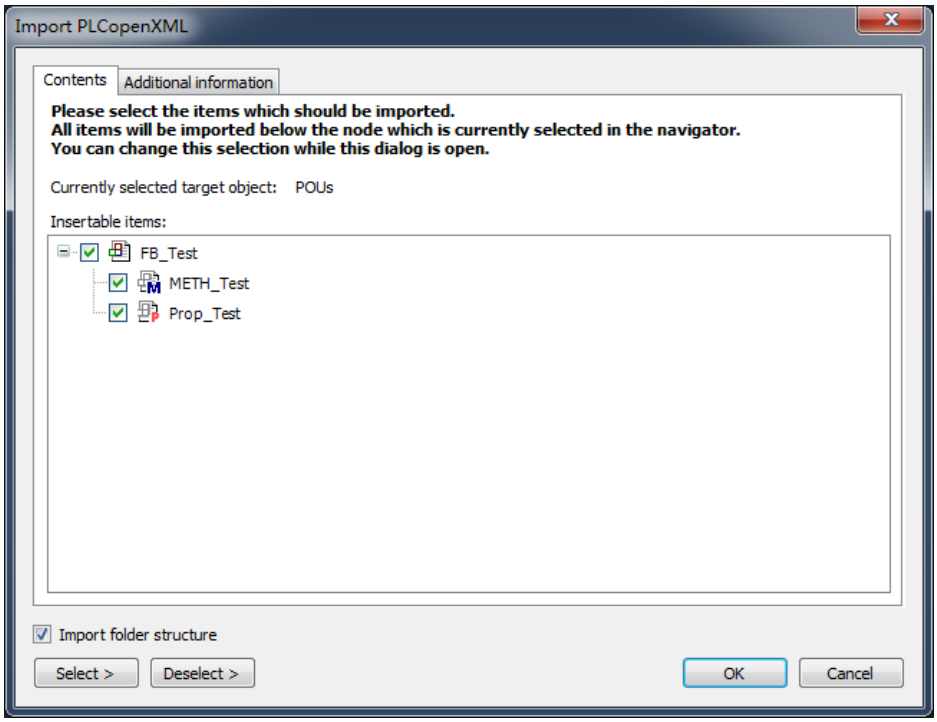


（2）打开功能块：右键 POUs 文件夹，选择 Import PLCopenXML...，在弹出的对话框中选择保存好的文件，单击打开。

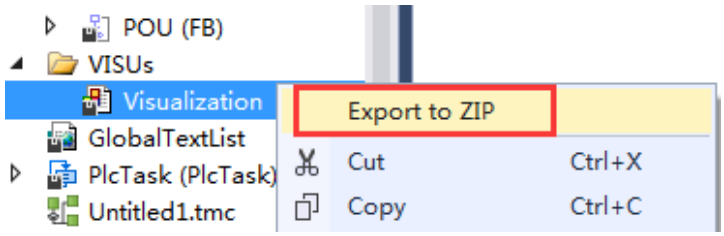




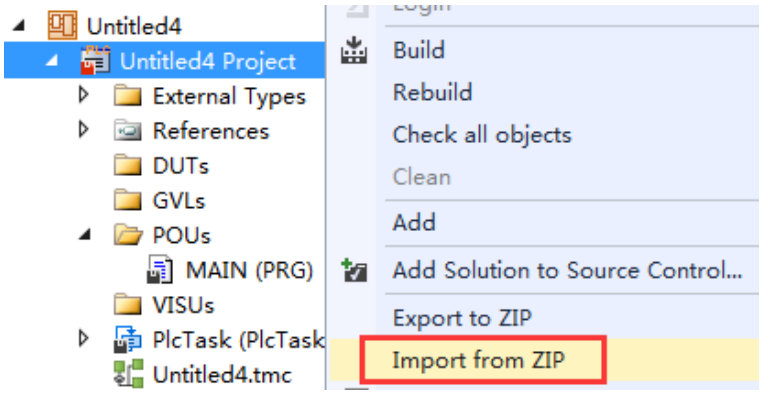
在弹出的对话框中可以勾选需要添加的已有 Method 和 Property（默认全选），点击 OK 打开。

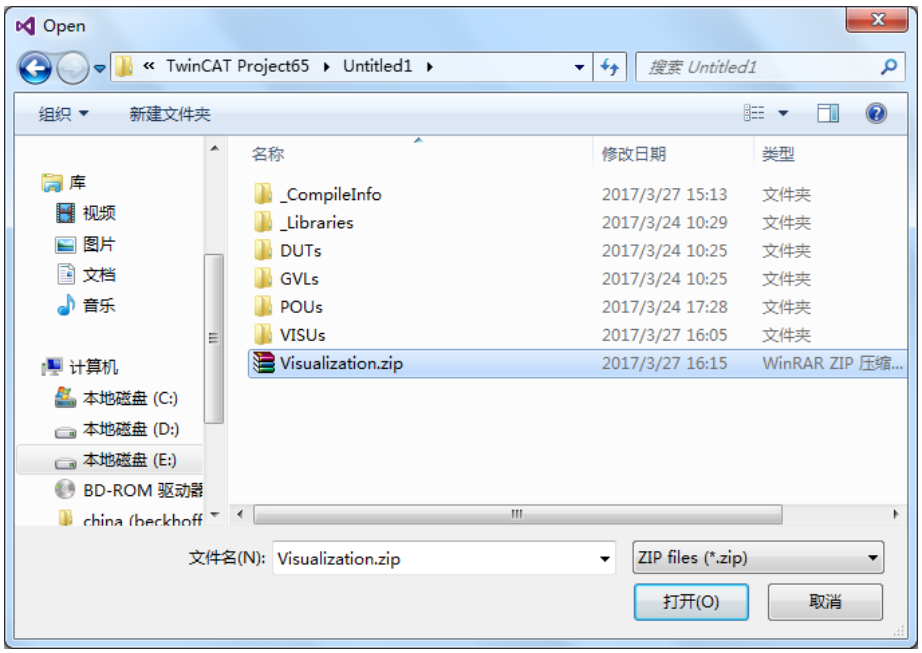


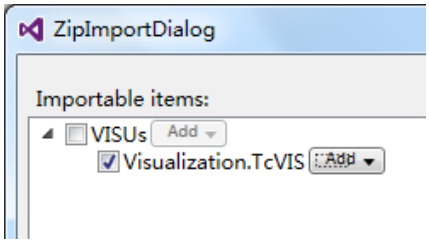
（3）Visualization 也是可以保存的，只不过不是以 XML 文件的形式，而是以 ZIP 的格式。



（4）打开 Visualization：右键，选择 Import from ZIP，在弹出的对话框中选择保存好的文件，单击打开，勾选需要的画面即可。



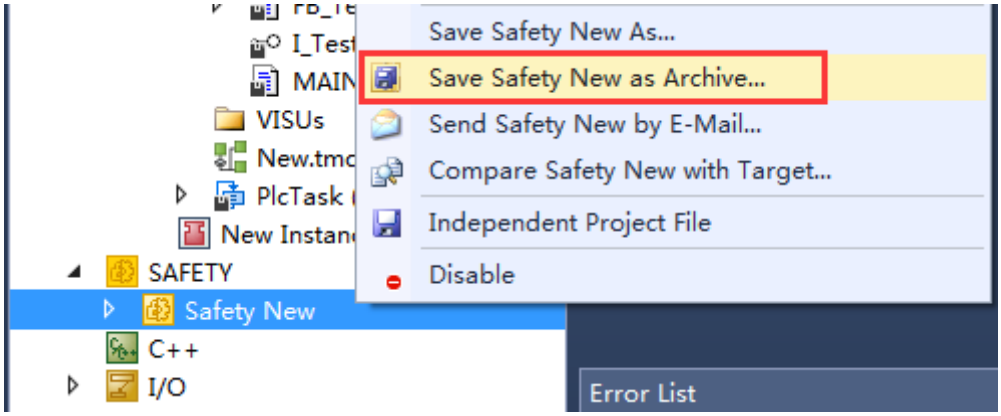


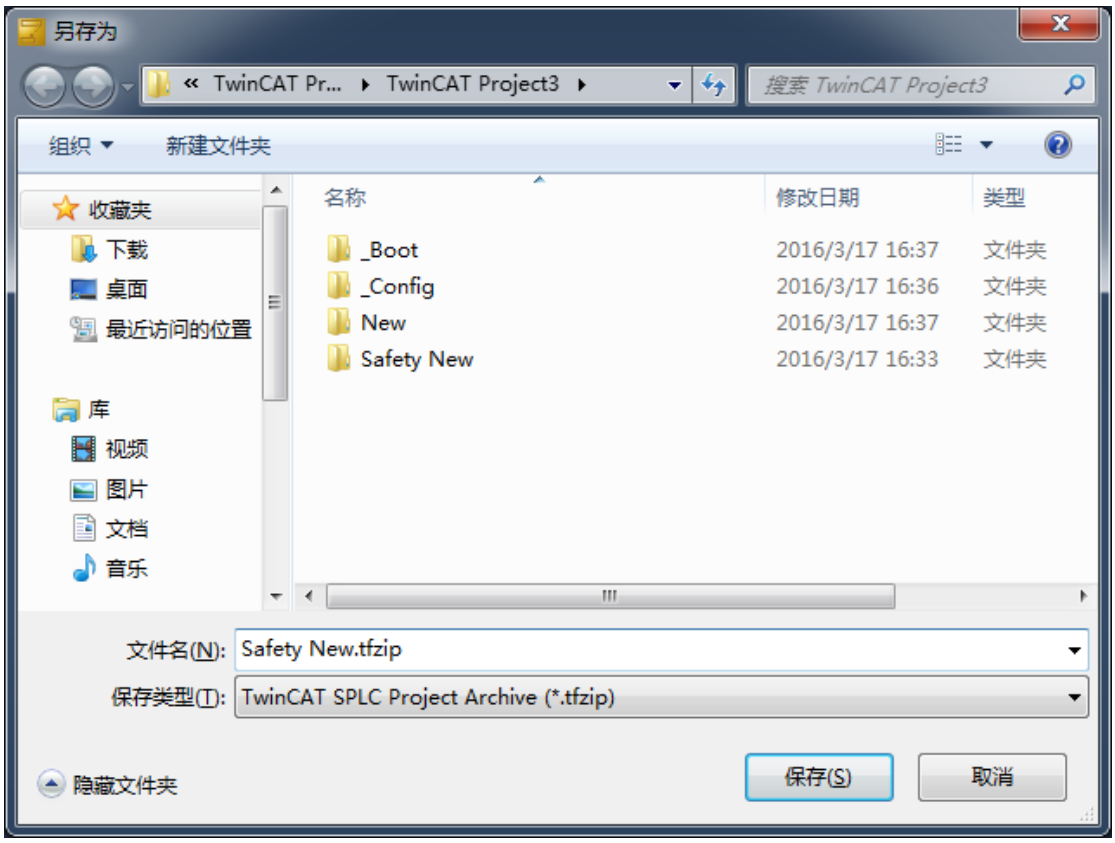


并且 Program、Function Block、Function、interface 等文件也可以保存成 ZIP 格式。导出和导入的方法同上。

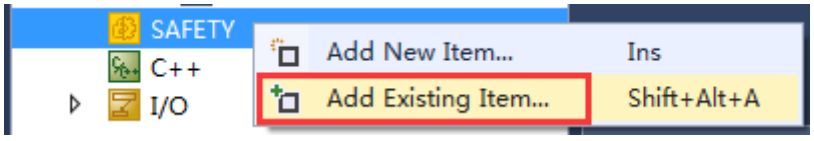
2.4 Safety 项目的保存与打开

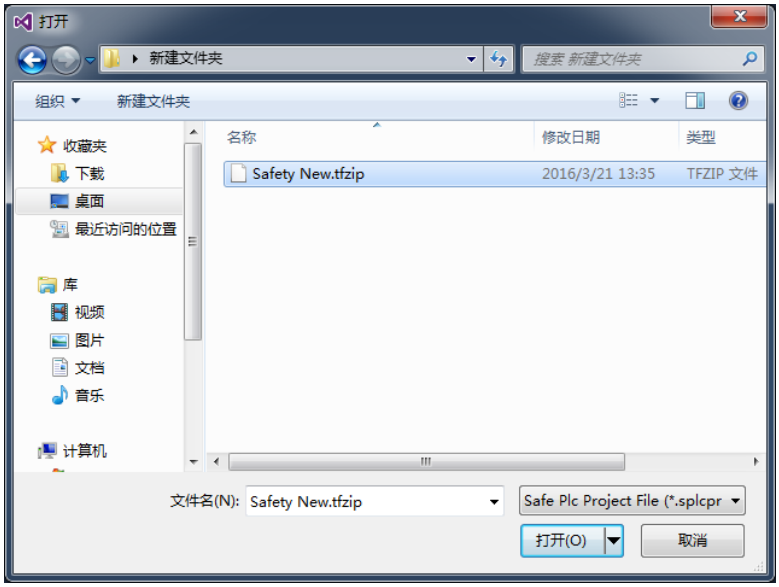
（1）保存 Safety 项目：右键 SAFETY 下的 Safety New，选择 Save Safety New as Archive...，在弹出的对话框中选择保存路径并将其另存为 tfzip 的格式，点击保存。





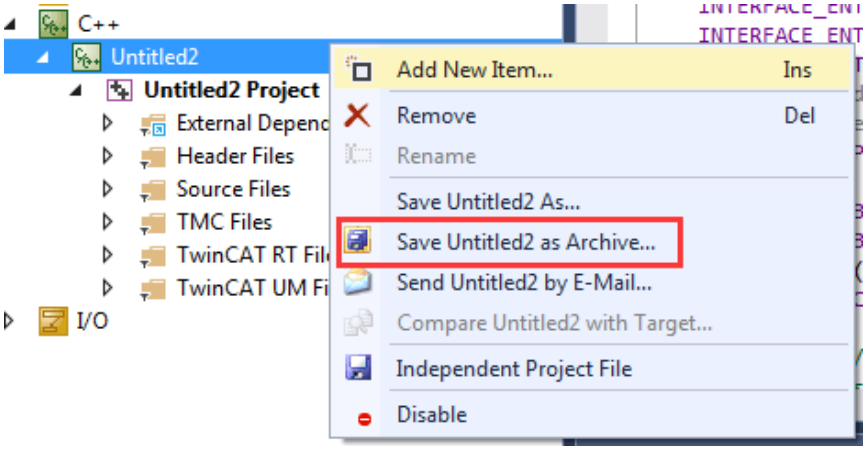
（2）打开 Safety 项目：右键 SAFETY，选择 Add Existing Item...，在弹出的对话框中选择保存好的文件，单击打开。

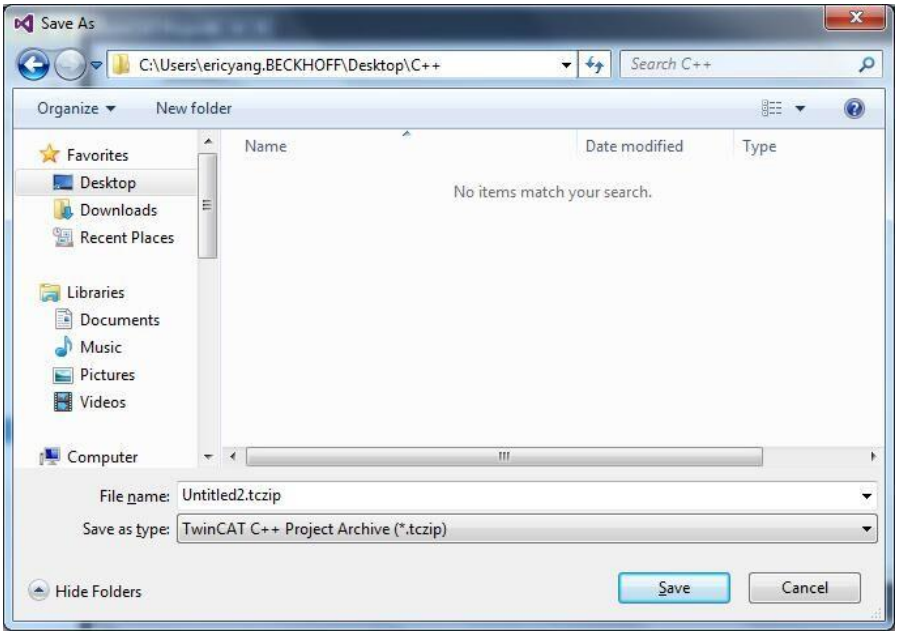




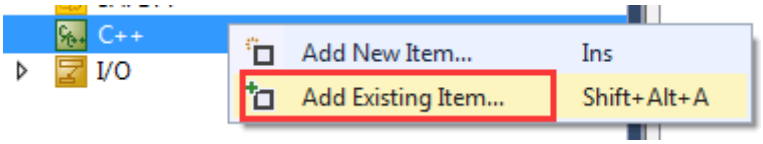
2.5 C++项目的保存与打开

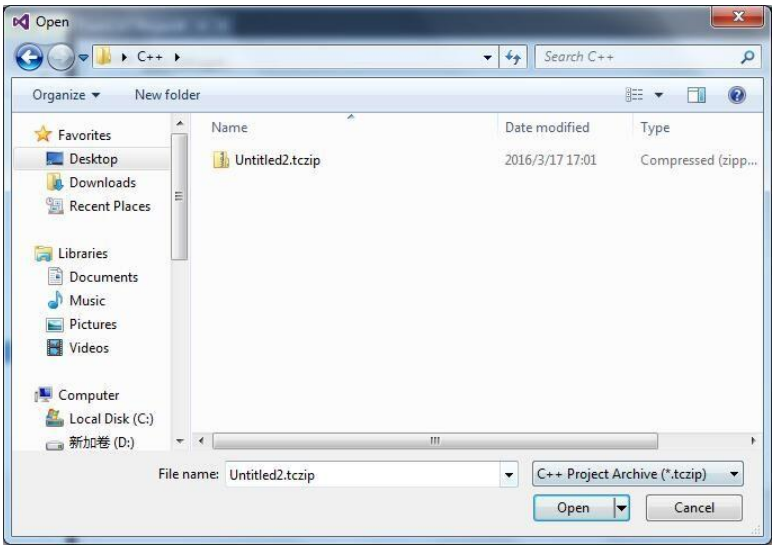
（1）保存 C++项目：右键 C++下的 Untitled2，选择 Save Untitled2 as Archive…，在弹出的对话框中选择保存路径并将其另存为 tczip 的格式，点击保存。





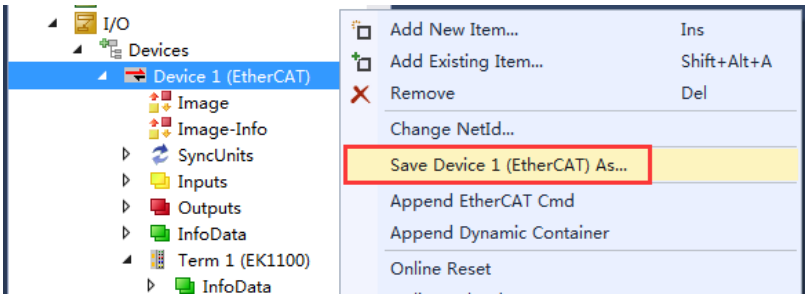
（2）打开 C++项目：右键 C++，选择 Add Existing Item…，在弹出的对话框中选择保存好的文件，单击打开。

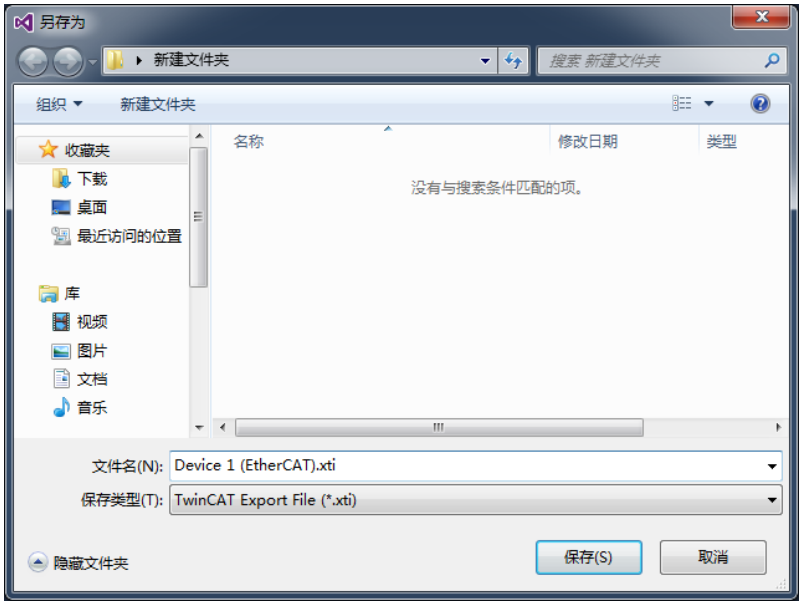




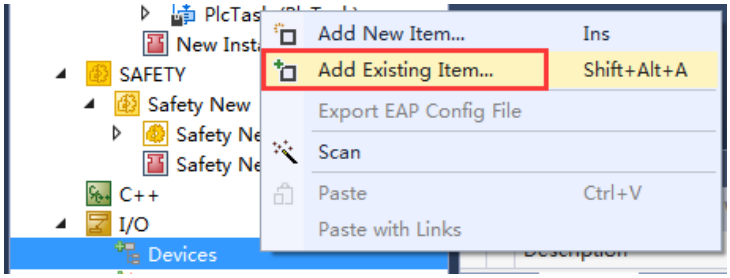
2.6 组态的保存与打开

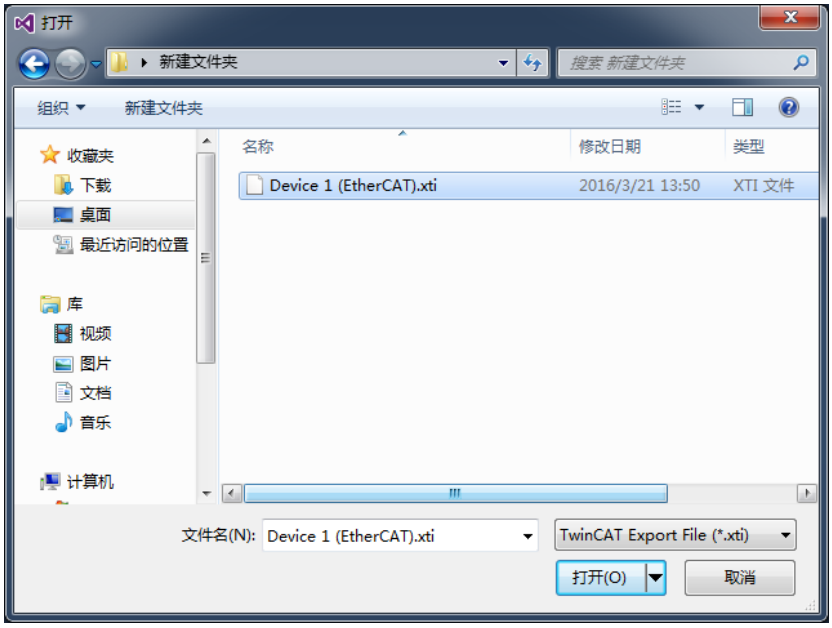
（1）保存组态：右键 Devices 下的 Device 1，选择 Save Device 1（EtherCAT） as…，在弹出的对话框中选择保存路径并将其另存为 xti 的格式，点击保存。





（2）打开组态：右键 Devices，选择 Add Existing Item…，在弹出的对话框中选择保存好的文件，单击打开。



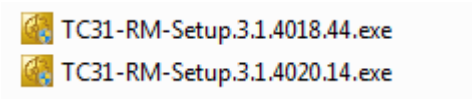


总结 TwinCAT3 中的几种文件保存格式：

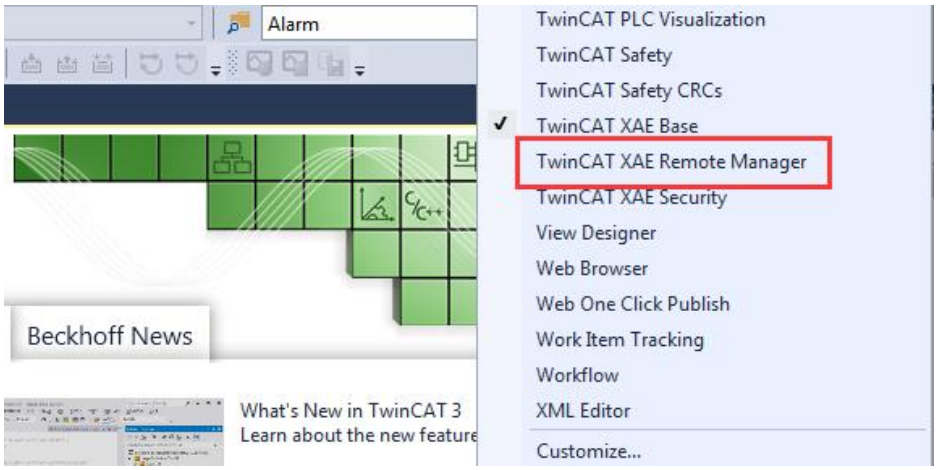
	保存类型	保存范围	保存内容
1	tpzip	PLC Project	自定义变量、Global List、所有 Program、Function Block（及其下所有 Action、Method、Property）、Function、Visulazation 及其他可在 PLC Project 下可建的文件类型
2	tfzip	SAFETY Project	包含 Group 以及其下 Devices 及逻辑程序
3	tczip	C++ Project	包含已添加的头文件的全部 C++文件
4	xml	Pro、FB、FUN、Interface	工程、函数、接口、功能块本身（接口和功能块下如果有 Method 和 Property，保存时也会包含）
5	zip	PLC Project	PLC 项目中所有文件夹、工程、函数、接口、功能块、HMI 等，导入 zip 必须选中文件夹完成导入
6	xti	IO 配置	网络及其下所有硬件模块、模块通道与 PLC 工程变量的绑定信息（导入时，需先导入 tpzip 再导入 IO 配置，变量绑定信息才会加载生效）
7	tszip	整个 Project	轴配置、PLC 工程、SAFETY 工程、C++工程和 IO 配置
8	tnzip	整个 Solution	打包多个 Project（TwinCAT、HMI、Measurement 等）

3. TwinCAT3 版本切换

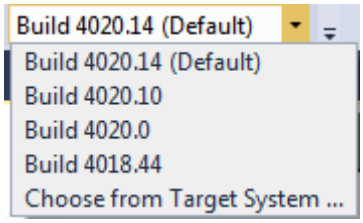
TwinCAT3 自从 4020 版本开始增加了 Engineering Remote Manager 功能，解决了不同版本 TwinCAT3 的切换，因此从 4020 版本开始每次 TC3 版本更新都会附带发布一个 RM 版本的安装包，当然在 4020 之前也发布了一个 RM 版本，TC3.1.4018.44。因此从 4020 版本开始单独安装一个 RM 版本就可以集成这一个版本的内容，方便大家做版本切换（如果用 FULL 版本的 TC3 覆盖安装老版本也会保留老版本内容方便做切换），这样就可以方便对不同 TC3_runtime 版本控制器进行编程，避免因为开发电脑中的版本的 TC3 与控制器中 runtime 的版本不兼容，从而报错的情况。
目前在官网上有以下两个 RM 版本可供下载单独安装，随着版本提示会有越来越多的 RM 版本发布：



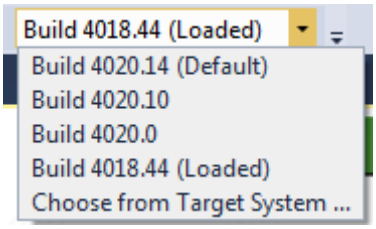
（1）比如我电脑中已经集成了多个版本，如果希望完成版本切换，首先打开 VS，右键工具栏空白部分，勾选 TwinCAT XAE Remote Manager



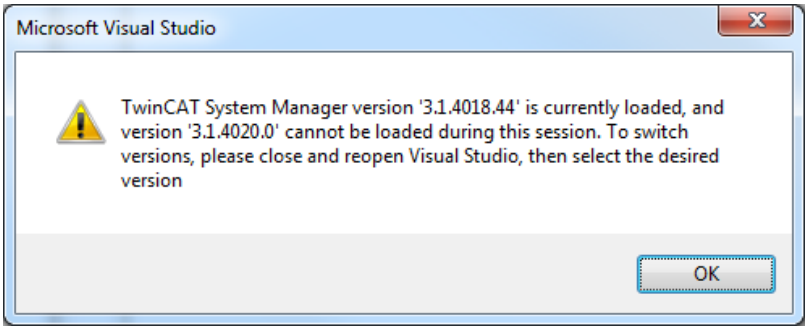
（2）此时在在工具栏中会出现 TwinCAT XAE Remote Manager Toolbar 选型可供多版本切换，如果电脑中有多个版本实现被安装，比如下图中我有 4 个版本的 TC3（有的是通过安装 RM 版本，有的通过 FULL 覆盖老版本）被保留了下来。



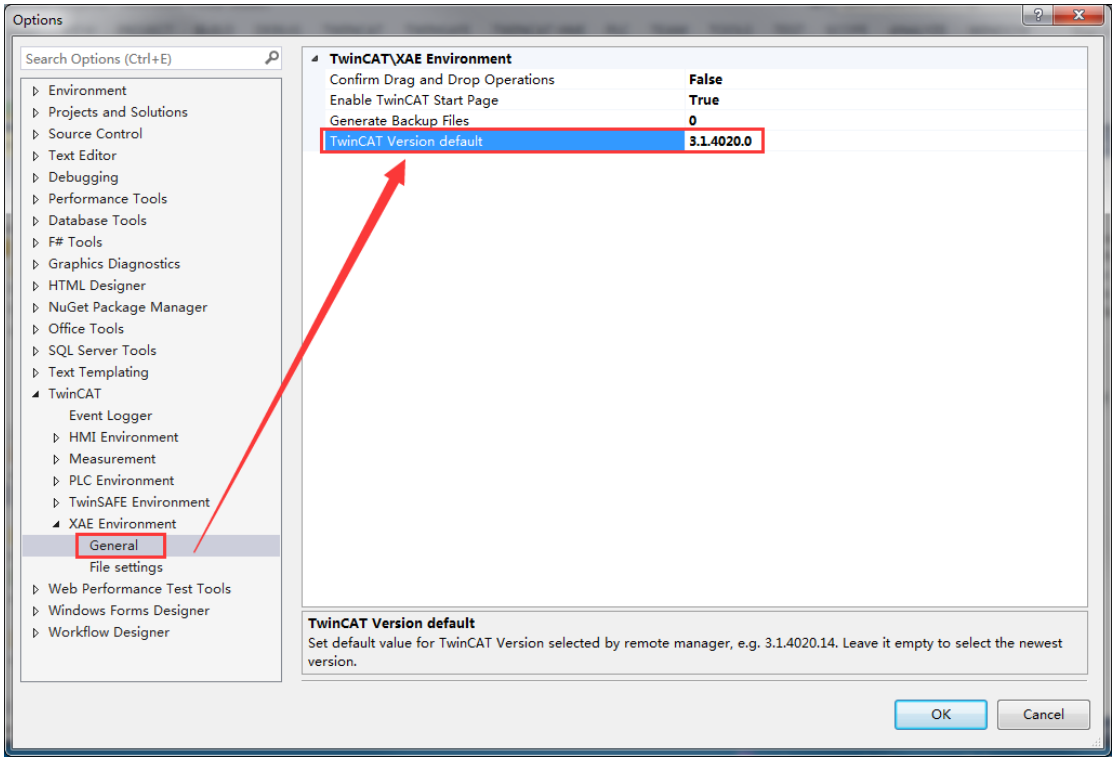
（3）假设目标控制器现在是 TC3.1.4018.44 版本的 Runtime，我们就可以在这里选择 Build 4018.44，就很容易把当前 VS 版本切换到了一个老版本



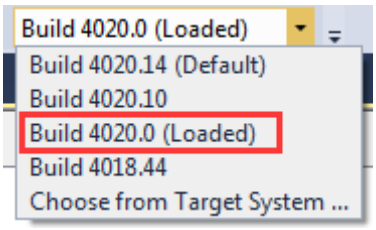
（4）不过每次版本的调整只允许改动一次，如果还想再切换到另一个版本，那需要重新关闭 VS，再次打开才可以



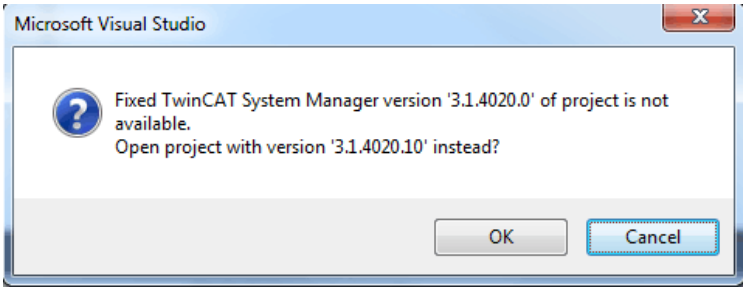
（5）因为 TC3 版本很多，客户可以根据自身情况固定觉得功能相对稳定的版本进行开发，只需要在 Options→TwinCAT→XAE Environment→General 中设置 TwinCAT Version default 为一个固定版本即可，如下图设置为了 3.1.4020.28。这样用户每次新建 TwinCAT3 项目，都以此固定版本基础完成开发。



（6）一旦版本固定了，每次打开这个项目，具有 RM 功能的 TC3 变成电脑都会自动切换到与之相匹配的版本



（7）当然如果电脑中打开一个很早版本的 TC3 项目，没有与之匹配的版本会提示建议你切换到当前你电脑的最新版本

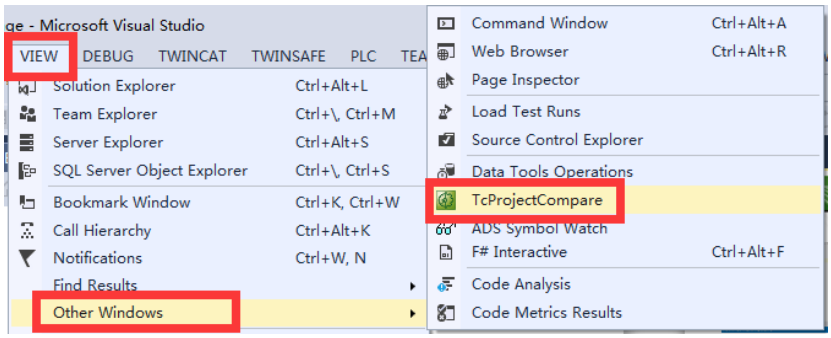


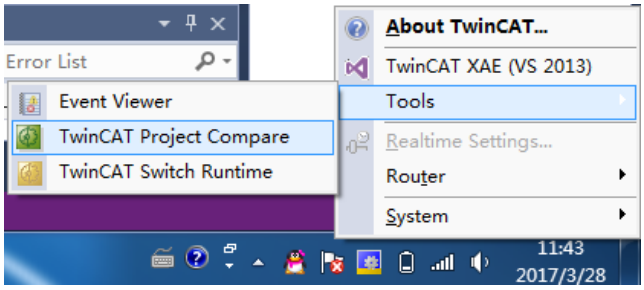
（8）不同 VS Shell 所支持的 TwinCAT3 RM 版本，如下列表：

TwinCAT 3 version	Supported VS Shells
TwinCAT 3.1.4016	VS 2010 - VS2013
TwinCAT 3.1.4018	VS 2010 - VS2013
TwinCAT 3.1 4020	VS 2010 - VS2015

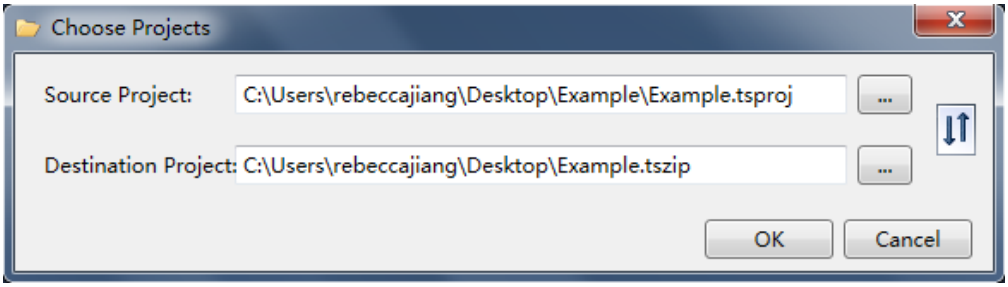
4. TwinCAT3 项目比较

（1）此功能用于项目比较，打开方式有两种，可以在 TwinCAT3 软件里选择 View->Other Windows->TcProjectCompare，或者直接在任务栏的小图标上选择 Tools->TwinCAT Project Compare

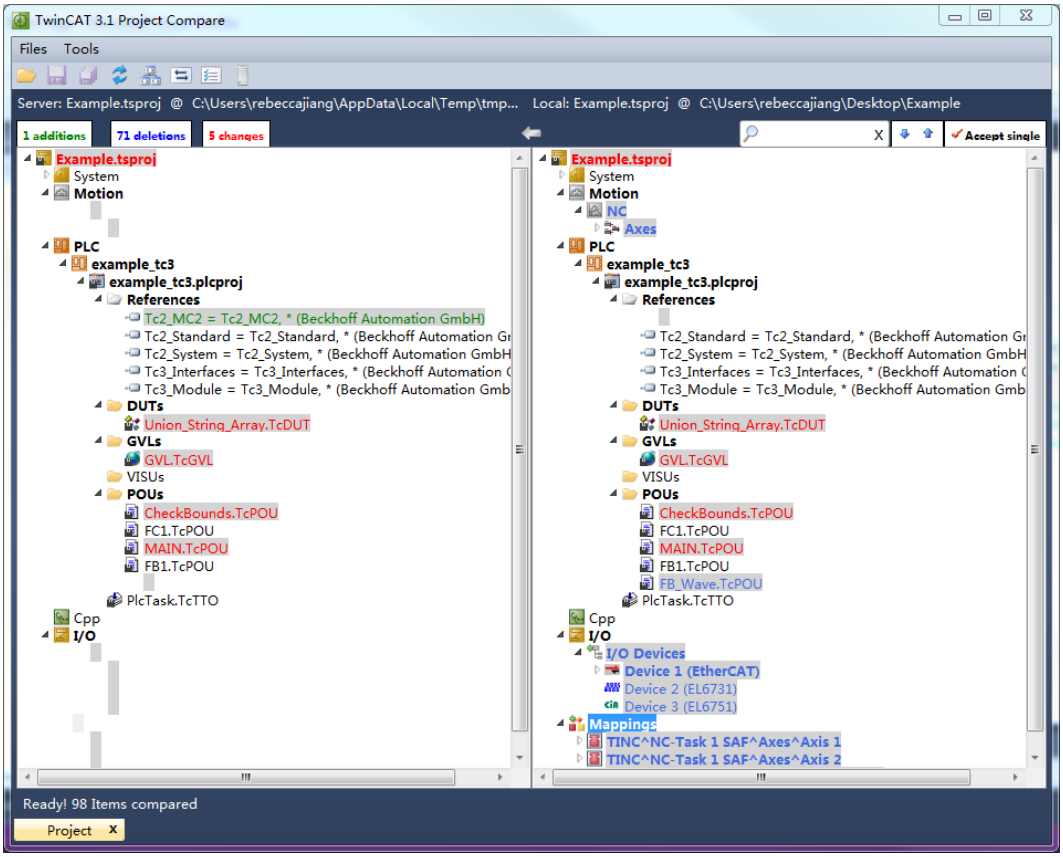




（2）选择需要比对的工程，文件格式不必一致，可以是 tszip 或者 tsproj。

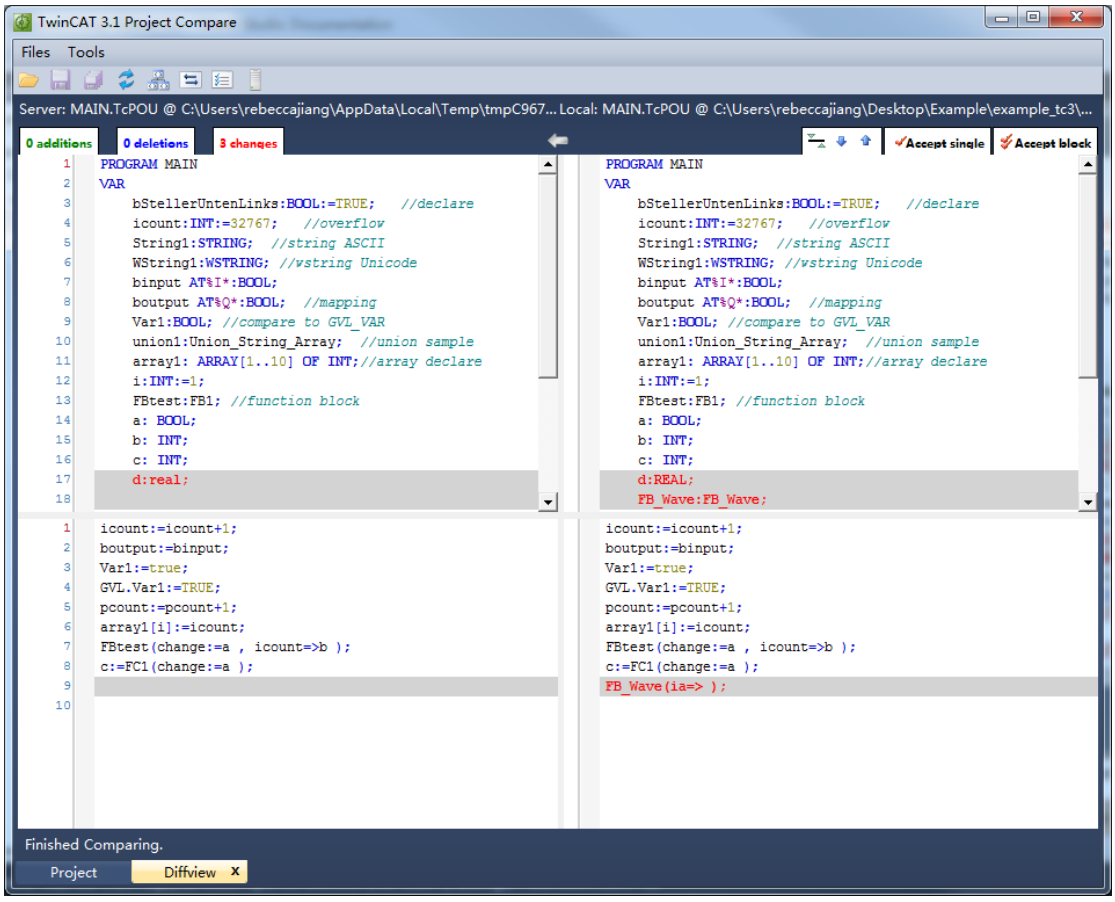


打开以后可以看见比较的详细信息。右边工程是原始工程，左边是被比较工程

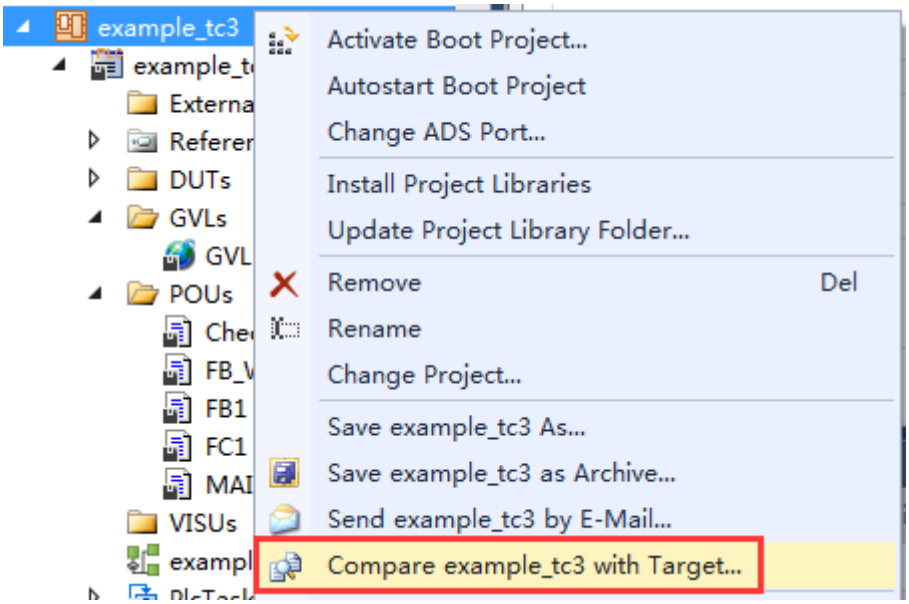


可以简单的根据颜色看出一些不同。绿色表示新增项，红色表示同一个程序功能块或者 Global List 中有不同之处，蓝色表示被删除项。

而针对红色不同的部分，可以双击打开，单独查看代码或变量申明等部分发生了哪些具体的改变。

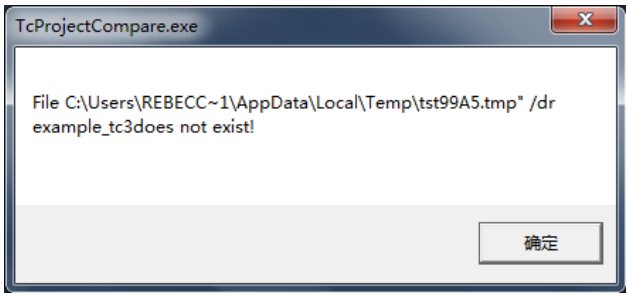


（3）TwinCAT3 还支持本机和控制器的比对，例如控制器上已有程序，同时本机也打开一个程序，在工程上右键选择 Compare with Target…就可以进行比对了。

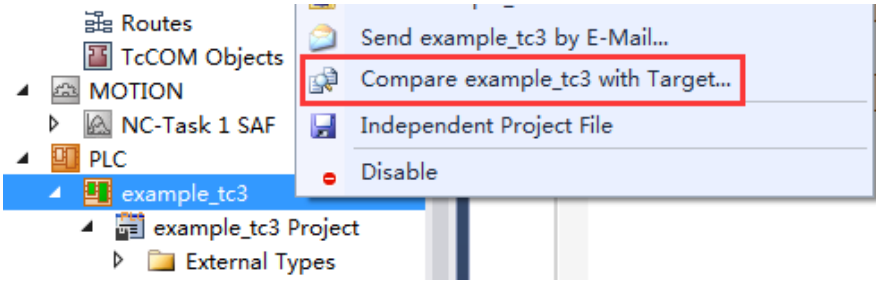


如果两个工程完全一致，则会报弹窗：The two Projects are equal.

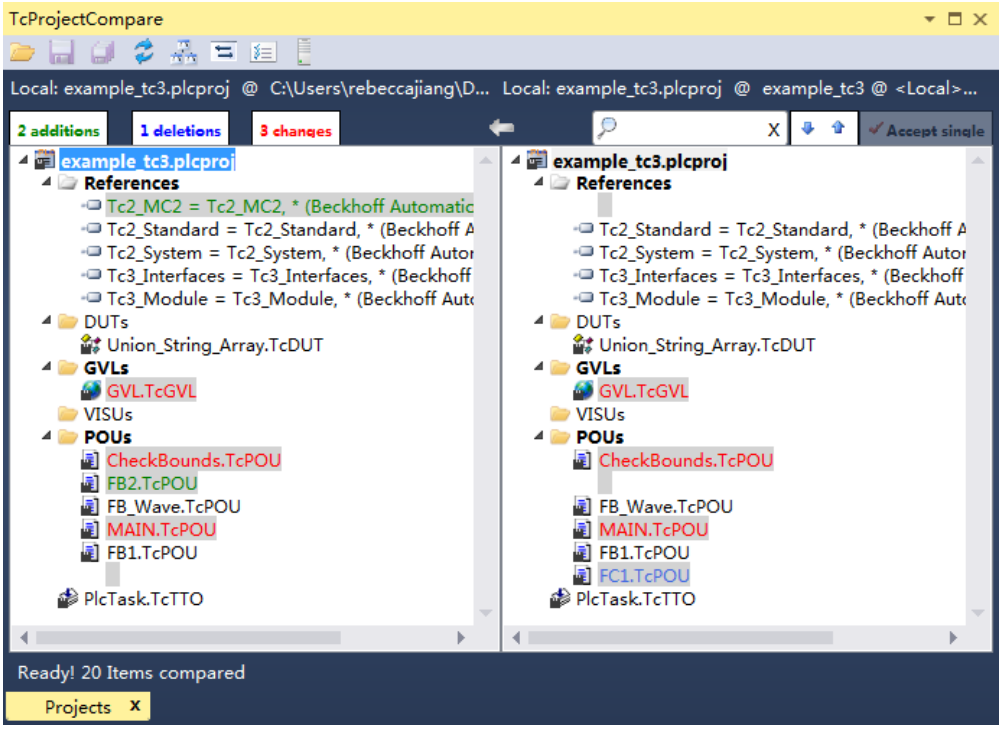
若没有对控制器下载源代码时作比对，会出现如下弹窗。



（4）此外，内嵌的比较功能可以针对本地和目标控制器中的 PLC Project 等进行比较，选中 PLC 下的工程右键可选 Compare with Target...



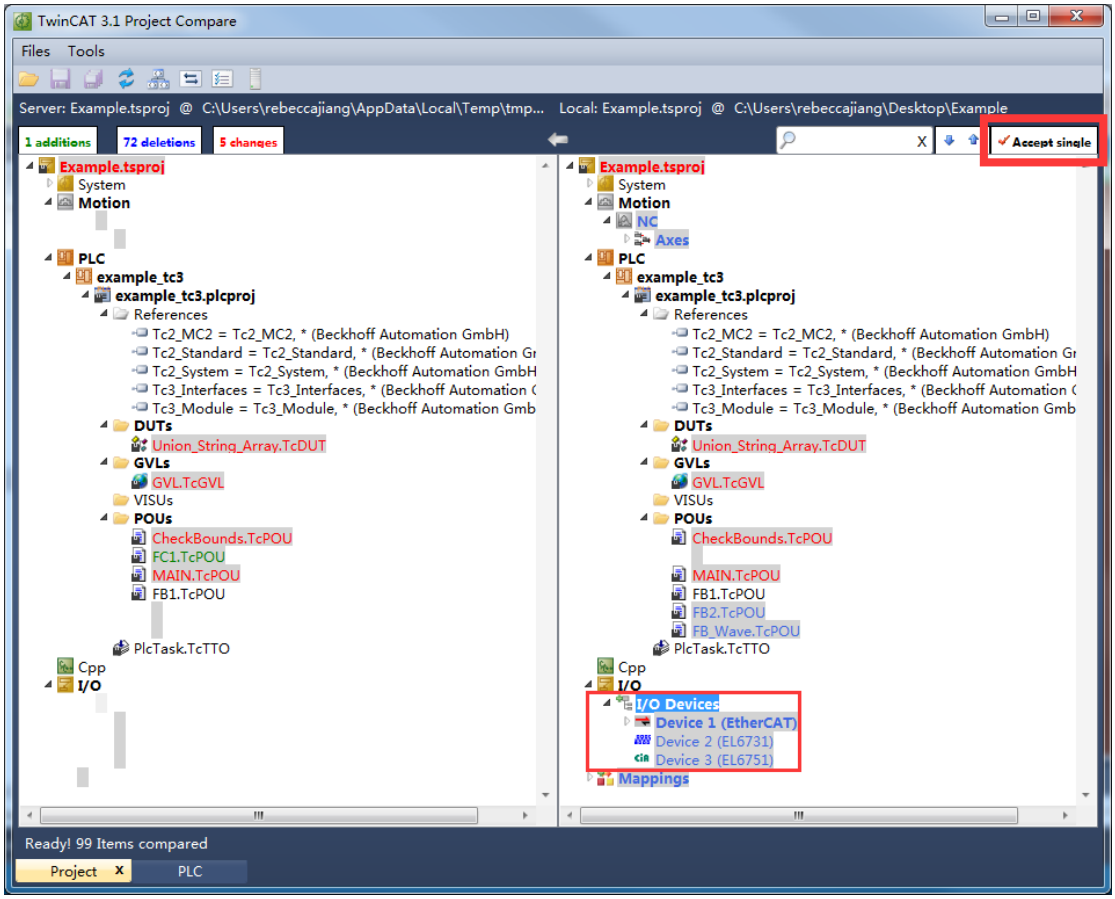
如同之前所说，也会出现新增项、删除项、改变项的提示，并可以查看细则。



（5）Merge 合并功能：通过此功能，可将比较文件的内容合并到一个工程中去。

合并方向是固定的，两个工程中间的小箭头  就是合并方向。

此处选中 I/O Devices 下的设备，点击右上角的 Accept single 就可以把右边的项目添加到左边去。



修改后结果如下。

所以如果希望修改合并方向，只要保存修改后点击工具栏的 Switch Files 就可以交换源文件和比较文件。

十、 TwinCAT2 项目到 TwinCAT3 转换

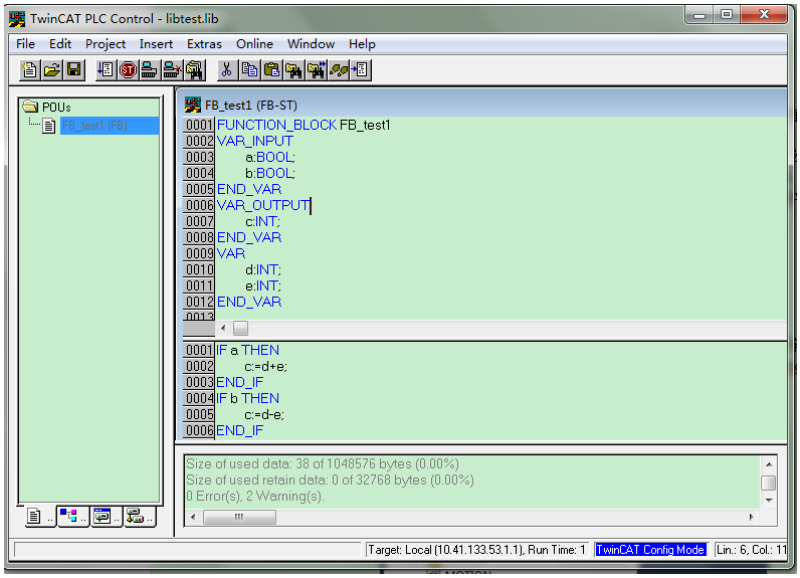
TC3 可以支持的转换 TC2 项目的文件有四种*.tsm *.tpy *.lib *.pro 也就是只有这四种文件可以转换到 TC3 项目中。那么如何进行转换呢。

进行转换项目的第一步是将项目的库文件进行转换，主要包含 TC2 自带的库以及客户自编的库，如果是 TC2 自带的库，那在 TC3 中已经包含了所有 TC2 的库无需客户转换，但如果是客户自编的库，就需要每个客户自己进行库转换，因此 TC2 项目转换 TC3 并不复杂，只需要把库转换好就可以了。

1. 库文件转换

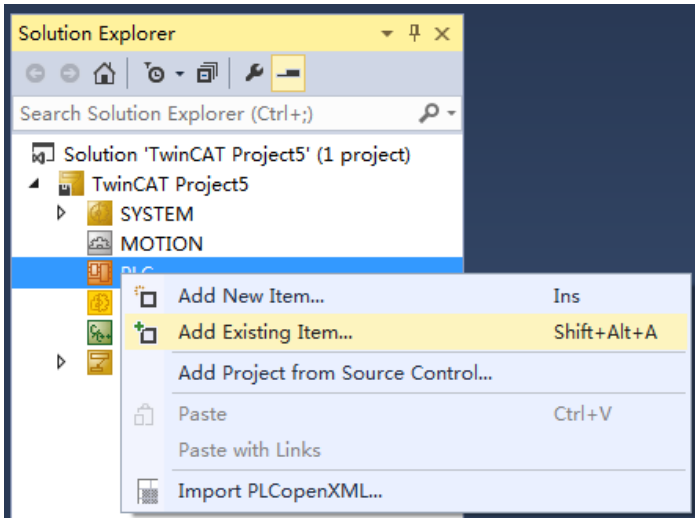
首先是将用户自定义的库文件进行转换，转换方法如下例所示：

（1） 将用户自定义的库文件在 TwinCAT2 中打开，编译。

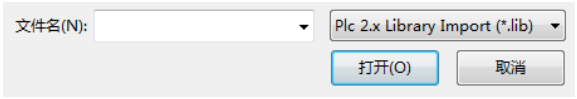


库文件需要在 TwinCAT2 中能够打开并且编译没有错误后才可继续下面的步骤。

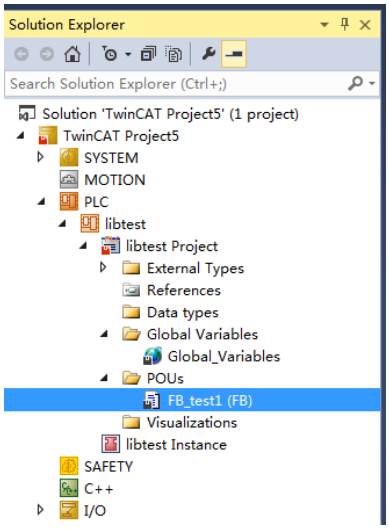
（2） 打开 TwinCAT3，新建一个 TwinCAT Project，在 PLC 上右击选择 Add Existing Item...。



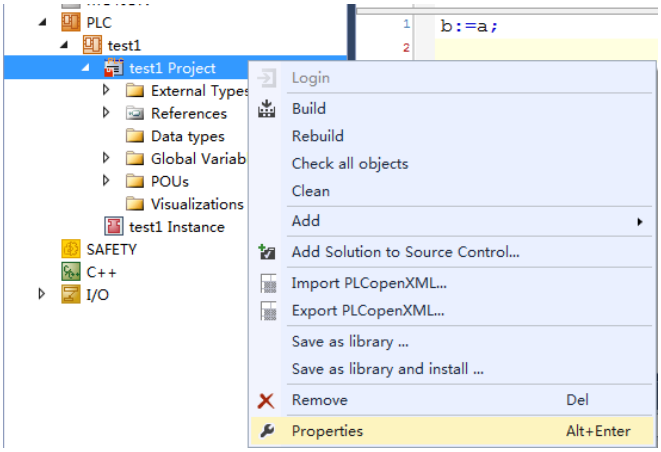
（3） 弹出文件选择窗口，选择查找 Plc 2.x Library Import（*.lib）类型的文件。



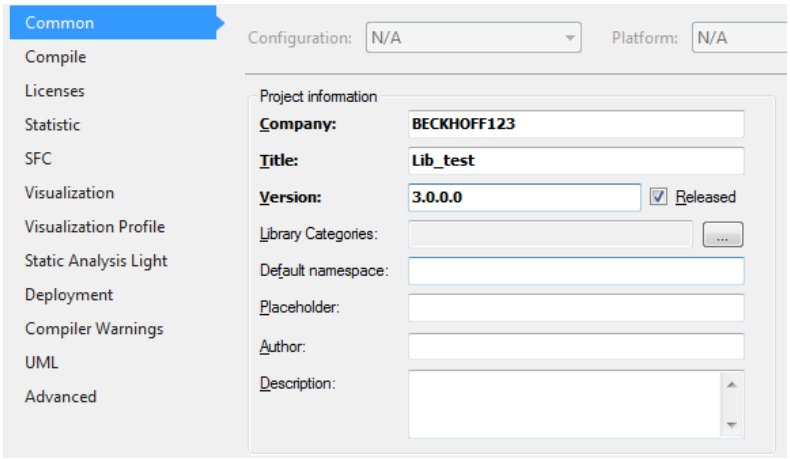
（4） 找到用户自编的库文件，点击打开。就完成了用户自定义库文件的加载。



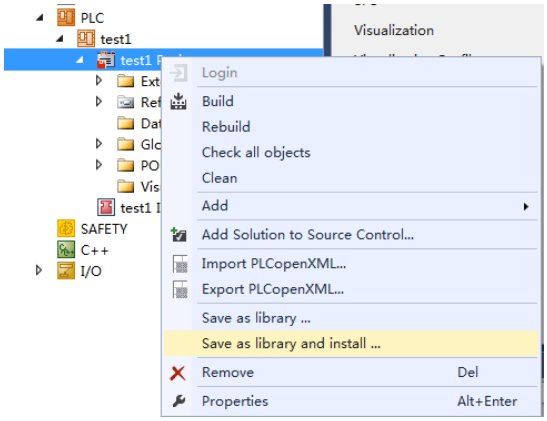
（5） 将转换好的库文件安装到 TwinCAT3 库中。在库文件项目上右击，选择 Properties。



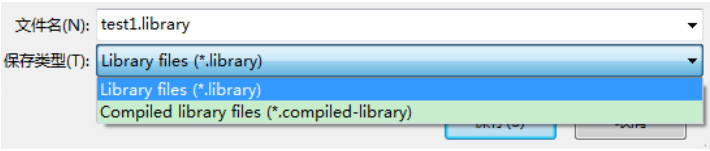
（6） 弹出的窗口中，粗体内容是必填项目，在粗体内容下面有 Default Namespace 和 Placeholder 两个选项推荐用户填写方便以后库文件的调用，如果不填那么与 Titlle 一致。



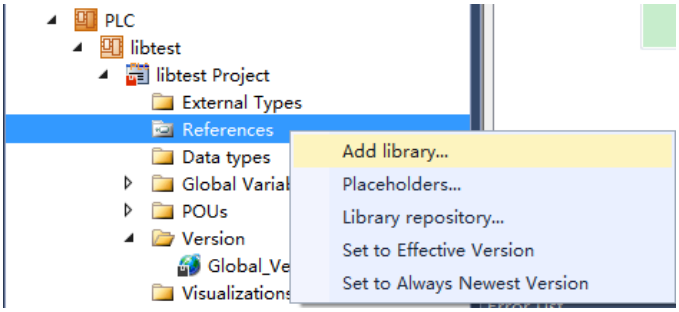
（7） 填写完成后，在加载的项目上右击选择 Save as library and install...，意思是保存库文件并安装到系统库中，而 Save as library...只是保存库文件。



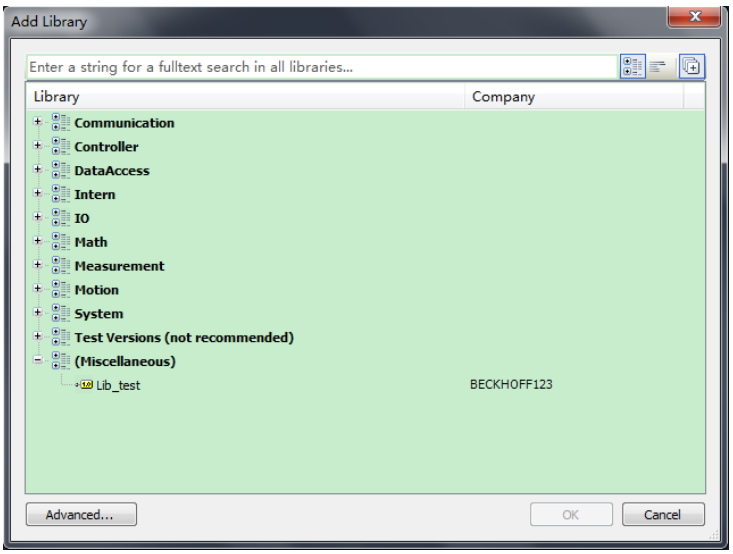
（8） 选择后弹出，保存窗口，在保存类型中*.library，*.compiled-library。保存成*.library 的格式后，在加载库文件后可以打开文件的程序代码，而*.compiled-library 是打不开程序代码的。



（9） 在项目的 References 右建选择 Add Library...。



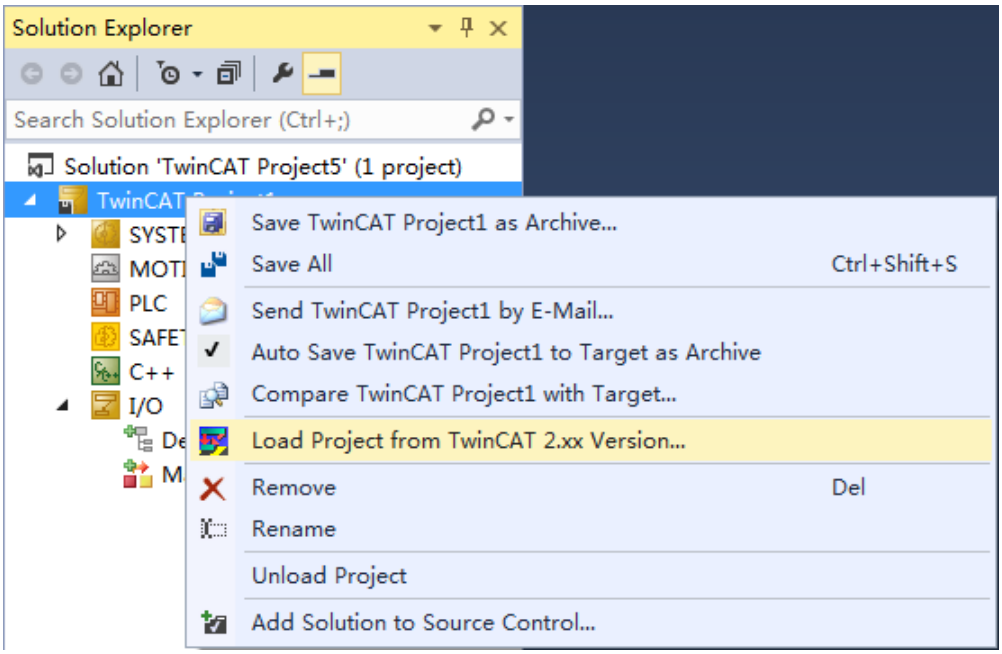
（10） 在 Add library..中的 Miscellaneous 中可以看到，用户自定义的库文件已经安装好了。



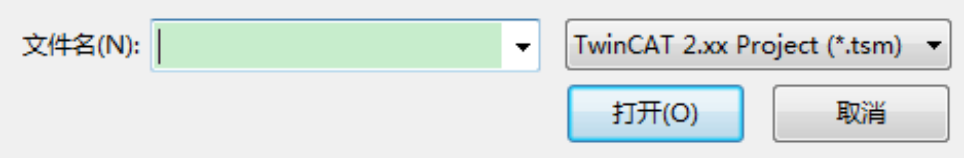
2. 项目转换

在所有项目中的用户自定义库文件转换完成并且安装在库中，便可以进行项目的转换。首先把项目用到的库文件放到自定义文件夹中，本例中放在桌面→Lib 中。

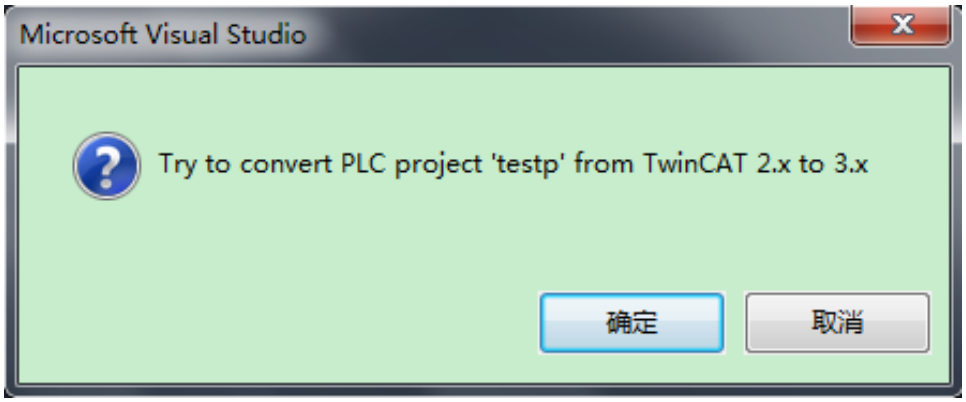
（1） 打开 TwinCAT3，新建 TwinCAT 项目，在 TwinCAT 项目上右击，选择 Load Project from TwinCAT 2.xx Version..。



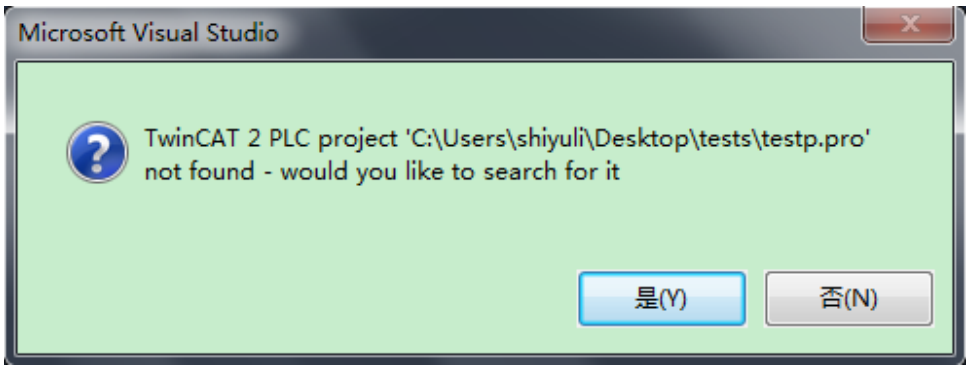
（2） 弹出文件选择对话框，找到需要转换的*.tsm 文件选择打开。



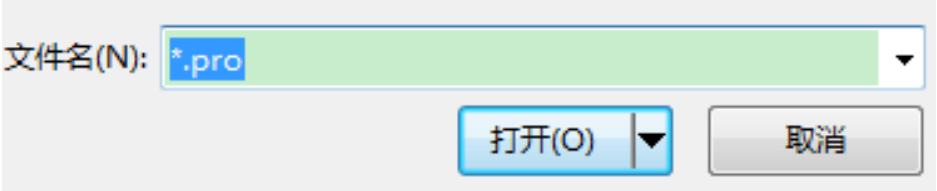
（3） 弹出的对话框，提示是否要进行转换，选择是（Y）。



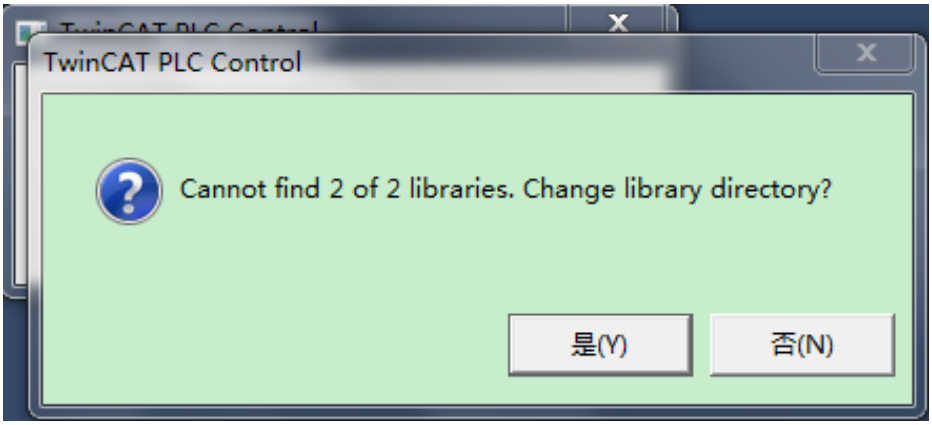
（4） 选择是（Y）



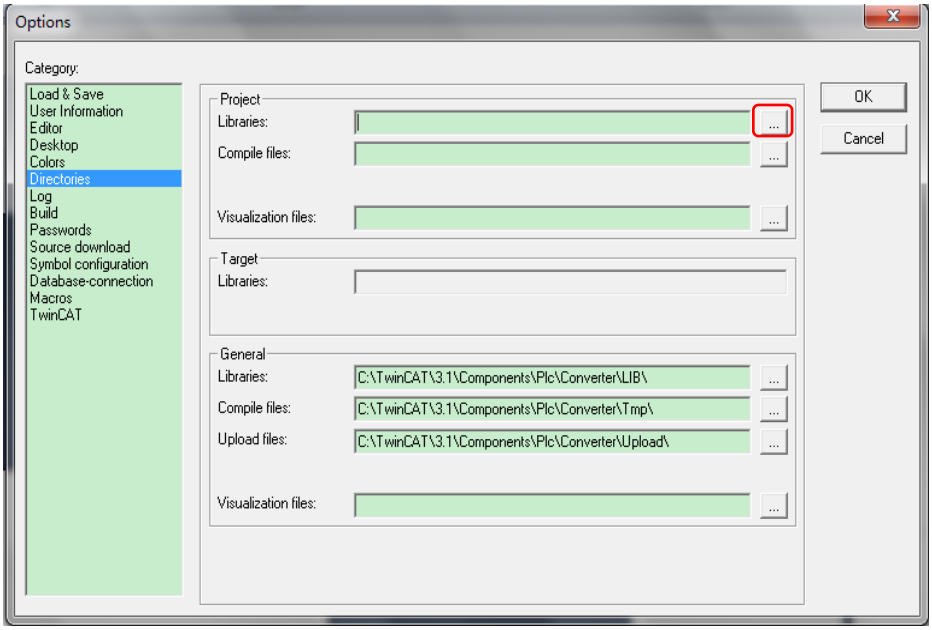
（5） 选择是（Y），弹出查找*.pro 文件。



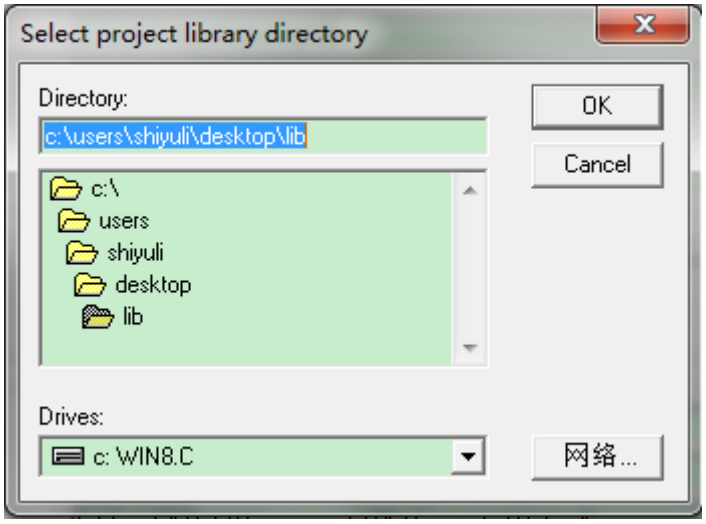
（6） 添加*.pro 文件后，弹出如下对话框，选择是（Y）。



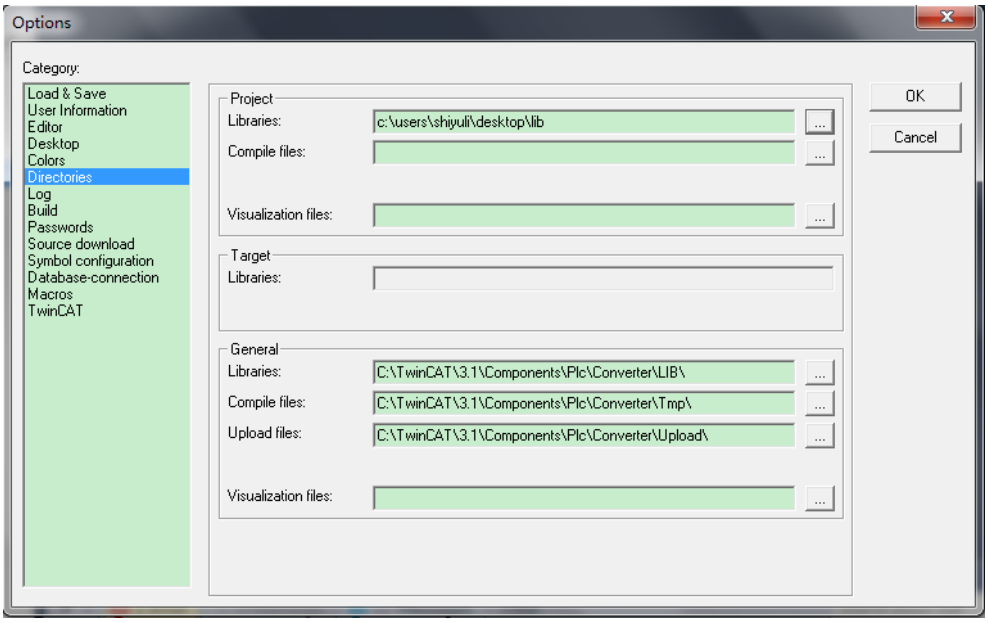
（7） 在 Libraries 选项卡中点击路径选择按钮。



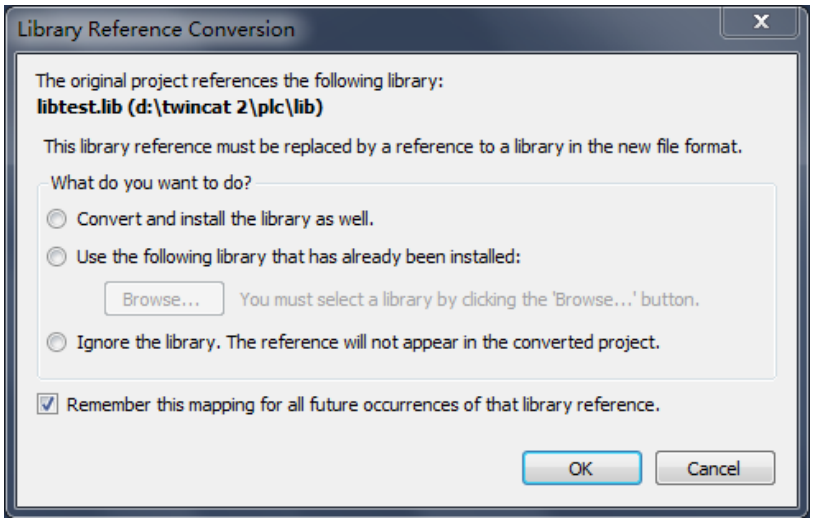
（8） 选择库文件所在文件夹，点击 OK。



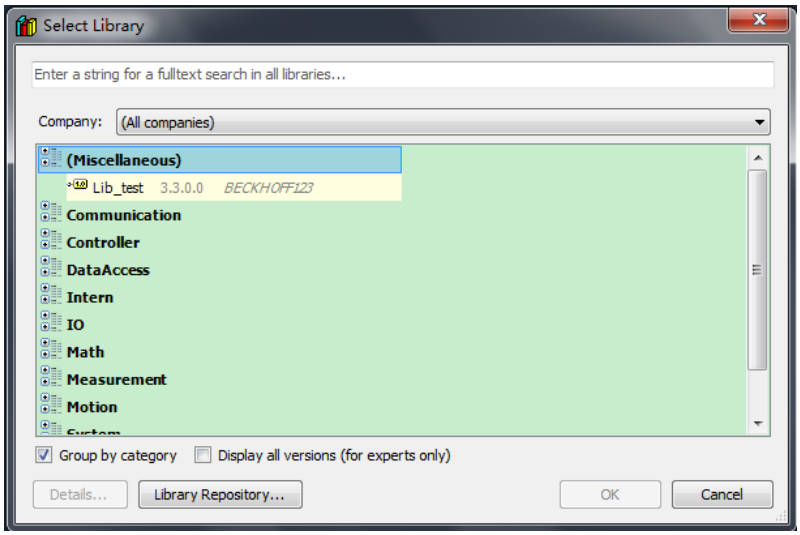
（9） 点击 OK。



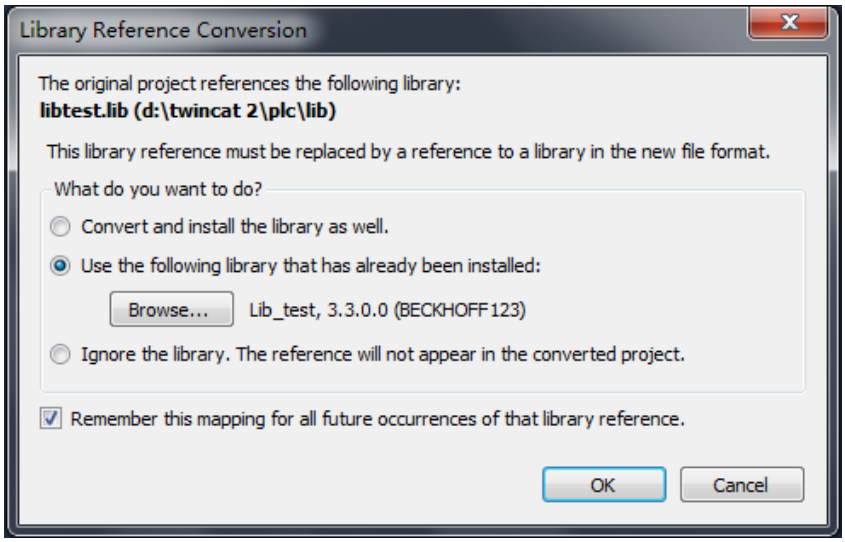
（10） 弹出库文件转换对话框。



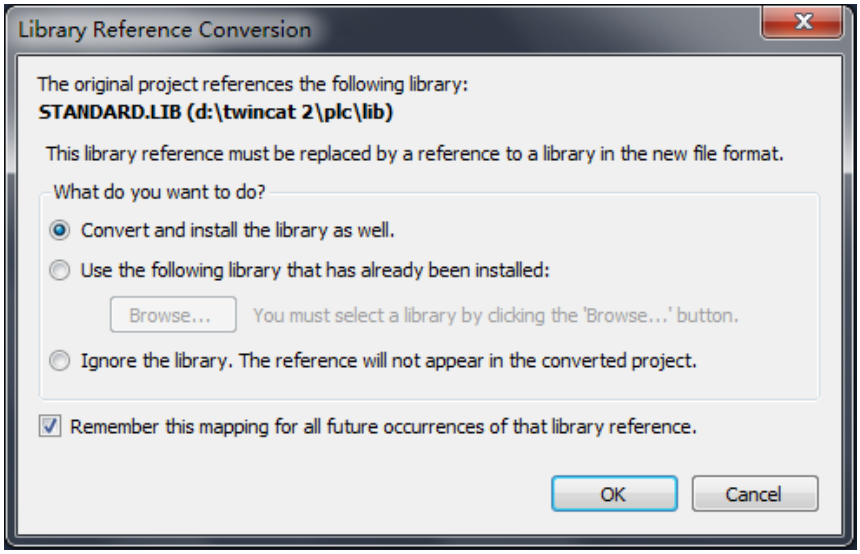
（11） 这个库文件是我们自定义的库，选择 Use the following library that has already been installed: 点击 Browse…。



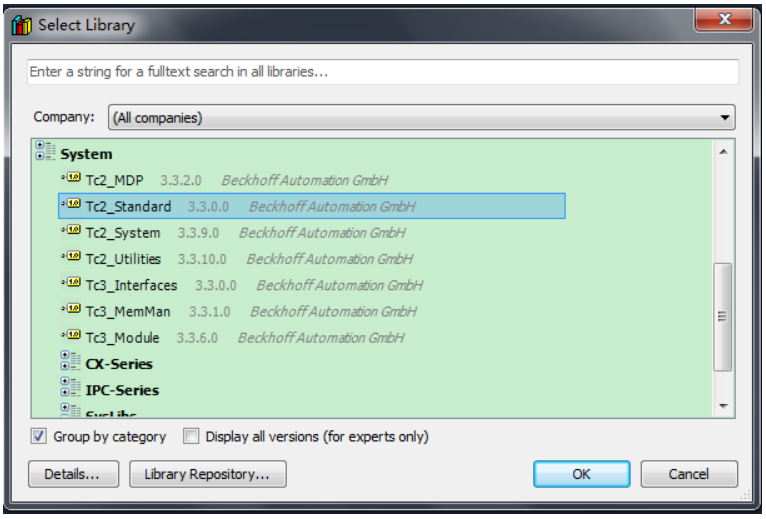
（12） 双击（Miscellaneous）中的自定义库。



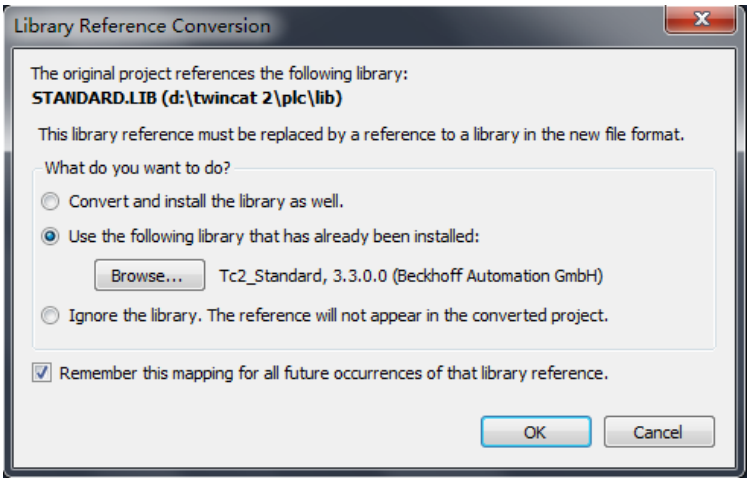
（13） 选择 OK



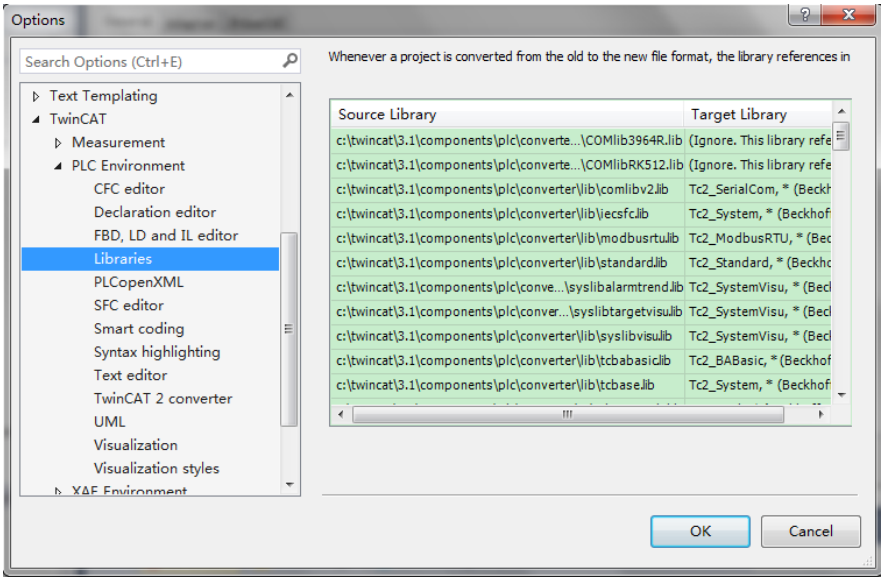
（14） 此库文件是 TwinCAT2 的系统库文件，同样手动进行库文件的转换，选择 Use the following library that has already been installed: 点击 Browse…。



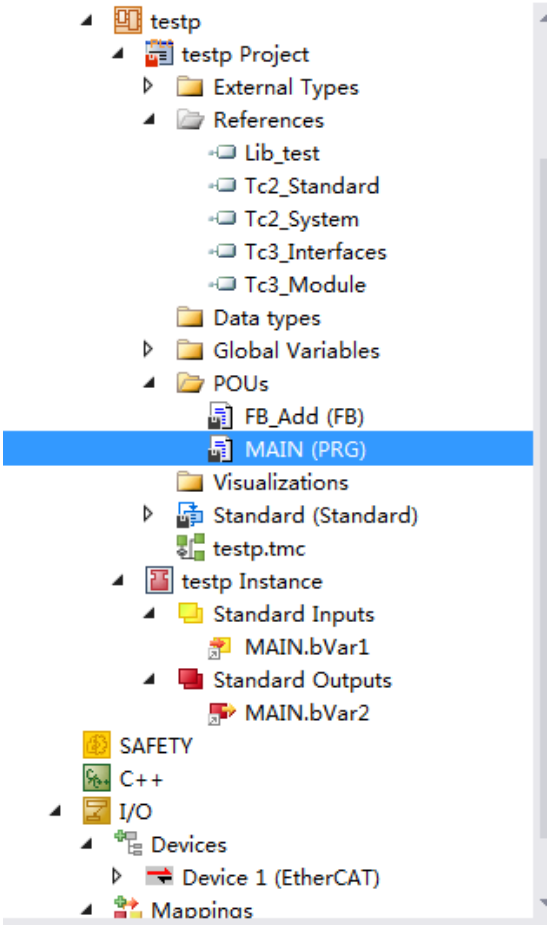
（15） 选择 System 下的 TC2_Standard 库。



TwinCAT2 与 TwinCAT3 库文件的对应关系可以参考，VS 的 TOOL 菜单中的 Options→TwinCAT→PLC Environment→Libraries。



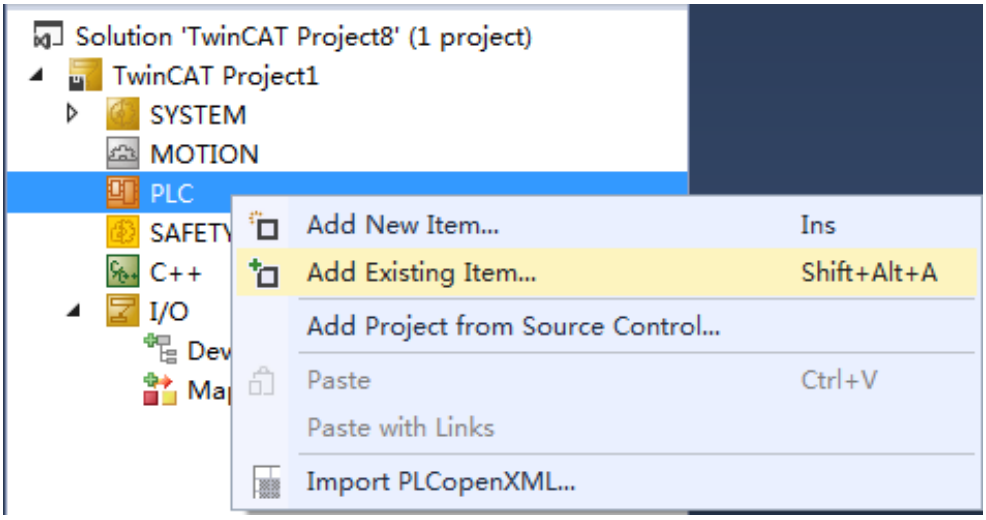
（16） 库文件转换完成后，可以看到项目转换完成，并且我们的变量链接也是在的。



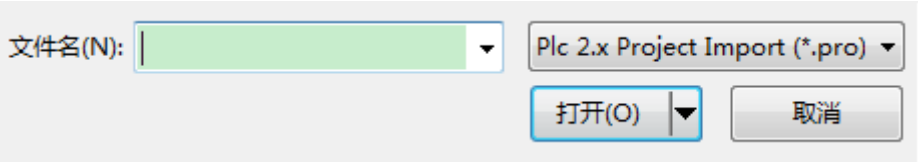
3. 单独转换程序文件

如果只转换程序文件，即只转换*.pro 文件，方法如下：

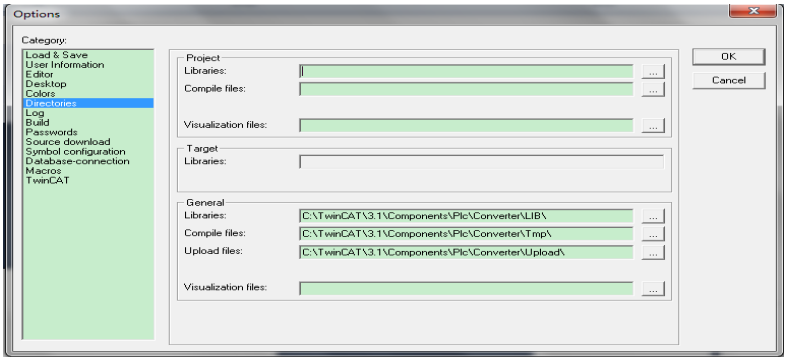
（1） 新建项目后，在 PLC 上右击，选择 Add Existing Item...



（2） 选择*.pro 类型的文件，找到需要转换的项目*.pro 文件。

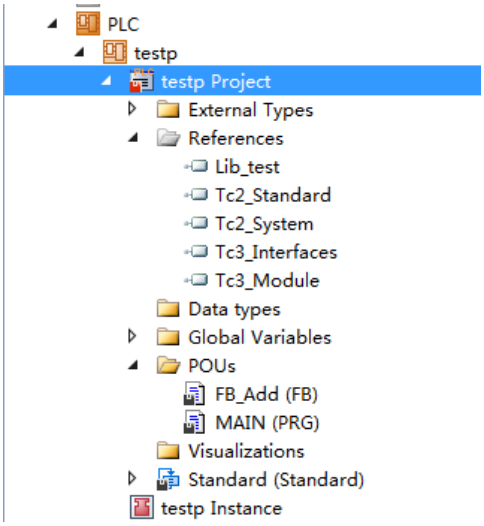


（3） 打开后弹出库文件的联接窗口。选择是（Y）



（4） 然后重复 2 中的（7）-（15）步。

（5） 转换完成。



注意:如果在转换完成后，可能在编译时还会有错误报出。这是因为 TwinCAT2 和 TwinCAT3 在对变量声明，赋值等等一些操作上做了小的改动，例如:

TwinCAT2 中数组赋初值不需要加中括号，（arr1：ARRAY[1..5] OF INT:=1,2,3,4,5;）。

TwinCAT3 中数组赋初值需要加中括号，（arr1：ARRAY[1..5] OF INT:=[1,2,3,4,5];）。

TwinCAT2 中 REAL 与 INT 数据类型转换使用 TRUNC。

TwinCAT3 中 REAL 与 INT 数据类型转换使用 TRUNC_INT。

那么在编译出现错误时，请根据错误信息，在 Information System 中查找关键词。例如:在 Information System 中查找 TRUNC 会有如下说明

Note	In TwinCAT 2.x PLC Control the TRUNC operator converts from REAL to INT. For this reason when importing a TwinCAT 2.x PLC project TRUNC automatically will be replaced by TRUNC_INT.

然后根据 Information System 给出的信息进行修改即可。

十一、 EtherCAT 性能介绍及诊断

1. EtherCAT 性能介绍及诊断

（1） EtherCAT 性能介绍

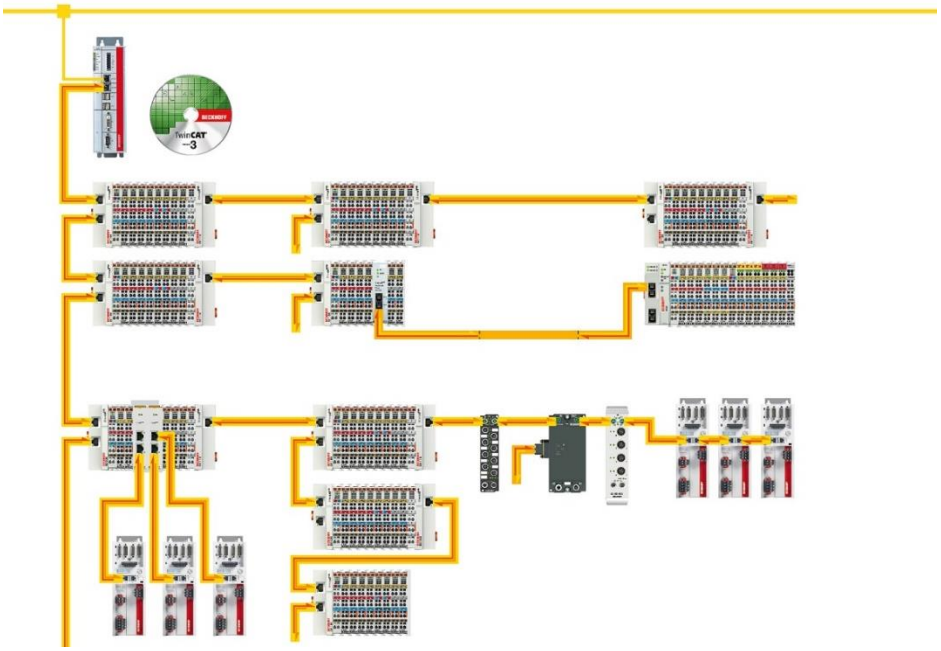
EtherCAT 是由 Beckhoff 自动化公司于 2003 年提出的实时工业以太网技术。它也是一种基于以太网的实时工业现场总线通讯协议和国际标准。具有高速和高数据的特点，支持多种设备连接拓扑结构。其从站节点使用专用的控制芯片，主站使用标准的以太网控制器。

EtherCAT 的性能具有以下特点：

- 卓越的网络性能。对 1500 台设备，吞吐量可达到 10kb/ms。1000 个分布式 I/O 数据的刷新周期仅为 30μs，与 100 个伺服轴的通讯只需 100μs。分布式时钟技术保证了这些轴之间的同步时间偏差小于 1μs。一家德国的现场总线技术专家经测试得到以下数据：（如下图所示），可以看出 EtherCAT 的性能表现非常优秀。

堆栈时间	Profinet IO	Ethernet/IP	EtherCAT
平均值	0.58 ms	1.89 ms	0.11 ms
最大值:	0.74 ms	2.96 ms	0.18 ms
最小值:	0.54 ms	1.23 ms	0.05 ms

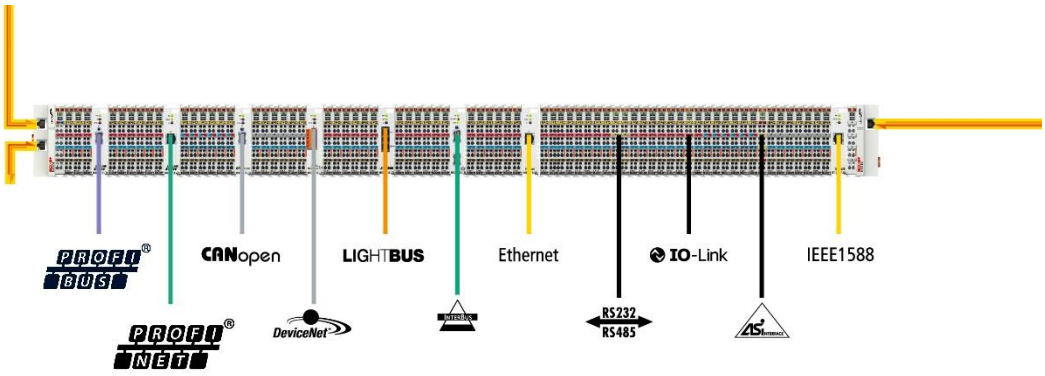
- 灵活的拓扑结构。EtherCAT 支持几乎所有拓扑结构，例如总线形、树形或星型，因此，源于现场总线的总线形结构也可用于以太网。将总线和分支结构相结合特别有助于系统布线。所有接口都位于耦合器上，无需使用附加交换机。
- 布线。所有接口都位于耦合器上，无需使用附加交换机。



- 可采取不同的传输电缆，最大限度地发挥布线的灵活性。用以下表格来反映在不同的通讯传输方式下，且能保证通讯数据传输正常，两台设备可间隔的距离。

传输方式	距离
工业以太网线（100BASE-TX）	最远 100m
E-bus	最远 10m
光纤	50-20000m

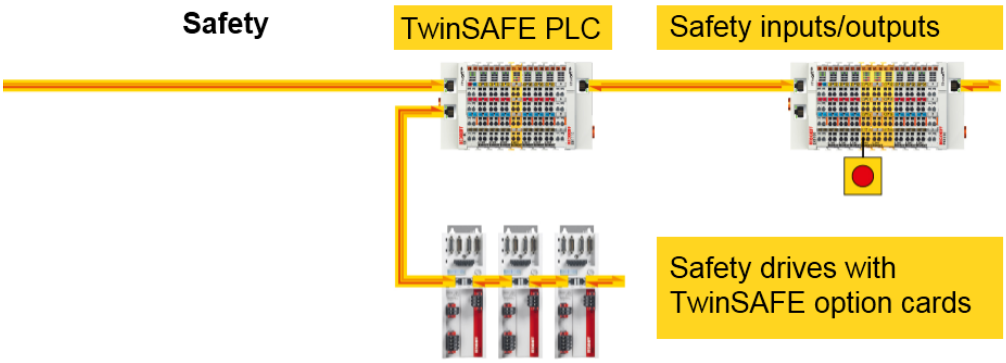
- 可无缝集成现有的总线系统，并能够兼容现有的总线端子模块产品。（可集成的总线系统如下图所示）



- 网络规模几乎无限制，理论值可以支持 65535 个节点。（下图是系统测试中心连接 10056 个节点的设备）

- 简单耐用，可精确定位故障位置，网络规划可轻松实现。在数据传输过程中可以通过有效的 32 位 CRC 校验码进行错误定位，还可以检测并定位 EMC 干扰、有缺陷的连接器或损坏的电缆等不断变化的错误来源。

- 可方便地将 Safety over EtherCAT 到标准 EtherCAT 网络，既满足 SIL 标准，又经 TUV 认可。

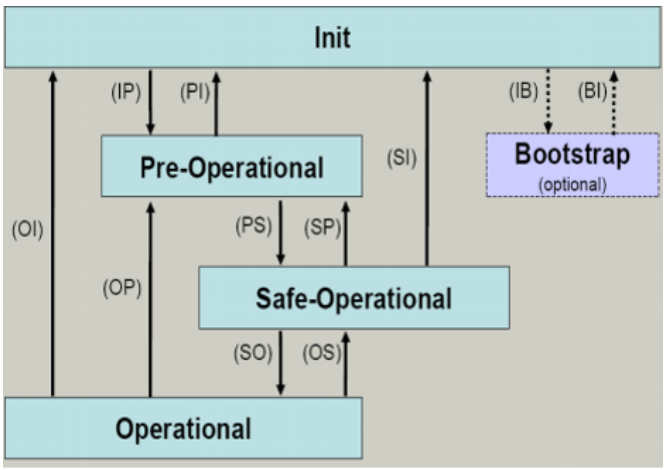


Ethercat 技术协会作为全球最大的工业以太网技术组织，截至 2016 年 2 月拥有会员人数已超过 **3500**。会员来自于全球各个自动化尖端领域，包括设备供应商、产品最终用户等。越来越多的自动化厂家推出 EtherCAT 产品。

（2） EtherCAT 状态机

每个 EtherCAT 从站都有状态机，分别为:

- Init
- Pre-Operational
- Safe-Operational and
- Operational
- Boot



每当一个 EtherCAT 从站设备启动，这个从站的状态机都会经过 INIT=>PREOP=>SAFEOP=>OP 的过程，BOOT 为更新模块 firmware 版本的模

式，OP 才是正常的工作模式，如果一个模块无法切换到 OP 模式的话，说明此模块配置有问题或者硬件出现故障。

设备的 EtherCAT 状态介绍：

Init	初始化状态	主要验证设备是否存在，以及型号是否正确
Preop	准备运行状态	Mailbox 通讯启用，可以带队从站的 COE 或 SOE 进行参数配置
SafeOP	安全运行状态	刷新 Input 变量但 Output 变量禁止
OP	正常工作状态	按任务周期刷新 Process Date(Input 变量和 Output 变量)

每当TwinCAT启动时，EtherCAT主站和从站设备的状态按以下顺序自动切换：

先初始化EtherCAT主站，初始化成功后主站切换到PreOP状态。

然后初始化每个从站，成功后也切换到PreOP状态。

等待任务启动，触发EtherCAT通讯。只要启动的PLC任务下有一个变量链接到一个EtherCAT设备的Process Data，那么每个PLC周期都会触发一次EtherCAT通讯。

主站切换到Safe OP状态

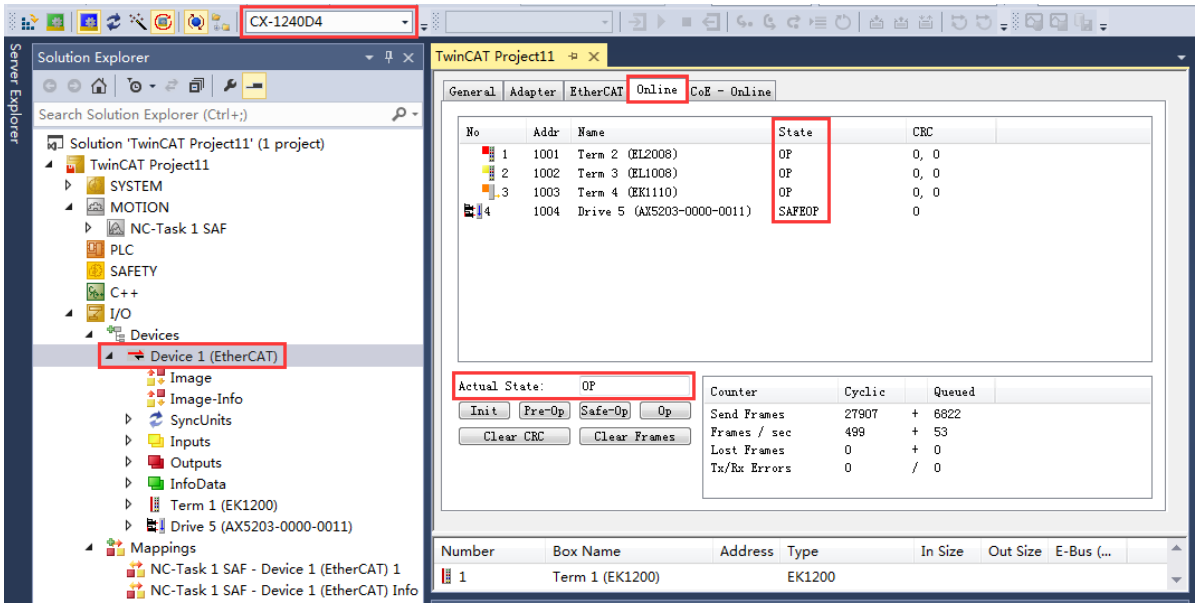
从站切换到Safe OP状态

主站切换到OP状态

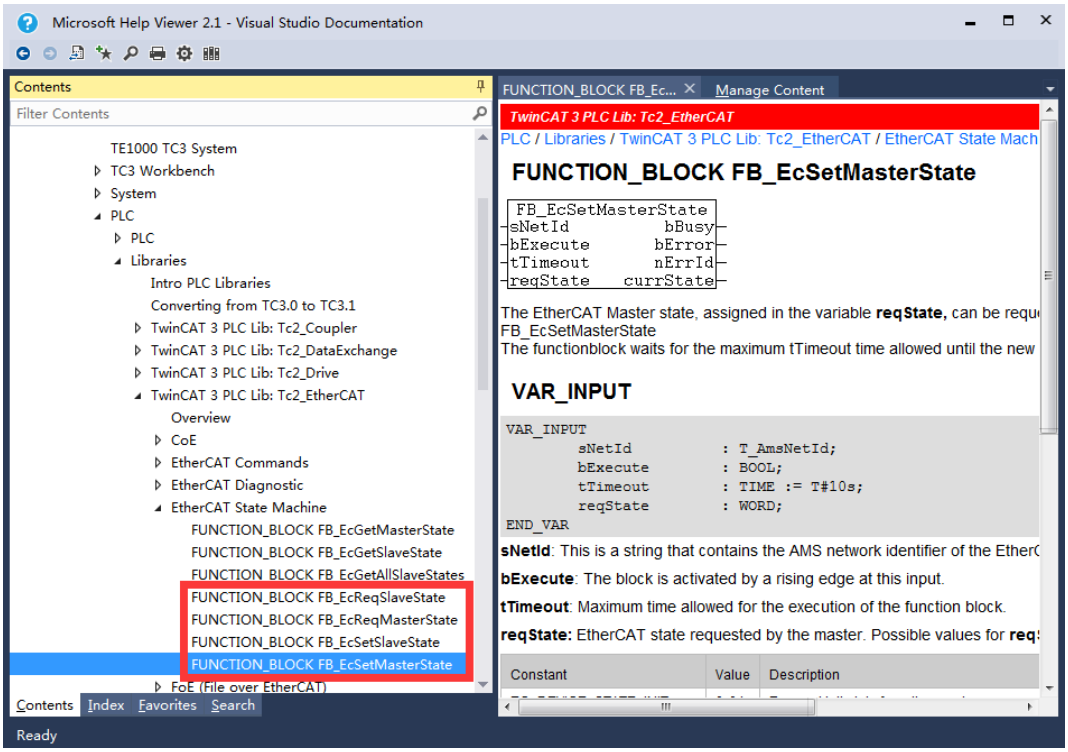
从站切换到 OP 状态

注意：从站的状态不能高于主站。比如，主站还在 PreOP 状态，从站不可能进到 OP 状态。

通过 System Manager EtherCAT 选项卡中 Online 可以方便地查看设备状态机：



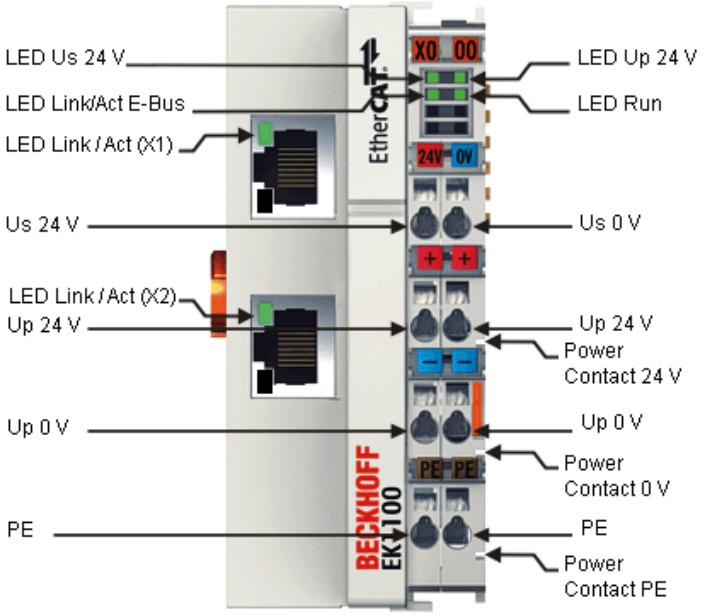
同样也可以使用相关的功能块，在程序中进行状态机监控：



获取EtherCAT主站的状态：FB_EcReqMasterState
设置EtherCAT主站的状态：FB_EcSetMasterState
获取EtherCAT从站的状态：FB_EcReqSlaveState
设置 EtherCAT 从站的状态：FB_EcSetSlaveState

（3） EK1100 EtherCAT 耦合器诊断

EK1100 耦合器用于将 EtherCAT 网线与 EtherCAT 端子 (ELxxxx) 相连。一个站由一个 EK1100 耦合器、任意多个 EtherCAT 端子组成。该耦合器将来自 100baseTX 以太网的传递报文转换为 E-bus 信号。耦合器通过上面的以太网接口与网络相连。下面的 RJ 45 接口可用于在同一条电缆上连接其它 EtherCAT 设备。



关于指示灯诊断：

- E-bus 的电源灯 Us 和 I/O 电源灯 Up 显示状态及意义

LED			意义
Us	绿	暗	耦合器以及 e-bus 没有供电
		亮	24 V 电源已连接
Up	绿	暗	I/O 触点电源没有连接
		亮	24V 电源已连接

- RUN 灯显示状态及意义

LED		状态（闪烁频率）	意义
RUN	绿	灭	INIT（初始化）：初始化终端
		闪烁（2Hz）	Pre-Operational（预操作模式）：邮箱通讯设置和变量标准设置
		闪烁（1Hz）	Safe-Operational（安全操作模式）：同步管理器和分布式时钟的通道检查，输出保持在一个安全操作状态
		亮	Operational（运行模式）：邮箱和过程数据通讯
		闪烁（10Hz）	Bootstrap（引导状态）：可实现功能如：终端的固件更新

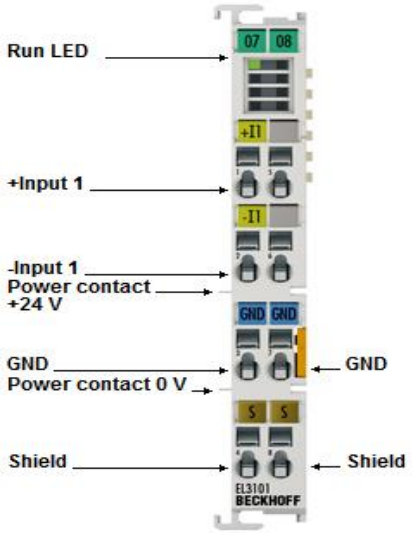
- LINK 口的 LED 灯显示状态及意义

LED			状态	意义
LINK / ACT (X1 IN)	绿	暗	-	与前一个 EtherCAT 设备没有连接
		亮	已连接	已经与前一个 EtherCAT 设备连接
		闪	活跃	与前一个 EtherCAT 设备连接并通信
LINK / ACT (X2 OUT)	绿	暗	-	没有与后一个 EtherCAT 设备连接
		亮	已连接	已经与后一个 EtherCAT 设备连接
		闪	活跃	与后一个 EtherCAT 设备连接并通信
LINK / ACT E-Bus	绿	暗	-	与 E-BUS 没有连接
		亮	已连接	已经与 E-BUS 连接
		闪	活跃	已经与 E-BUS 连接并通信

（4） E-BUS 模块诊断

很多 EL 模块都有 RUN LED 这个指示灯，此灯指示此模块的工作状态，只有 RUN LED 常亮才代表此模块工作正常。

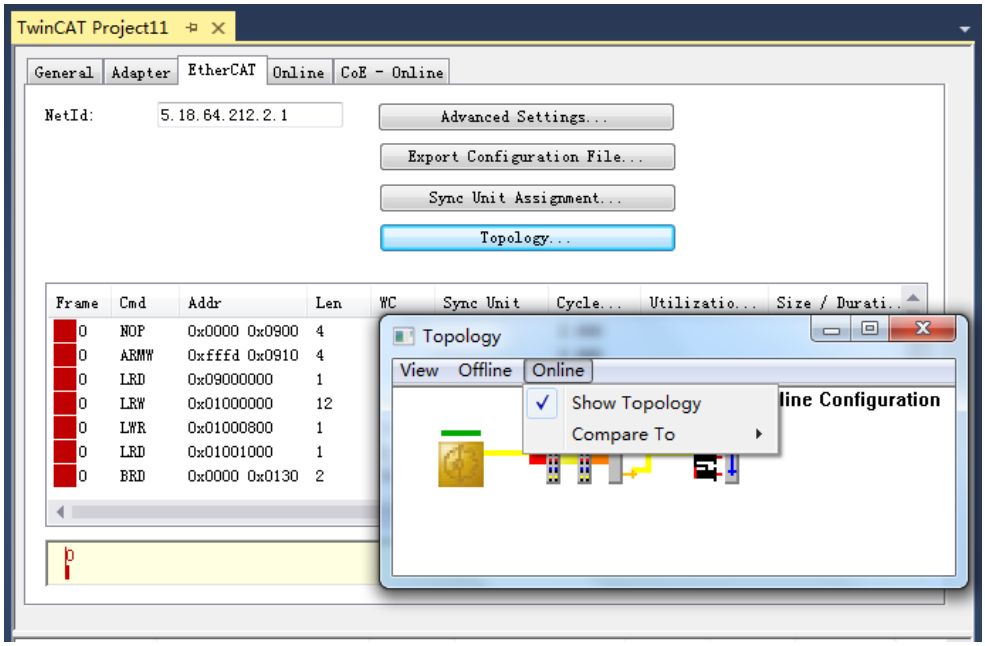
LED		状态（闪烁频率）	意义
RUN	绿	灭	设备处于初始状态
		闪烁（2Hz）	设备处于预运行状态
		闪烁（1Hz）	设备处于安全运行模式
		亮	设备处于运行状态
		闪烁（10Hz）	设备处于更新状态（耦合器固件更新）



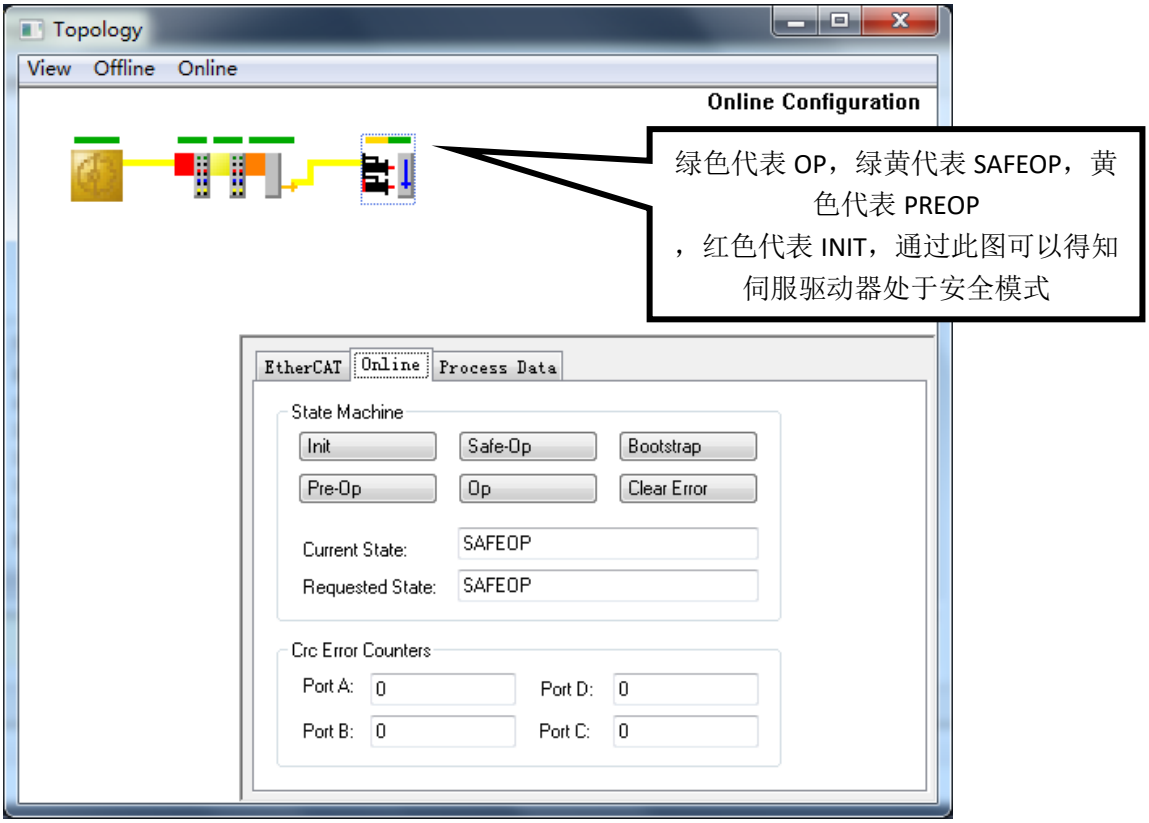
（5） TwinCAT3 软件诊断

- 通过拓扑结构查看各个设备的工作状态

首先扫描 I/O 硬件，然后点击 Device1(EtherCAT)，然后选择 EtherCAT 选项卡，点击 Topology，在弹出的对话框的 Online 选项卡中将 Show Topology 勾选，通过这个方式可以在线查看整个 EtherCAT 网络的拓扑结构。

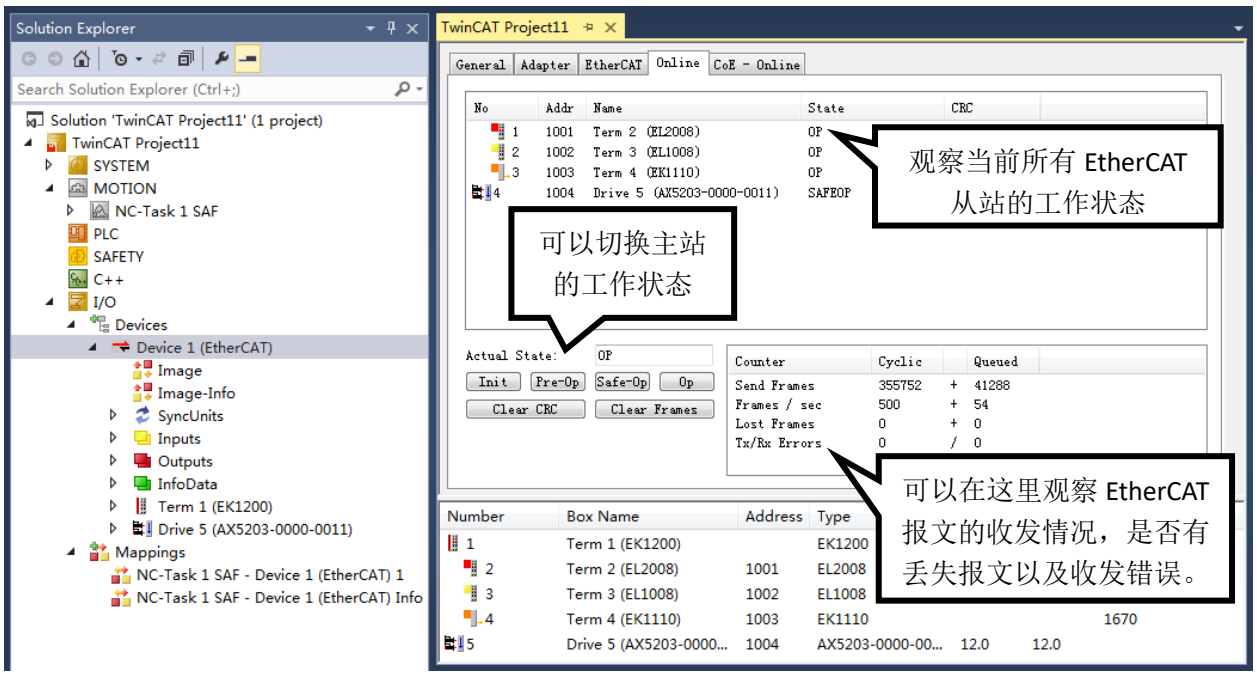


例如下图所示，可以观察模块上面横条的颜色来判断模块的状态。将伺服驱动器的网线拔掉，此时驱动器上面的状态会变为红色，重新插上网线后，驱动器会自动从 init 模式向 op 模式切换。

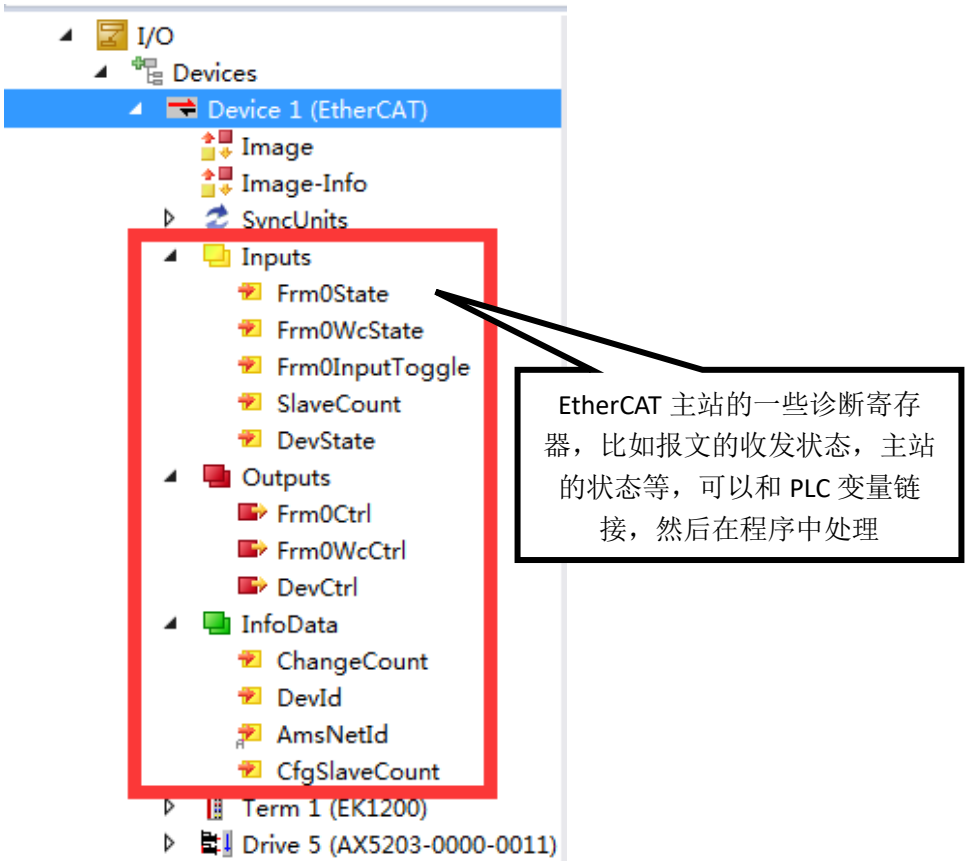


- 在 Online 选项卡中查看各个设备的工作状态

Online 选项卡里面可以观察所有从站的工作状态，EtherCAT 报文的收发情况，以及切换 EtherCAT 主站的工作状态。

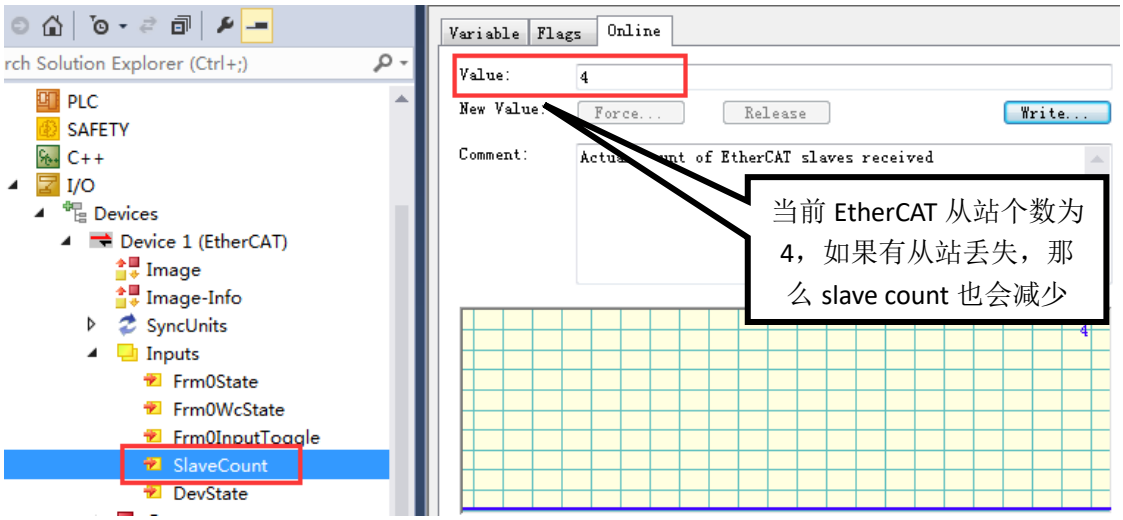


- EtherCAT 自带了很多诊断寄存器，可以和 PLC 变量链接



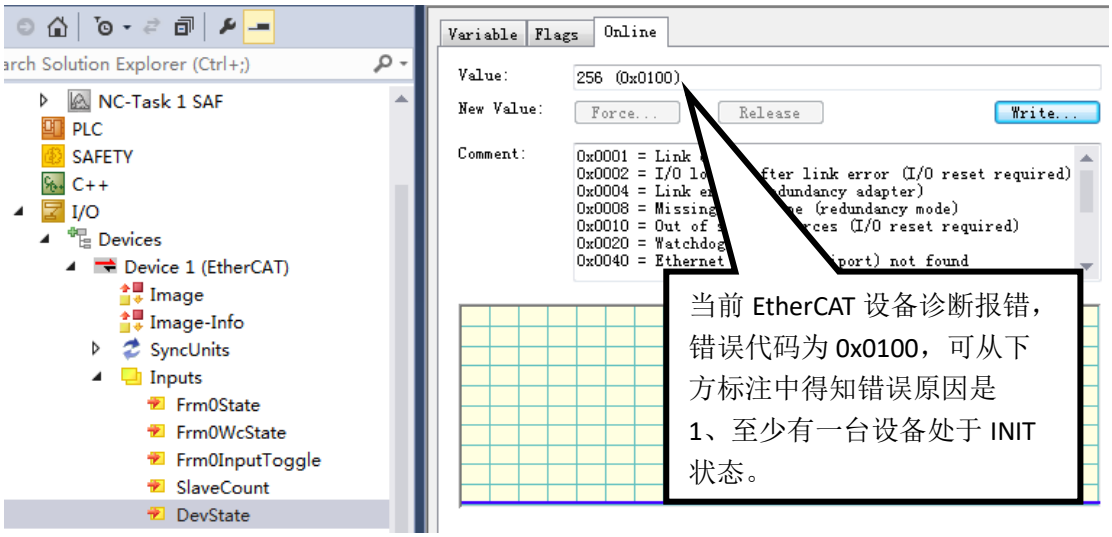
- SlaveCount:发现的从站数量

这一状态字用来反映连接的从站所带的模块或驱动器的数量。一旦模块发生异常情况 SlaveCount 都会第一时间做出反映，显示的数字是处于正常运行状态无报错的数量总和。

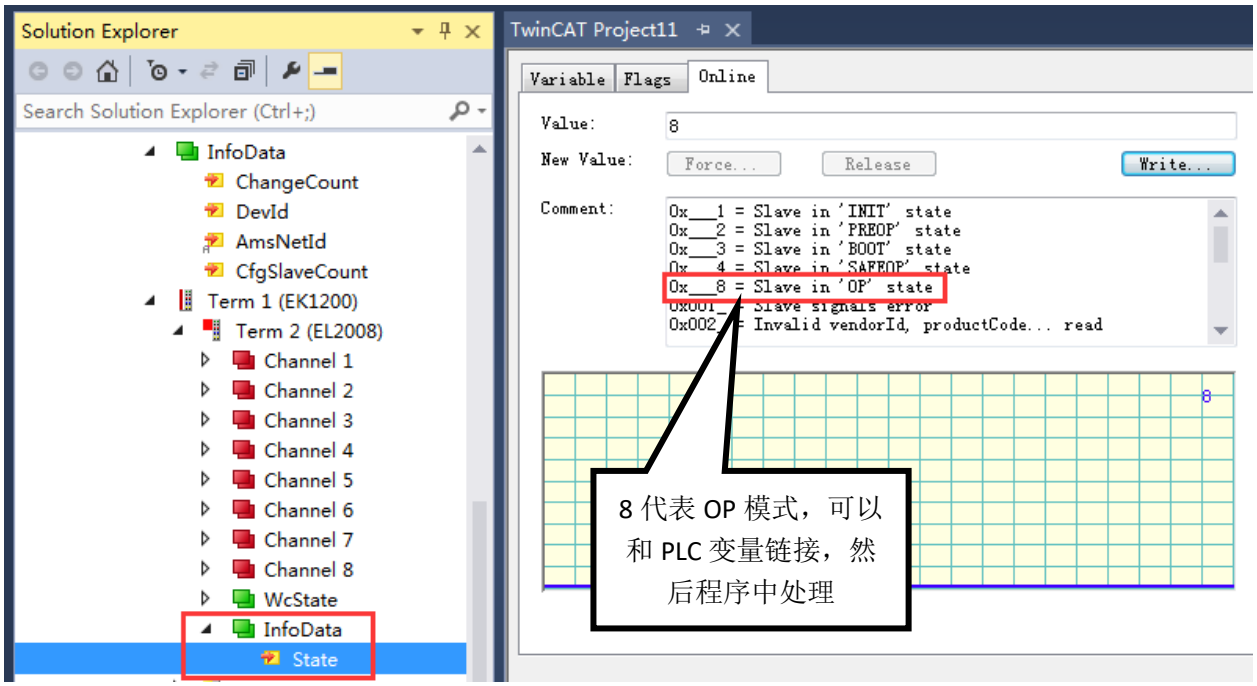


- DevState:设备状态信息状态字（WORD）

这一状态字用来反映设备状态信息的，一旦设备诊断报错，出错原因就会在这一位状态字上进行显示。



- 每个从站也有自己的寄存器可以反映模块的当前工作状态。



2. EtherCAT 驱动安装步骤

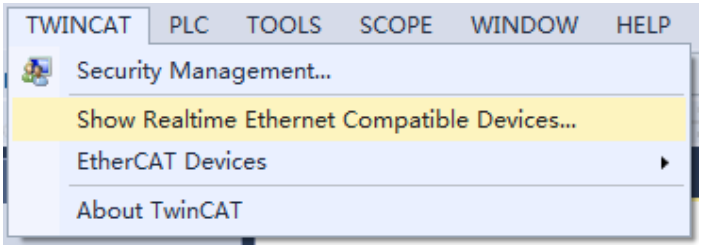
如果使用倍福 IPC 直接扫描 EK1100 等 EtherCAT 从站设备，那么因为 EK1100 上的网口走 EtherCAT 协议，而 PC 端的网口默认支持 Ethernet 协议，所以在扫描 IO 模块之前，首先需要在 PC 端安装 EtherCAT 驱动。

针对 EPC 连接 EtherCAT 设备时，通常不需要手动安装 EtherCAT 驱动，因为出厂的时候已经完成了对嵌入式 PC 的驱动安装。

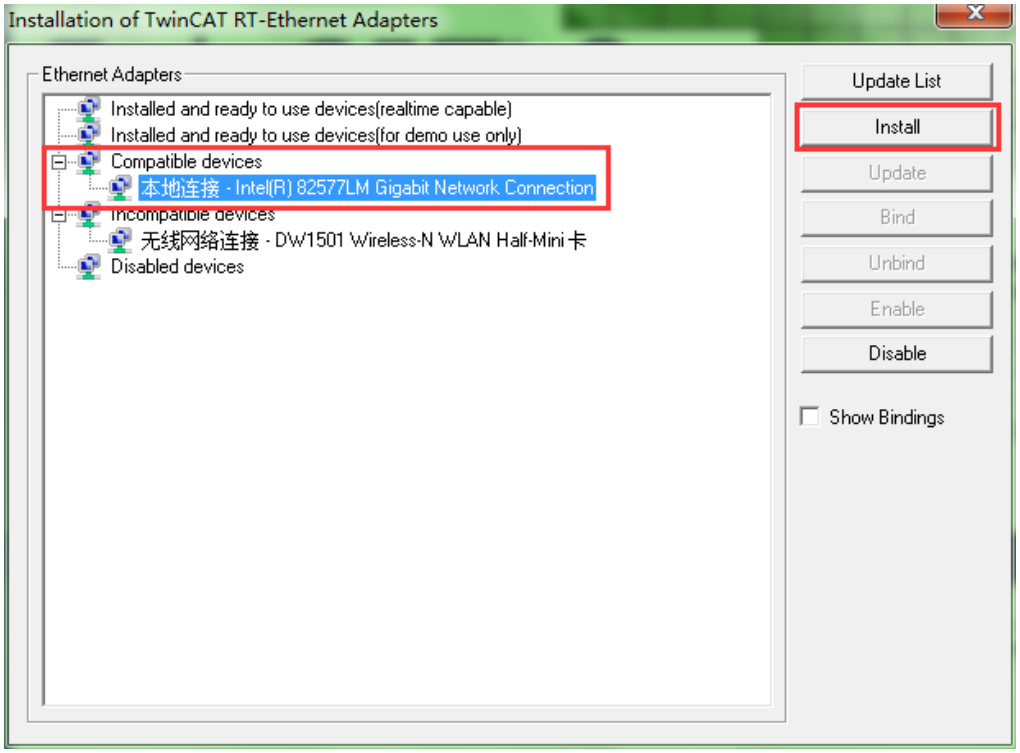
这里以本地为例。将 PC 与 EK1100 上面的以太网口连接，确保指示灯显示状态正确，模块和 PC 之间通讯正常。

如果 PC 端已经安装了 TwinCAT3 Full 版本，请根据以下方式进行 EtherCAT 驱动安装。

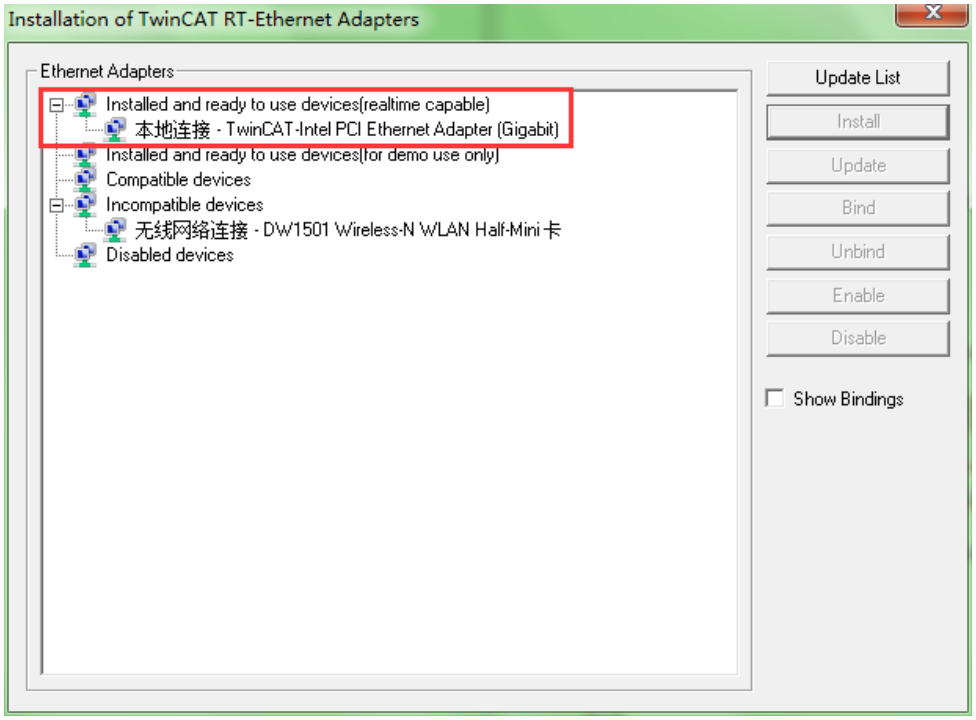
（1）打开 TwinCAT3，找到菜单栏 TWINCAT 下的 Show Realtime Ethernet Compatible Devices…并单击。



在弹出的对话框中找到 Compatible devices 下需要安装 EtherCAT 驱动的网卡并选中，点击右侧的 Install 进行 EtherCAT 驱动安装。

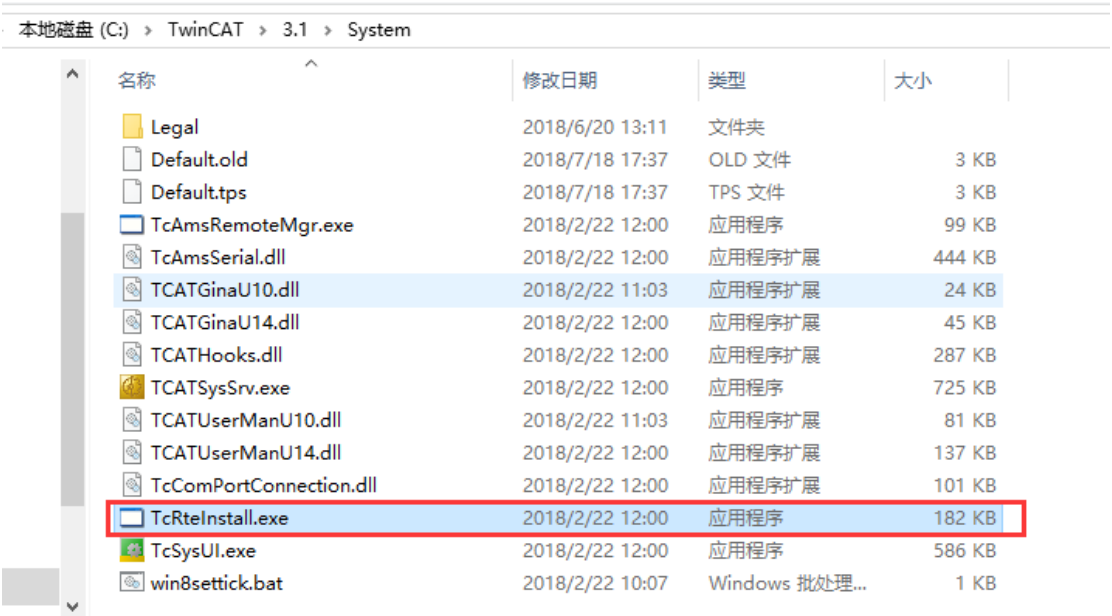


安装完成后，在 Install and ready to use devices(realtime capable)下就会显示已经安装 EtherCAT 驱动的网卡。

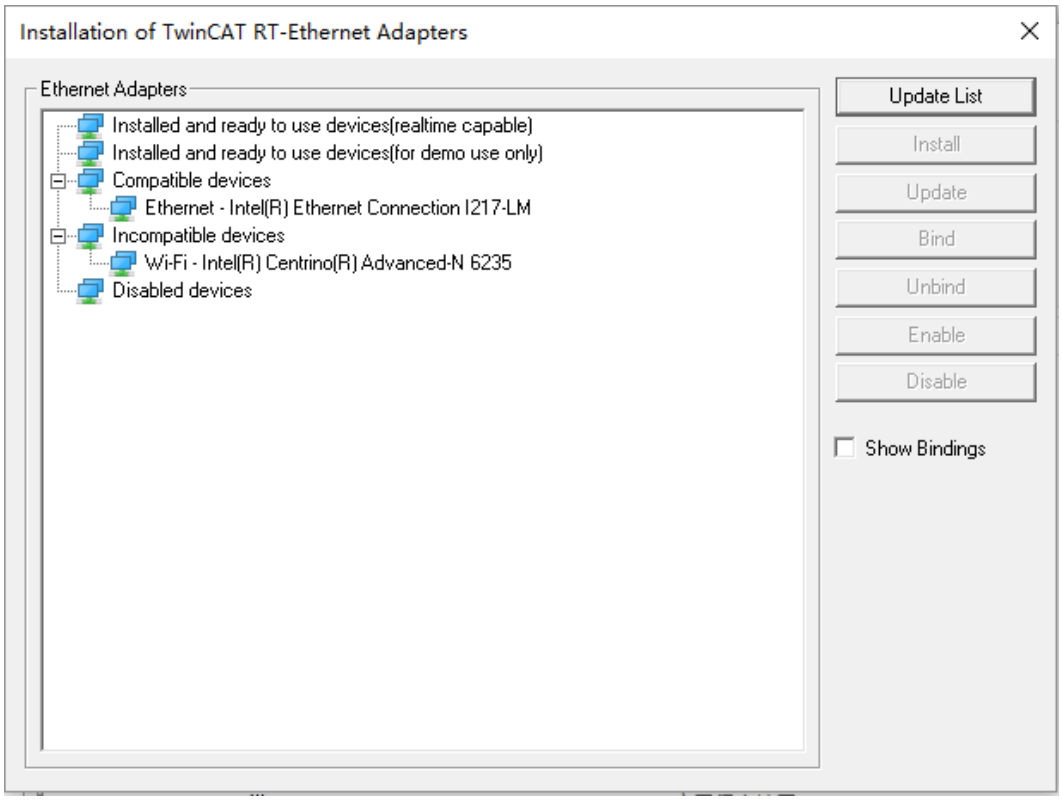


（2）如果倍福 IPC 或者 EPC 单独安装 TwinCAT3 的 XAR，系统是不会自动加载 EtherCAT 驱动的，此时无法进行 EtherCAT 设备的扫描，需要手动安装驱动。请根据以下方式进行 EtherCAT 驱动安装。

在 TwinCAT 的安装盘找到如下路径中的文件 C:\TwinCAT\3.1\System\TcRteInstall.exe



执行之后会弹出如下对话框：



之后只需要参考方法（1）中的说明安装 EtherCAT 网卡驱动即可。

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