BECKHOFF

TwinCAT SFC编程入门教程

Version 1.11

毕孚自动化设备贸易（上海）有限公司

2019 年 4 月

SFC 编程入门

目录

一、SFC 编程介绍

PLC 的编程语言主要有以下几种：梯形图（LD）、指令表（LI）、顺序功能图（SFC）、结构化文本（ST）、连续功能图编辑器（CFC）。这五种编程语言都是符合 IEC61131-3 标准的编程语言。

顺序功能图（Sequential Function Chart）是近年来发展起来的一种程序设计语言。它采用顺序功能图描述程序结构，把程序分成若干“步”（Step），每个步可执行若干动作。而“步”之间的转换靠其间的“转移”（Transition）的条件来实现。至于在“步”中要做什么，在转移过程中有哪些逻辑条件，则可以用其他任何一种语言（例如 ST 结构化文本）来实现。

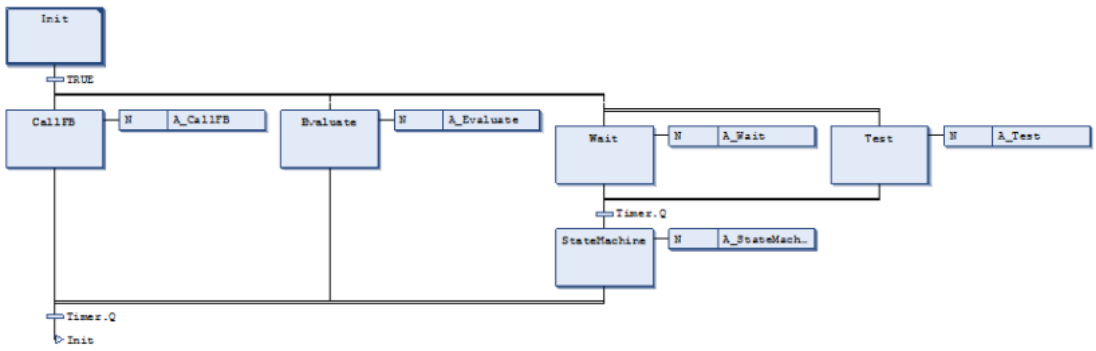
SFC 编程的特点：

1、 以功能为主线，条理清晰，便于对程序操作的理解和沟通；在程序中可以很直观地看到设备的动作顺序。比较容易读懂程序，程序按照设备的动作顺序进行编写，规律性较强。即使不是程序编写人员，在初看时，也非常容易上手理解。

2、 对大型的程序，可分工设计、采用较为灵活的程序结构，可以节省程序设计时间和调试时间，在设备故障时能够很容易的查找出故障所处在的位置，不用检查整个冗长的程序。

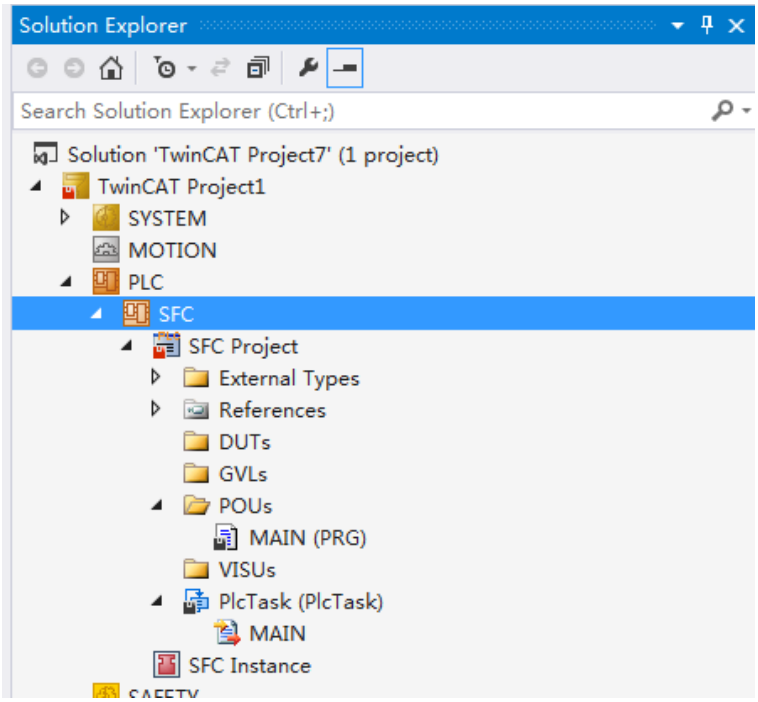
3、 常用在系统规模大，程序关系较为复杂的场合。不需要复杂的互锁电路，更容易设计和维护系统。

SFC 程序的基本运行顺序是：从初始步开始，依次执行每一个步，每次转移条件成立时执行下一步，走到末尾会返回到初始步，然后进行循环执行。



二、创建 SFC 工程

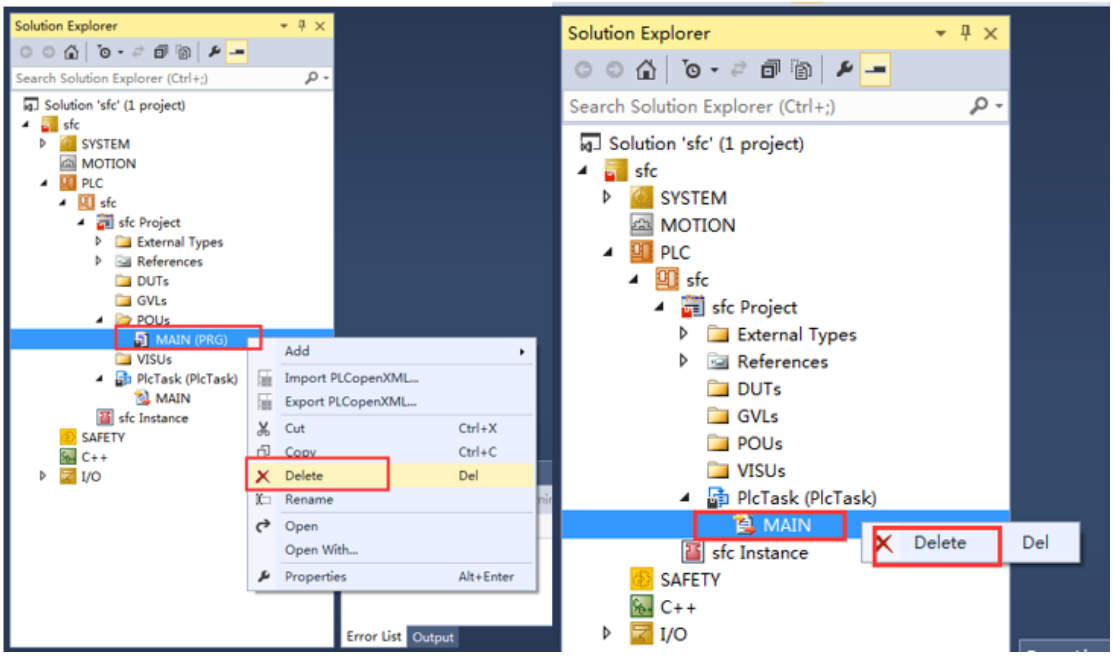
1. 打开 TwinCAT3 软件，新建工程并且在 PLC 下新建一个新 Project 项目



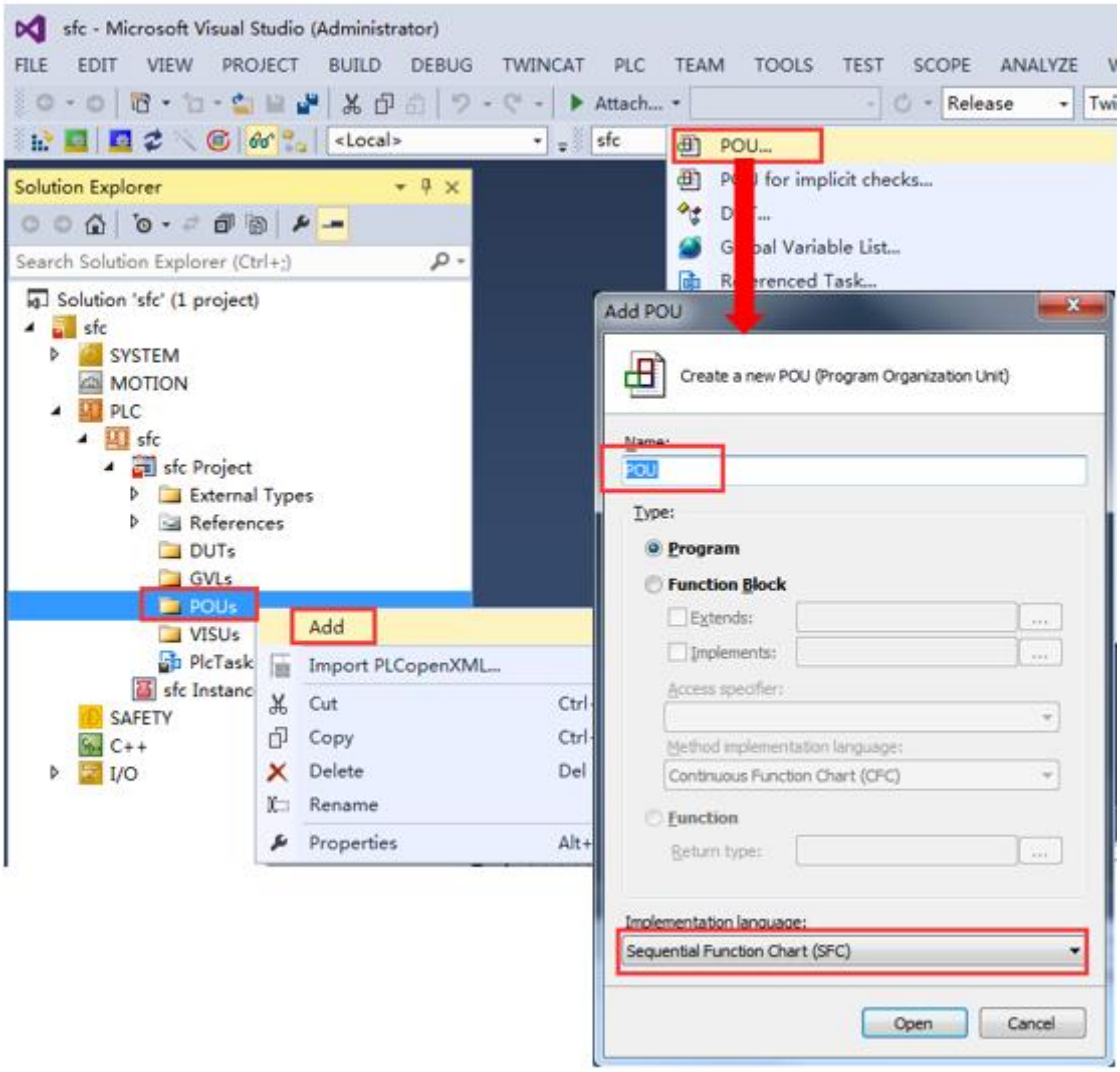
注意：

由于 TwinCAT3 软件新建的 PLC 程序是默认生成使用 ST 语言的主程序，所以需要先将原先的主程序删除，添加新的使用 SFC 语言的程序。

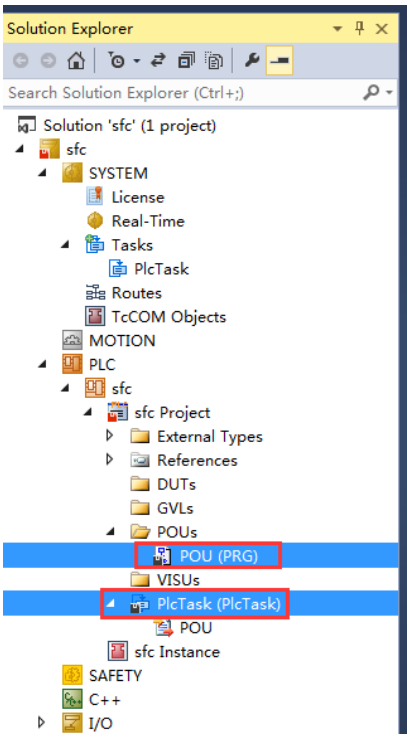
2. 删除 POUs 下方的 MAIN（PRG）主程序，和 PlcTask 下方的 MAIN 任务



3. 在 POUs 下方添加新程序，并将编程语言手动选为 SFC 顺序功能图；



4. 将创建好的 SFC 语言程序 POU 选中通过“拖动”，添加到 Plc Task 中。



5. 这样就创建了一个以 SFC 为编程语言的 PLC 程序

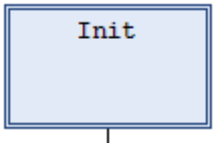
三、SFC 编程基础入门

顺序功能表图编程语言的基本图形符号是**步**、**转换**和**有向连线**。

1. Step 步

顺序功能表图编程语言把一个过程循环分解成若干个清晰的连续的阶段，称为“步”（Step）。步的图形表示:

Init Step 初始步用带步名的双线矩形框表示；

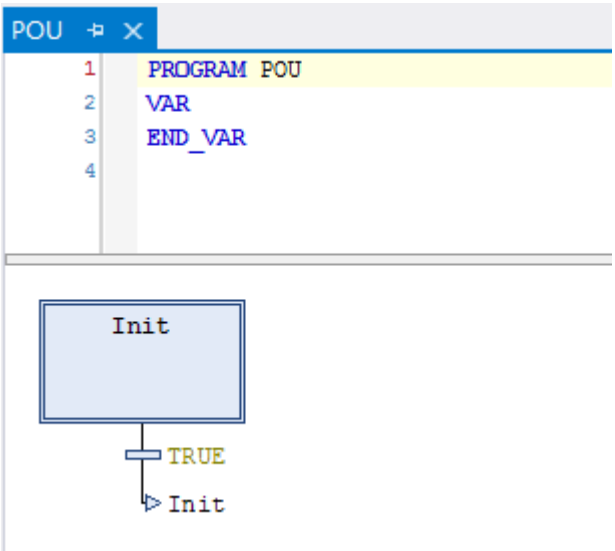


普通步，中间步用带步名的单线矩形框表示；

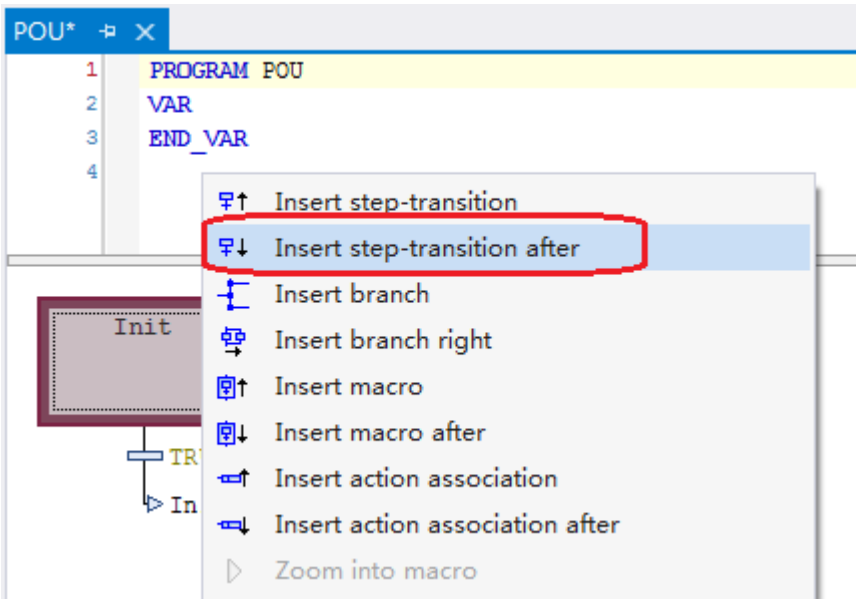
初始步放在程序的开始处，执行该步无需条件，此时所有其他步都不执行。另外初始步可以放置在程序中的任何一位，不一定需要放在程序开头。每个程序只有一个初始步，但在初始步中可以有多个动作块。

添加 Step 步的步骤:

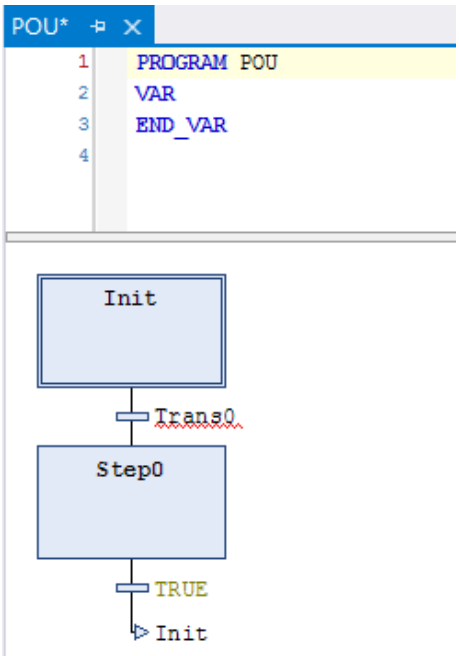
（1） 双击打开 POU 程序，便可以看到已经默认添加的 Init Step



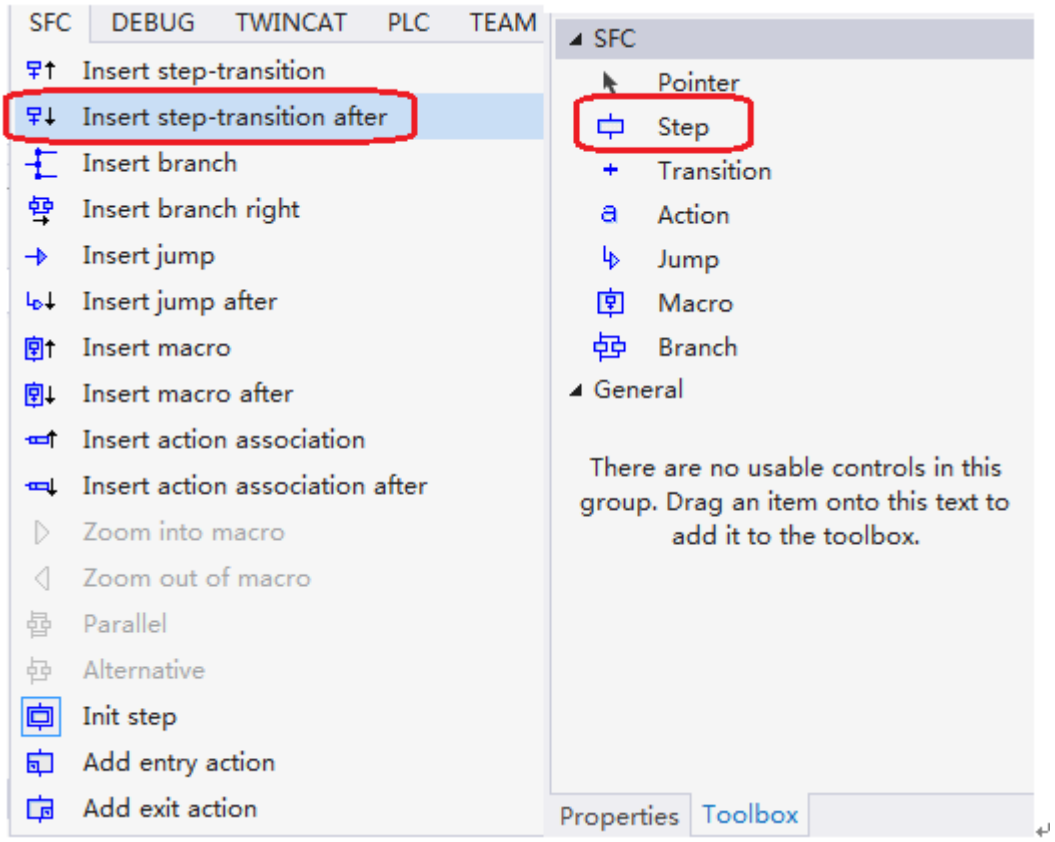
（2） 右键 Init Step，在弹出的对话框中选择 Insert step-transition after



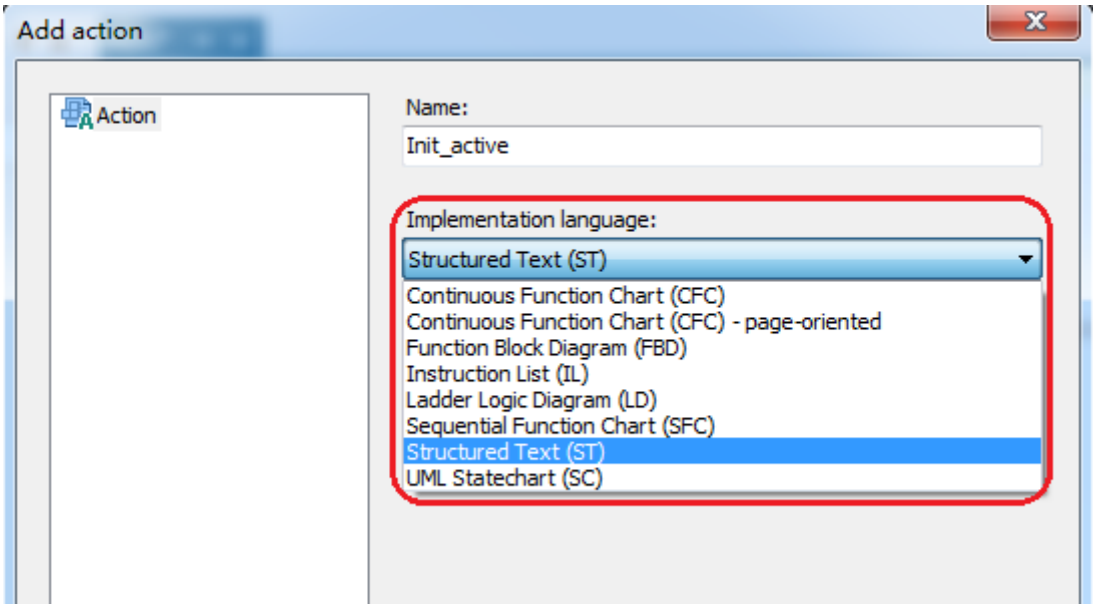
这样就可以在 Init Step 后添加一个新的 Step



另外，也可以通过 SFC 菜单栏和 Toolbox 进行 Step 的添加



（3） 双击进入 Step 步完成对步程序的编写



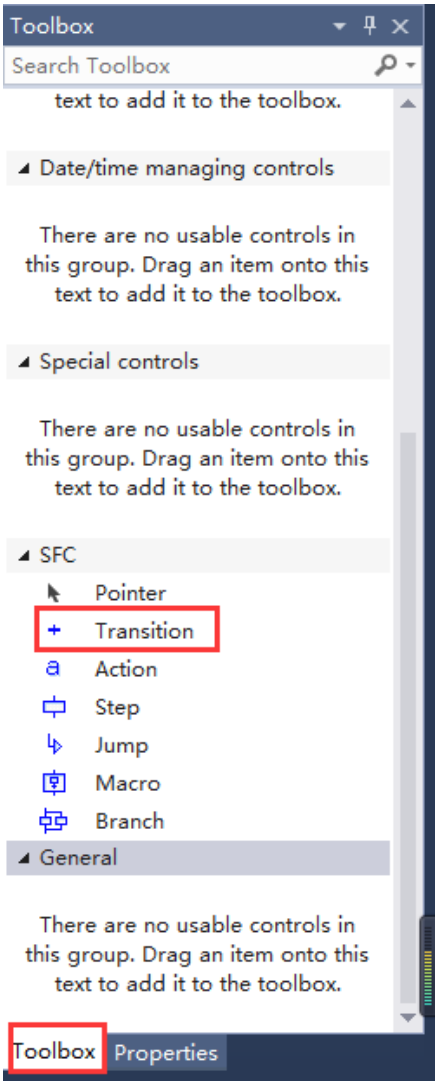
可以看到在每一个 Step 中编程语言也十分丰富，例如 ST 结构化文本、LD 梯形图、FBD 功能块图等。用户可以根据实际项目，选择最合适的编程语言对 Step 步程序进行编写，提高了编程的灵活性。

2. Transition 转换

顺序功能表图中，步活动状态的进展按照有向连线规定的路线进行。在每一个 Step 步之间必须，并且只能有一个转移条件（Transition）。转移条件可以是一个变量，也可以是一个或一组判断条件。当转移条件满足时，程序会向下转移，执行下一个 Step 步中的程序。

添加 Transition 转移条件的步骤:

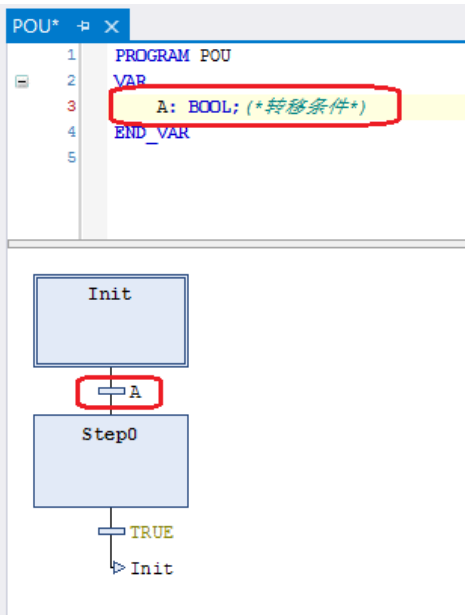
（1） 通过界面右侧的 Toolbox 进行添加 Transition：



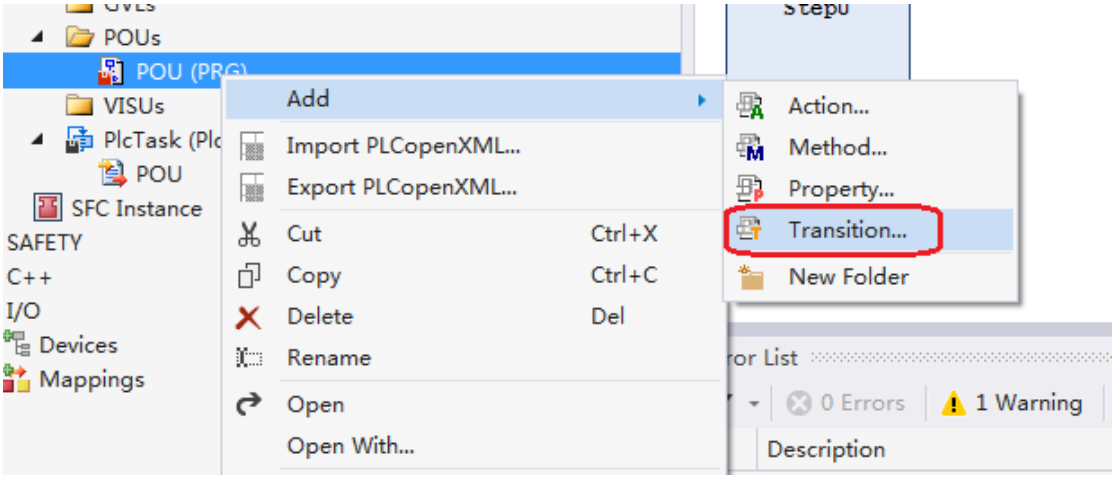
（2） 手动修改转移条件

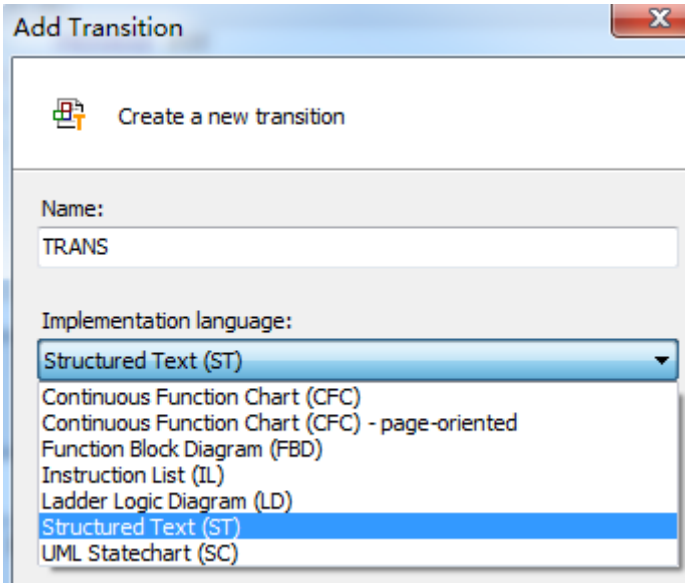
当转移条件为 True 时，则每次自动向下一个步转移；

当转移条件为 Bool 变量时，则当该变量置位 True 时向下一个步转移；

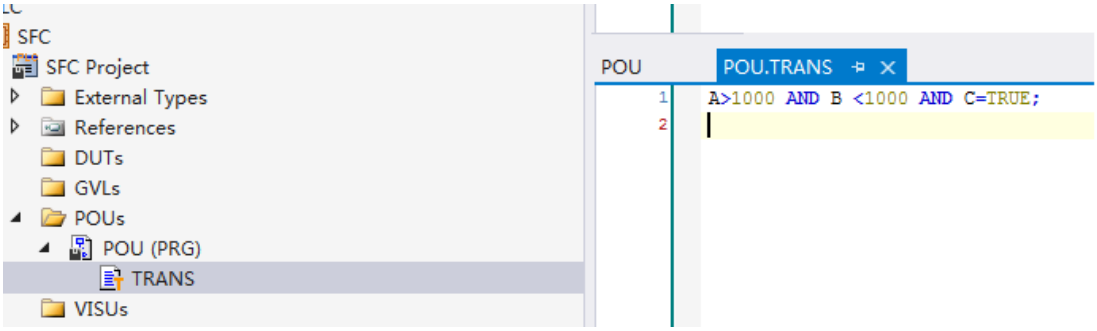


（3） 如果需要些多个转移条件时，可以在 PLC 程序下新建一个 Transition 用来转移条件的编写，同样转移条件也可以使用多种编程语言来进行编写。

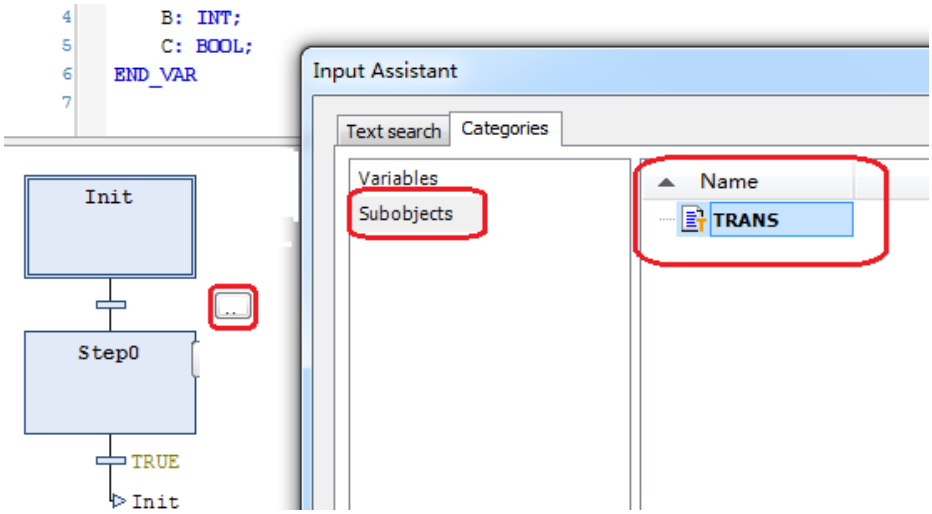




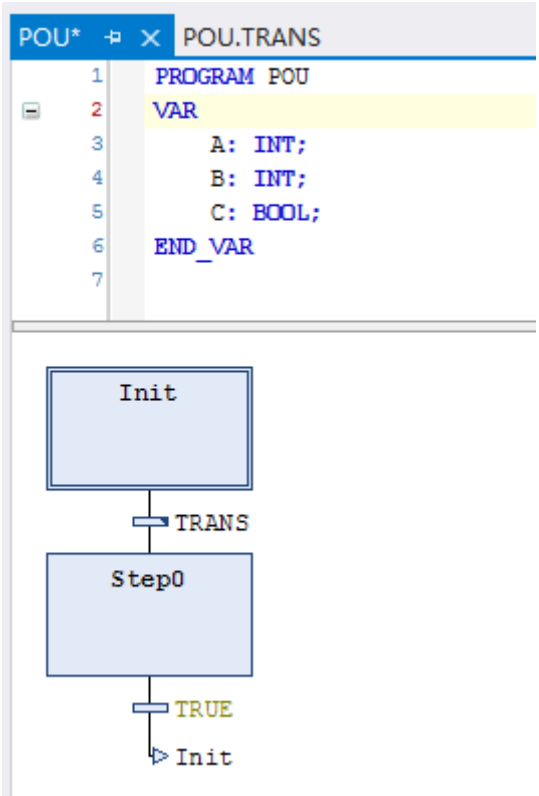
（4） 在新建的 Transition 转移条件中就可以输入多个转移判断条件



（5） 在对应的转移条件位置，对转移条件 TRANS 进行添加



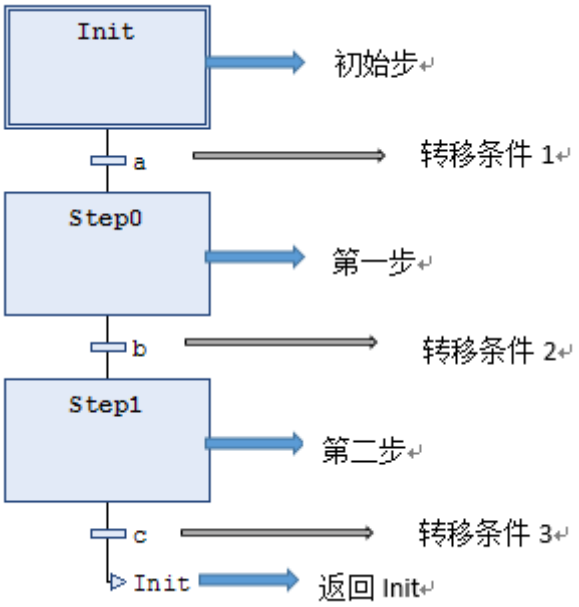
（6） 这样就完成了转移条件的添加。当 ABC 满足转移条件时，程序才会向下一个步转移。



3. 串行转移

当满足转移条件时进入当前步后面的一步处理的转移。

1) 当起动 SFC 程序时，首先执行起始步 Init。在初始步处理期间检查下一个转移条件（图示“转移条件 1”）以确定是否满足该转移条件。
2) 初始步处理继续，直到满足转移条件 1，当满足转移条件 1 时，初始步处理停止并开始下一步 Step0 的处理。
3) 在 Step0 处理期间，检查下一个转移条件（图示中的转移条件 2）以确定是否满足该转移条件。
4) 当满足转移条件 2 时，停止 Step0 处理并开始下一步 Step1 的处理。
5) SFC 程序的处理以该方式继续按顺序执行步和转移直到到达 Init 返回处。

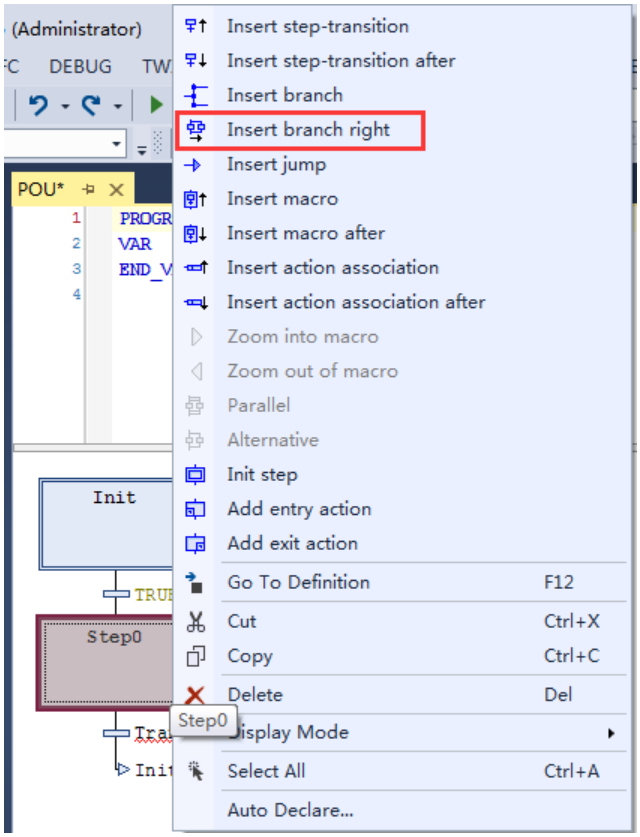


4. 选择分支

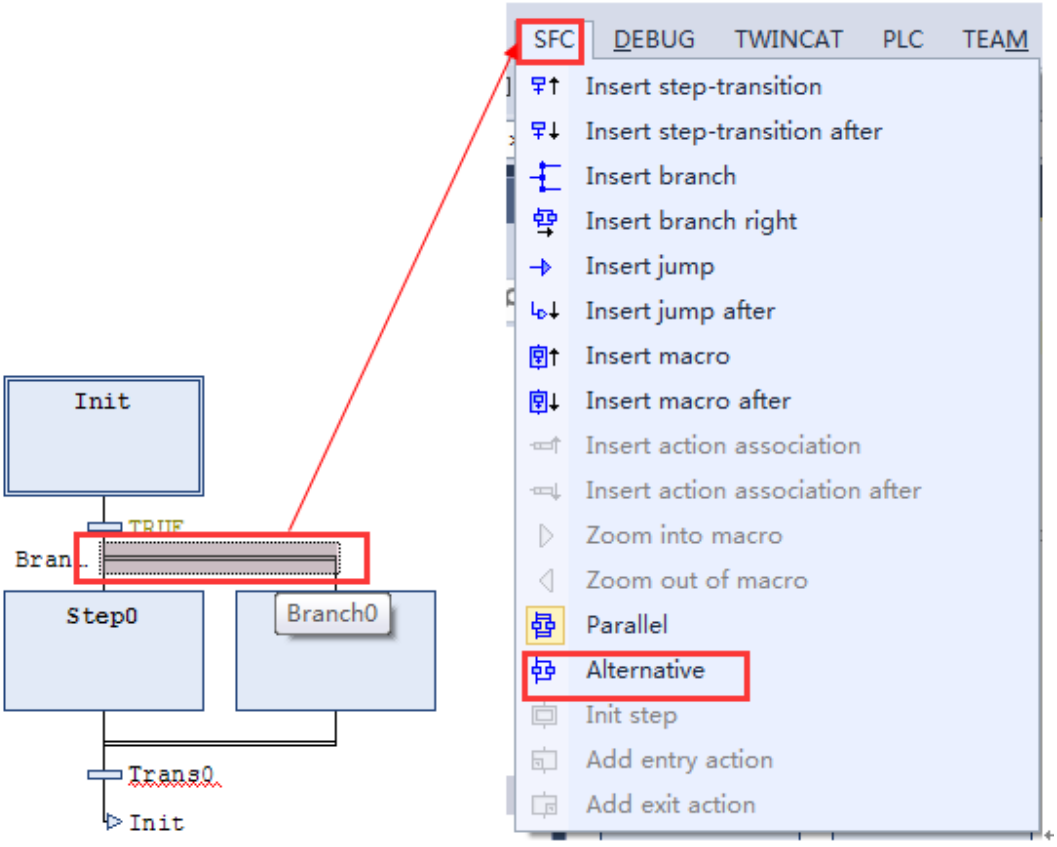
若干步以并行方式汇合在一起的转移格式，并且只处理首先满足转移条件的步。

添加选择分支的步骤:

（1） 创建选择分支结构，首先选中旁侧需要创建分支结构的 Step，然后右击，弹出菜单框，单击 Insert branch right，创建完成。

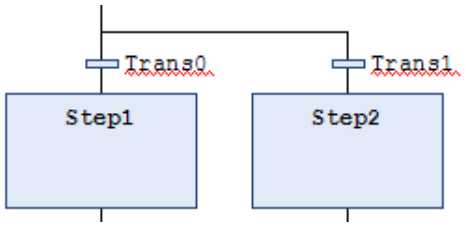


（2） 创建好旁侧分支后，选中分支线段，选择 SFC 菜单栏将默认的 **Parallel 平行分支**修改为 **Alternative 选择分支**。

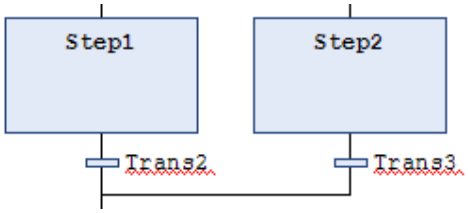


（3） 这样就完成了选择分支的创建

对于选择分支结构来说，特征如下：



一个单步连接到两个或多个转换以后，合并一个已被分割成两个或多个分支工艺流程的工艺流程。



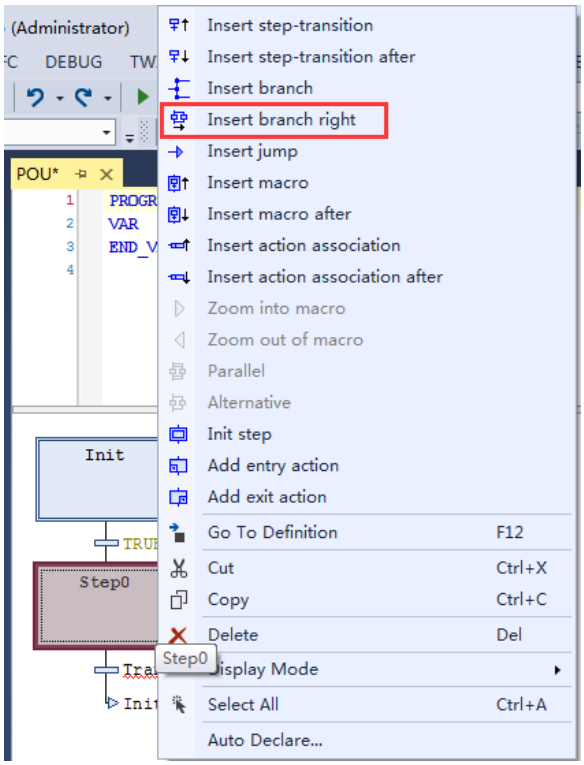
所以在使用选择转移时，需要先对每个分支进行转移判断是否可行，同时在每个分支选择汇合处也先进行转移判断是否可行。如果一个分歧的多个转换条件同时满足，默认时优先权将赋予最左边一个步。

5. 平行分支

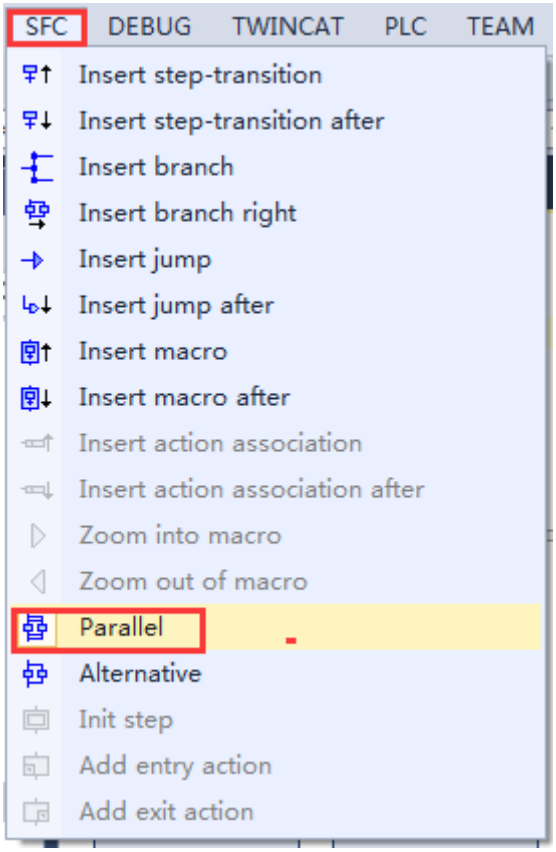
在满足相关的转移条件时同时处理并行链接的若干步的转移格式。

添加平行分支的步骤：

（1） 创建选择分支结构，首先选中旁侧需要创建分支结构的 Step，然后右击，弹出菜单框，单击 Insert branch right，创建完成。

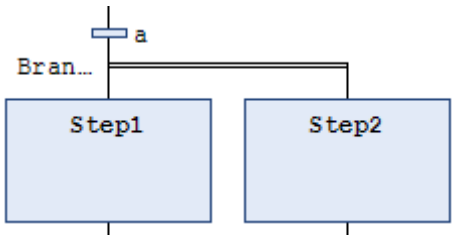


（2） 创建好旁侧分支后，选中分支线段，选择 SFC 菜单栏的 **Parallel 平行分支**

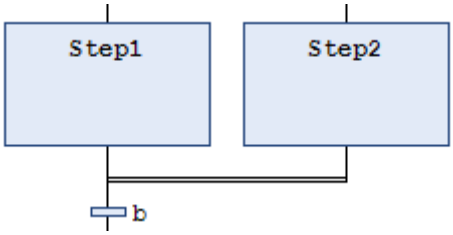


（3） 这样就完成了平行分支的创建

对于平行分支结构来说，特征如下：



转换条件达到活动状态，同时转换到分歧后的所有步，紧跟转换，双线后的步同时激活，执行顺序按照编辑的动作块的先后顺序执行。

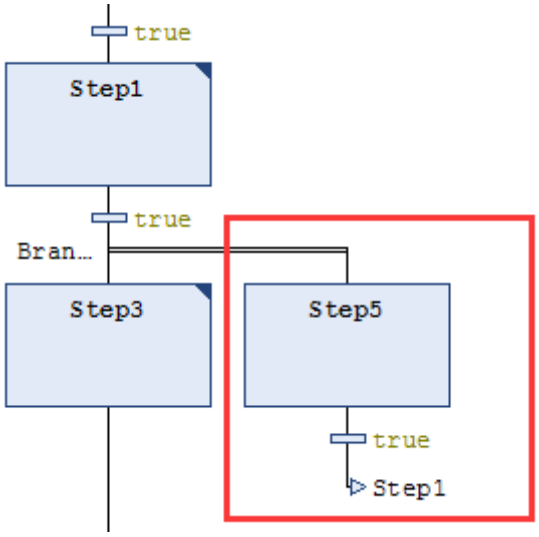


所以在使用并行转移时，需要先在进行并行分支前判断转移条件是否满足，满足后才能够进行并行步分支，同时在每个分支并行汇合处，汇合后也先进行转移判断是否可行。

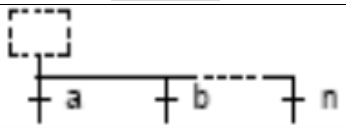
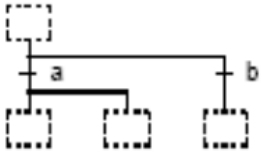
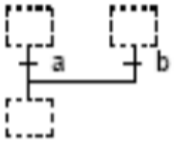
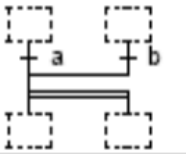
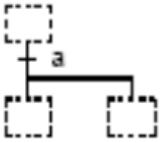
6. 跳转

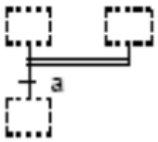
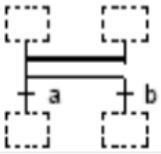
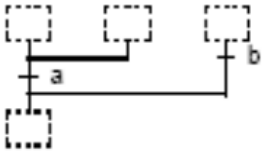
指当满足转移条件时跳转到指定步，开始执行对应步程序。

对于跳转，在转换后有一个箭头和跳转目地步号。如图示，在 step5 完成后，通过转移判断后，有一个跳转箭头指向 step1，也就是说该分支完成后会自动跳转到 step1 步，并且激活 step1 使其处于活动状态。



常见转移类型总结：

类别	名称	SFC 图符号
转移	串行转移	
	选择分支	
	选择分支—并行分支	
	选择汇合	
	选择汇合—并行分支	
	并行分支	

	并行汇合	
	并行汇合—并行分支	
	并行汇合—选择分支	
	并行汇合—选择汇合	
	跳转	

四、SFC 进阶使用

1. 输入输出步

基本概念：

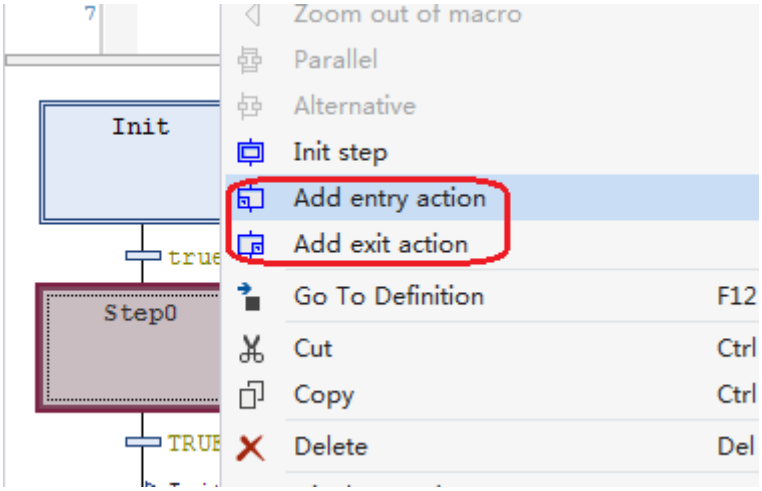
输入输出步受所在步的控制。当所在步被激活的瞬间，输入步先进行执行，随后再执行步程序；当所在步结束动作的瞬间，输出步立即开始执行。

应用场合：

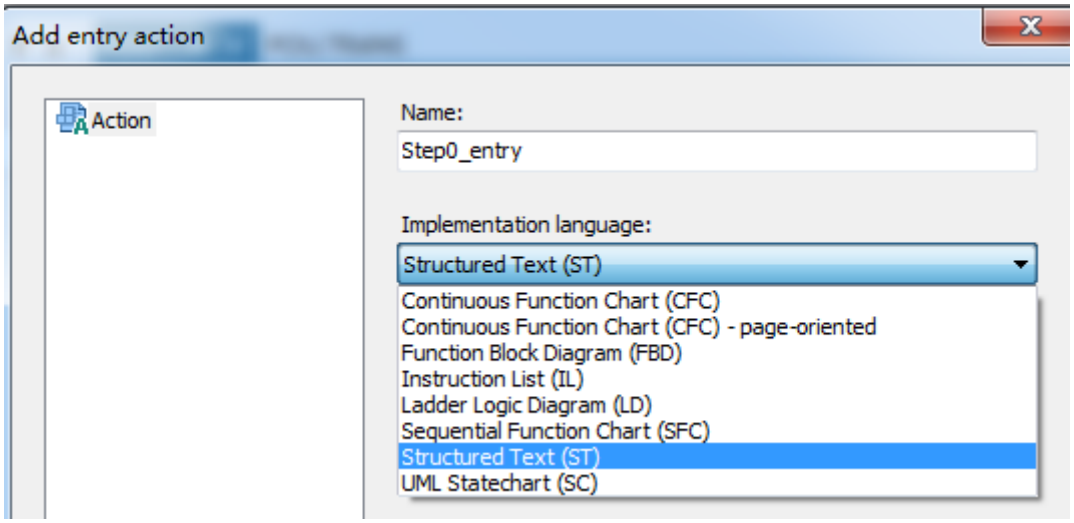
在执行程序前对参数进行初始化赋值；在执行完程序后，需要将程序处理的结果进行保存。这一系列程序动作，就可以在基本 Step 步的基础上，采用“输入输出步”来使得程序逻辑更加清晰。

创建步骤：

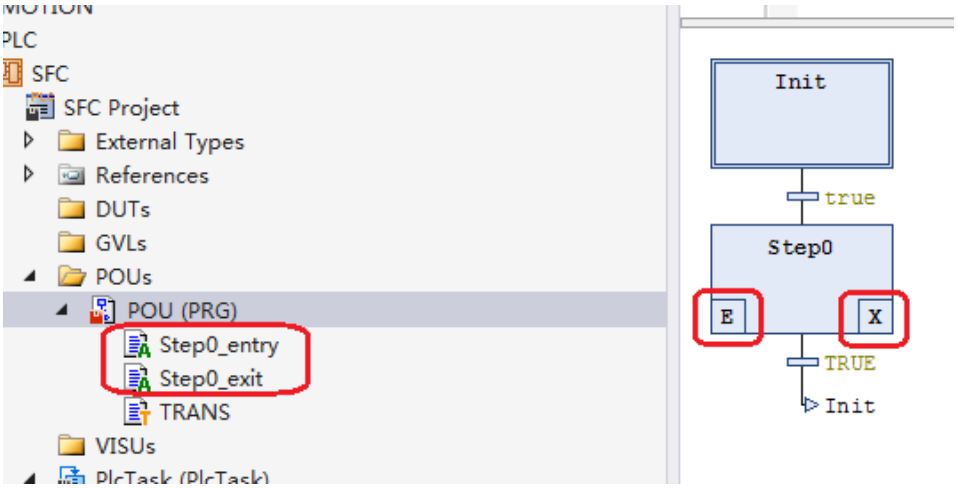
（1） 右键对应 Step，在弹出的对话框中选择 Add entry action 和 Add exit action



（2） 在对应添加的输入/输出步中，可以使用多种编程语言进行输入输出步程序的编写

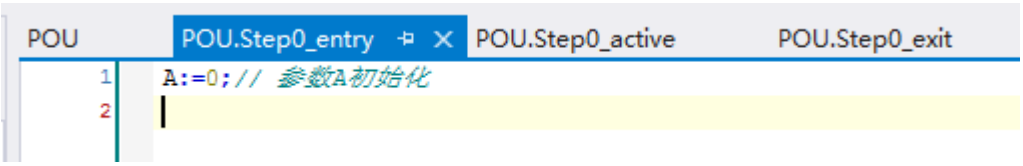


（3） 添加完成后，可以在所在步看到对应标识，标识该步有输入步和输出步。其中 E 执行 Step0_entry；X 执行 Step0_exit

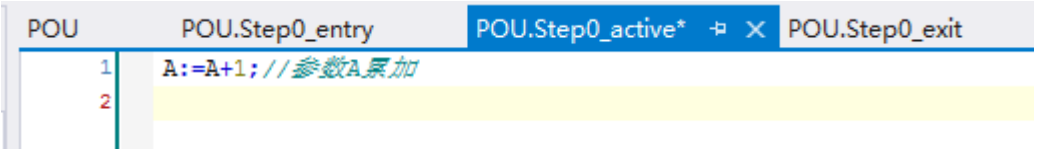


使用举例：

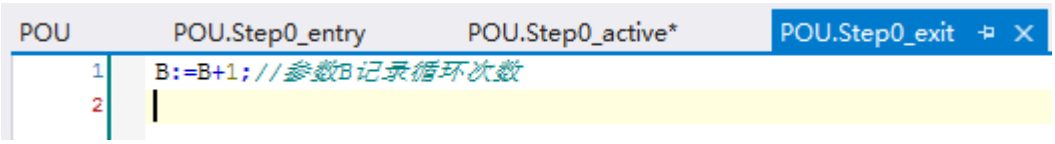
输入步 Step0_entry：



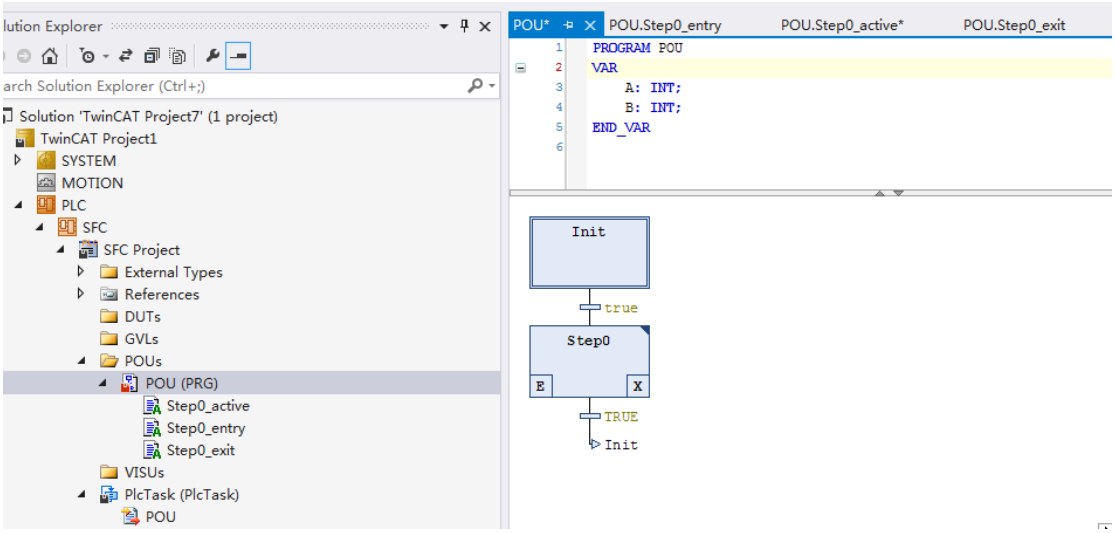
执行步 Step0_active：



退出步 Step0_exit:



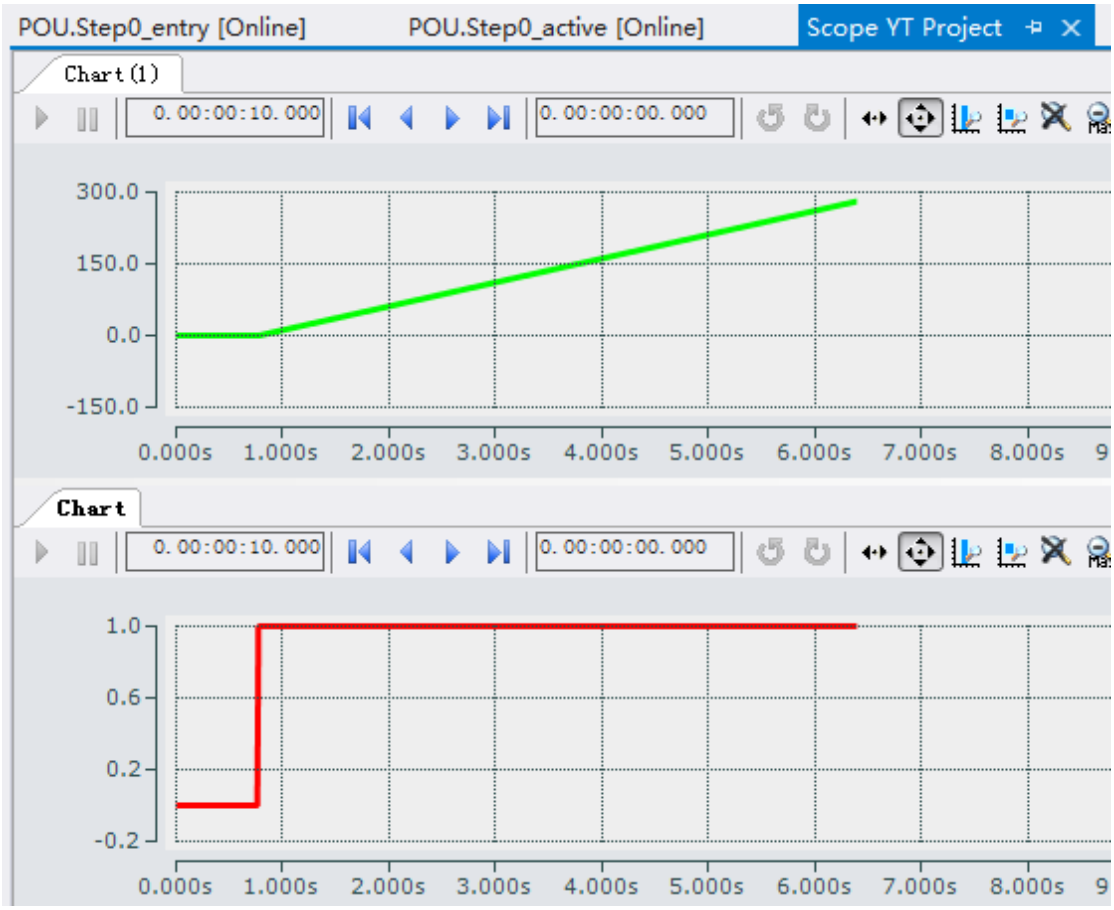
SFC 主程序：



程序效果：

结合 TwinCAT 3 自带的 Scope View，可以观察到

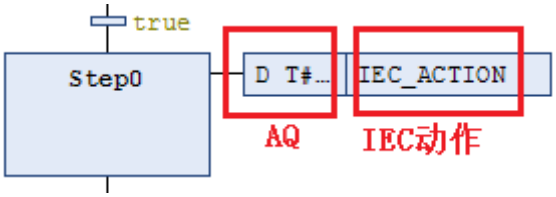
参数 A 每个循环周期都会从 0 开始累加；
参数 B 会随着程序的多次执行，而不断累加



2. IEC 动作块

基本概念：

一个单步可以注册多个动作块，与一个动作块一起登记的动作个数没有上限。



动作块包括了：AQ：动作限定符；IEC actions：IEC 动作；

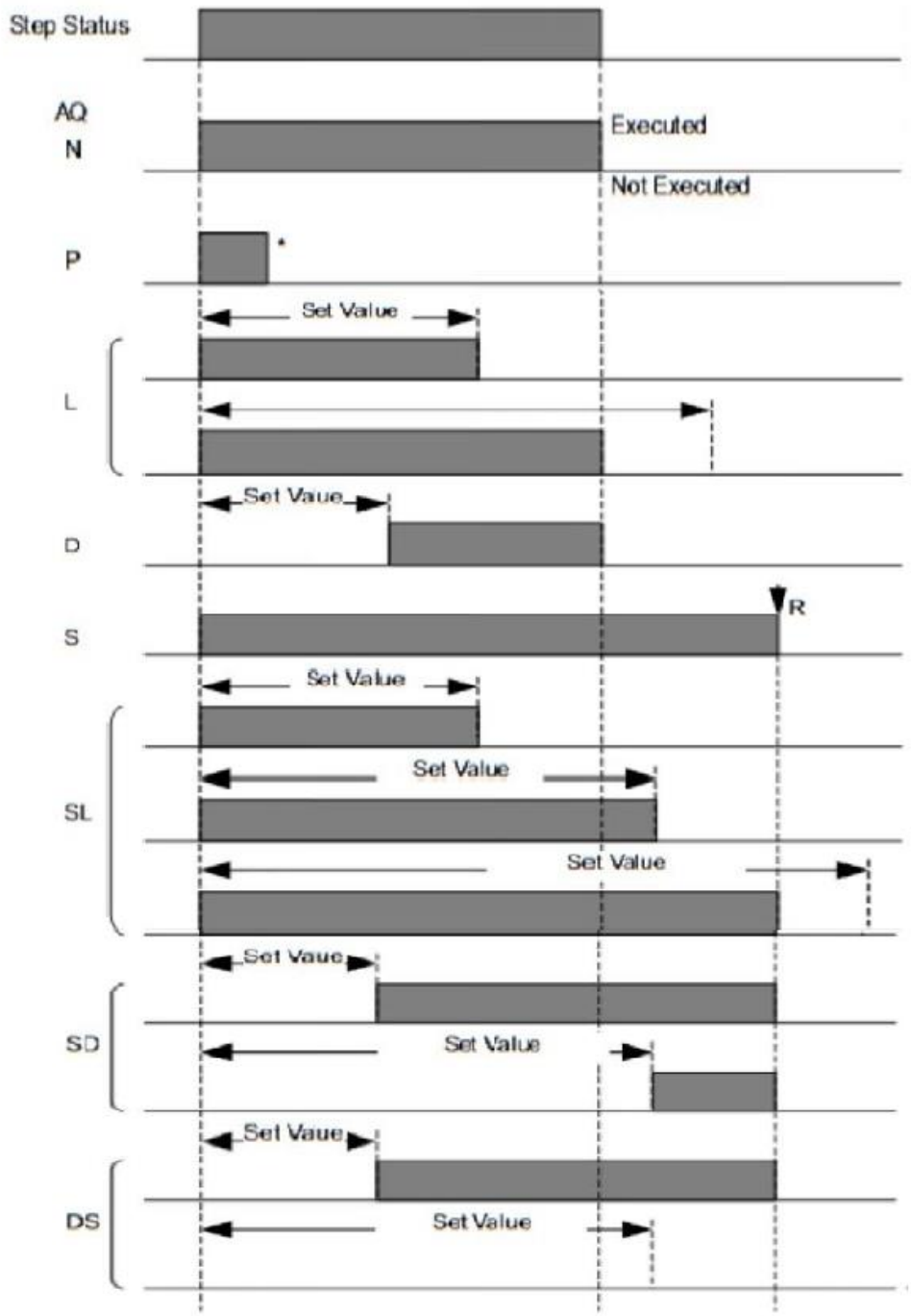
应用场合：

在执行 Step 步程序的同时，执行另一段或多段程序，并且彼此之间存在着一定的时间关系，此时就可以添加动作块

AQ 动作限定符功能介绍：

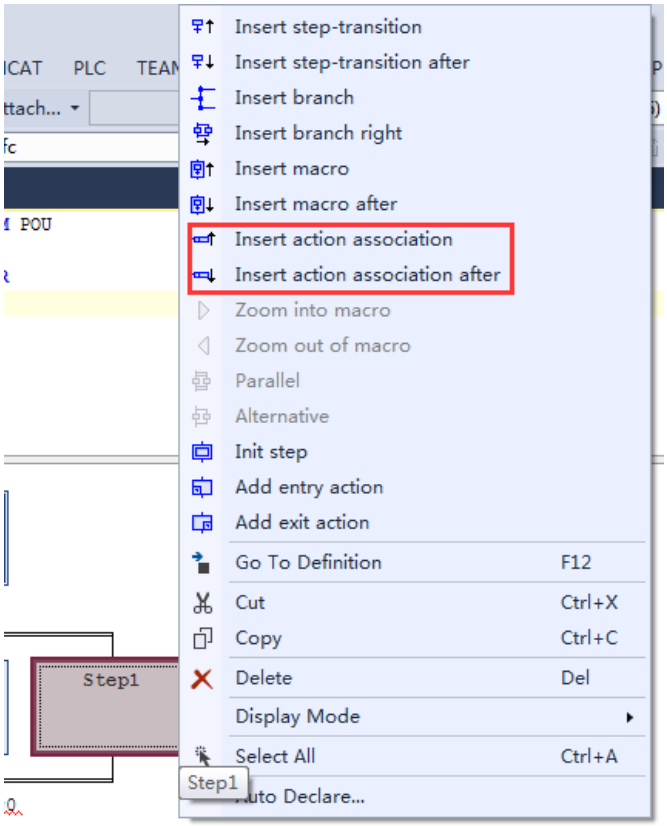
AQ 符号	名称	功能	是否需要设置定时器
D	延时	从步活动开始，过去用户指定时间后动作执行。如果在用户指定时间到期之前步变为非活动，动作将不执行。	是
DS	延时置位	步活动后直到用户指定时间到达，动作开始执行，步在用户指定延迟时间到期之前变为非活动，动作将不操作。要取消执行，使用 R 限定符。	是
L	延期执行	当步活动后，动作执行直到用户指定时间到达。如果步变为非活动，动作执行将终止。	是
N	默认	动作执行跟步活动时间一样长。	
P	脉冲	若最终扫描功能使能，当步活动时动作执行 2 个扫描周期。 若最终 扫描功能非使能，当步活动时，动作执行 1 个周期。	
R	复位	当步变为活动，且他的动作是由“S、SL、SD 或 DS”限定符，指定动作停止并复位。如果动作正在执行，但不是由“S、SL、SD 或 DS”限定符执行，动作只是复位（不停止）。	
S	置位	步活动时动作执行 ，即使步变成非活动也持续执行，要取消执行，使用 R 限定符。	是
SD	置位延时	步活动后直到用户指定时间到达，动作开始执行，即使步变成非活动也持续执行，要取消执行，使用 R 限定符。	是
SL	置位限制	步活动时动作执行，直到用户指定时间到期。与“L”不同的是，即使步变成非活动也持续执行，要取消执行，使用 R 限定符。	是

AQ 动作限定符的动作时序图：

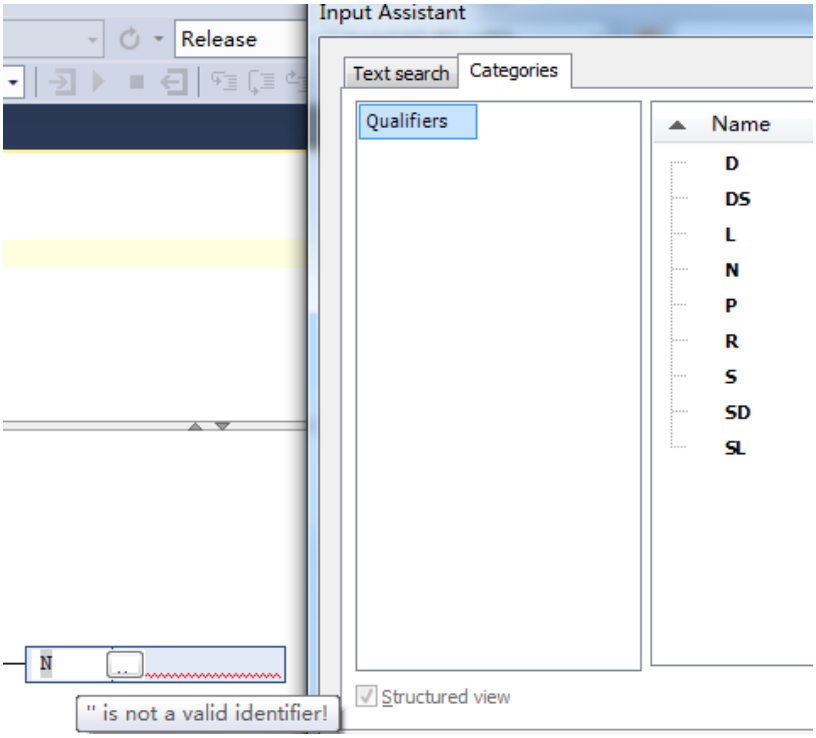


创建步骤：

（1） 选中要创建 Action 的 step，然后右击，从弹出菜单栏中找到 Insert action association(after)单击添加 Action。

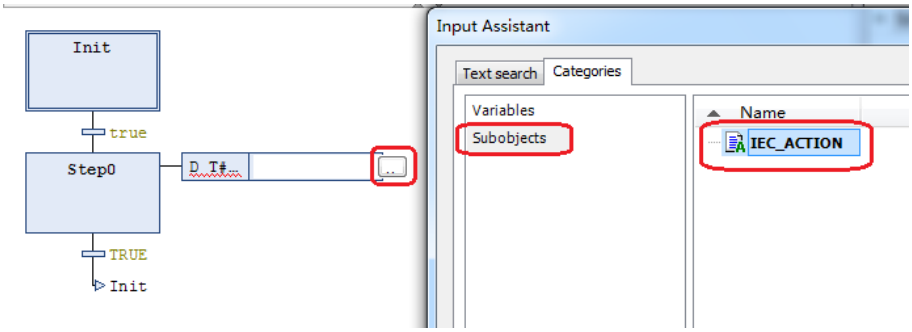


（2） Action 左侧框是动作限定符，默认是 N，如需更改可单击，再单击 ...，即可弹出对话框 input assistant，选定需要的动作限定符。

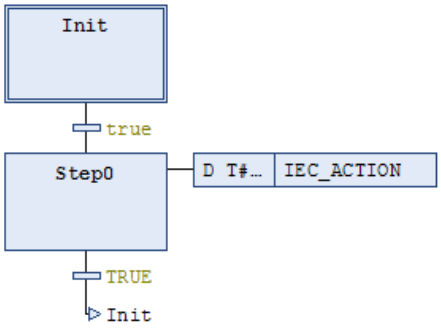


（3） 对于需要设定时间的 AQ，在限定符后输入 TIME 格式的时间，例如 T#2s

（4） 在 PLC 程序下创建一个 Action，并且添加到对应的 IEC 动作



（5） 这样就完成了 1 个 IEC 动作块的创建



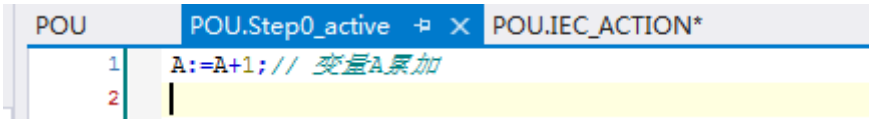
注意：

TwinCAT 2 软件中需要手动添加 System.Lib 后，才能使用 IEC 动作块

TwinCAT 3 软件中默认添加 TC2_System，所以可以直接使用 IEC 动作块

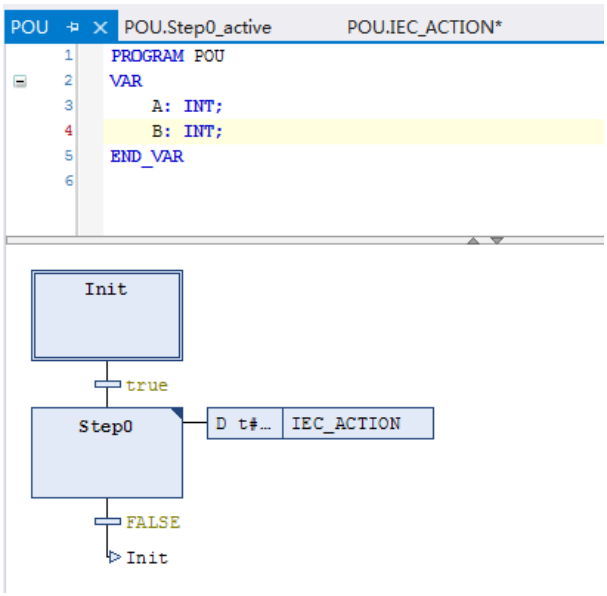
使用举例：

创建 Step 步动作：



创建 IEC Action 动作，AQ 为 D T#2s：

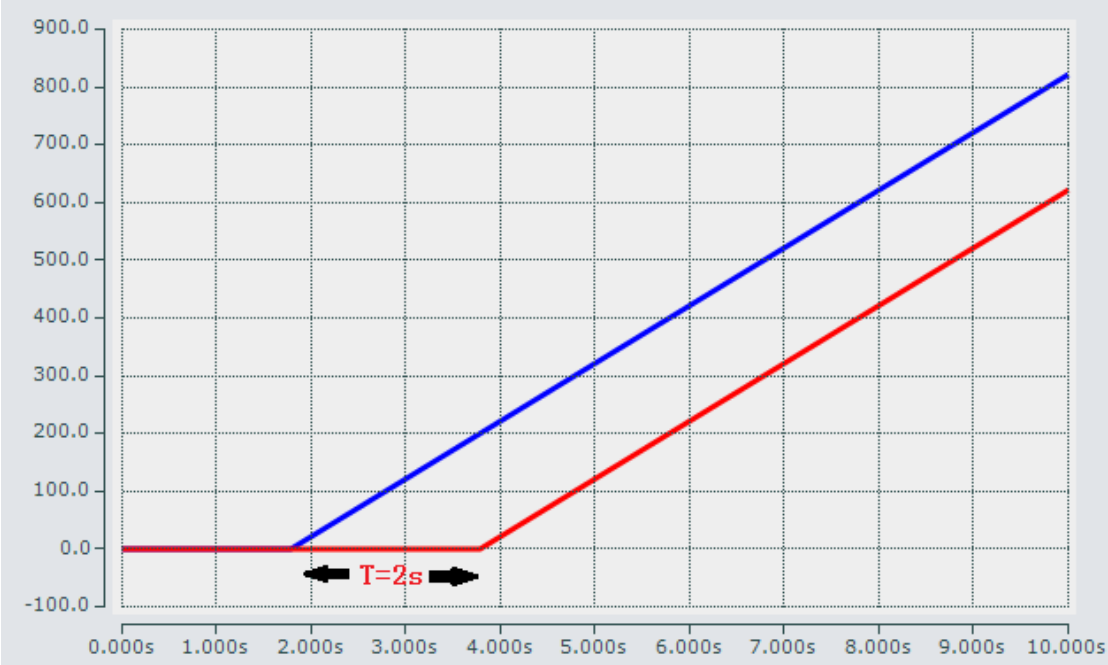
SFC 主程序：



程序效果：

结合 TwinCAT 3 自带的 Scope View，可以观察到：

变量 B 相对于变量 A 晚累加 2s，与设置的 AQ 和延迟时间一致



3. SFC 标志位

基本概念：

SFC 编程中，已经定义好了一部分的标志位，而且这部分标志位已经具有特定的功能。这些标志位的数据类型被定义在系统功能库，用户只需要启用对应 SFC 标志位，即可在程序中监控 SFC 程序的执行状态

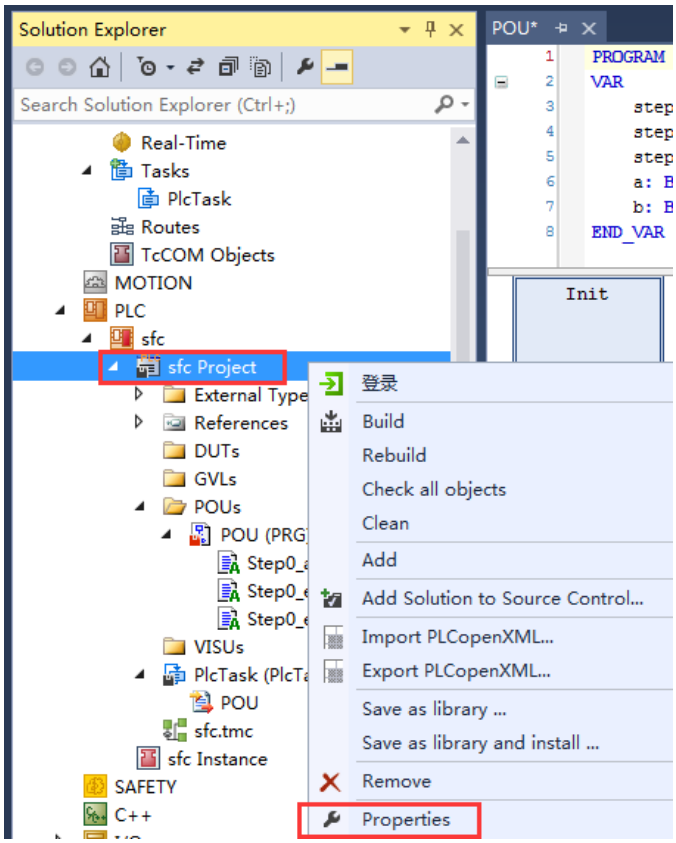
应用场合：

执行程序过程中，希望对程序进行初始化、暂停或查看是否有报错、当前执

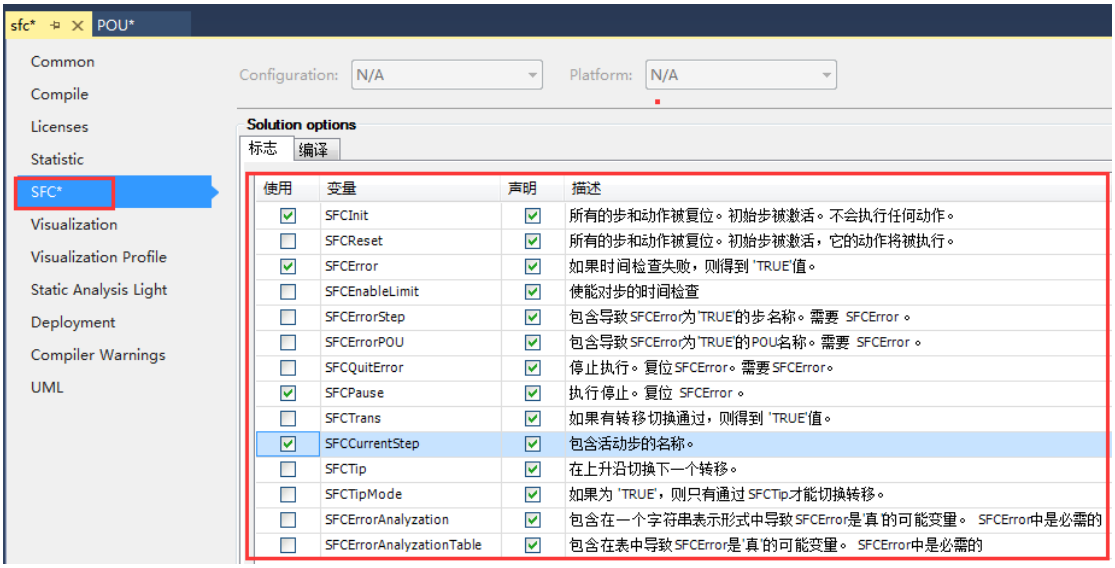
行步时，就可以使用这些已经定义好的 SFC 标志位，进行程序监控。

创建步骤：

（1） 右键 Project，选择 Properties 属性，在属性窗口中选择 SFC



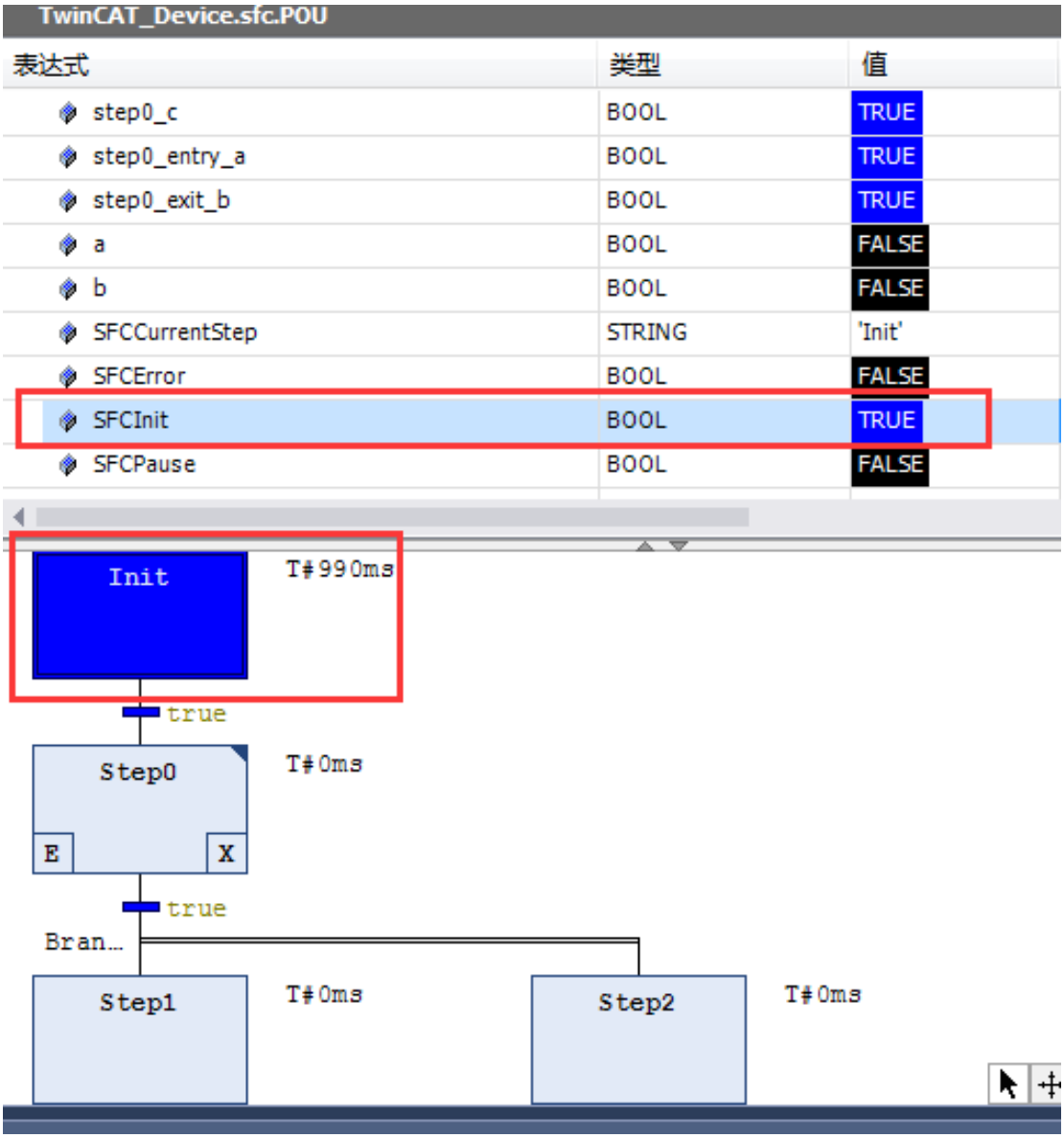
（2） 确认查看标志位是否申明，并选择启用所需要使用的标志位



以下介绍几个常用的标志位：

（1） SFCInit (Bool)

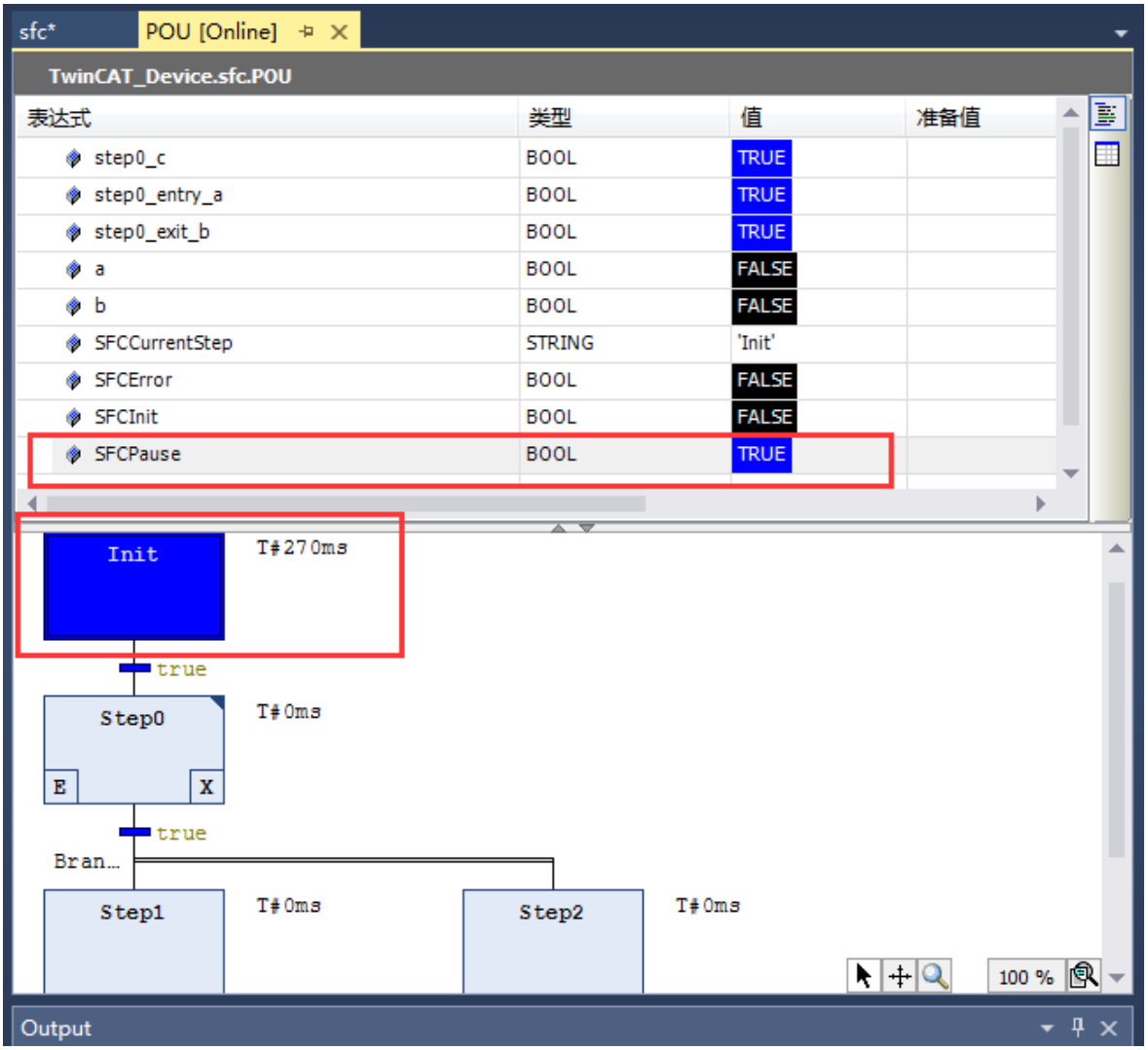
如果该标志位置为 True，在连续功能图表将会被返回到 Init 步。初始步将保持在活动状态。为了回到正常的进程，SFCInit 必须被置为 False。



（2） SFCPause (Bool)

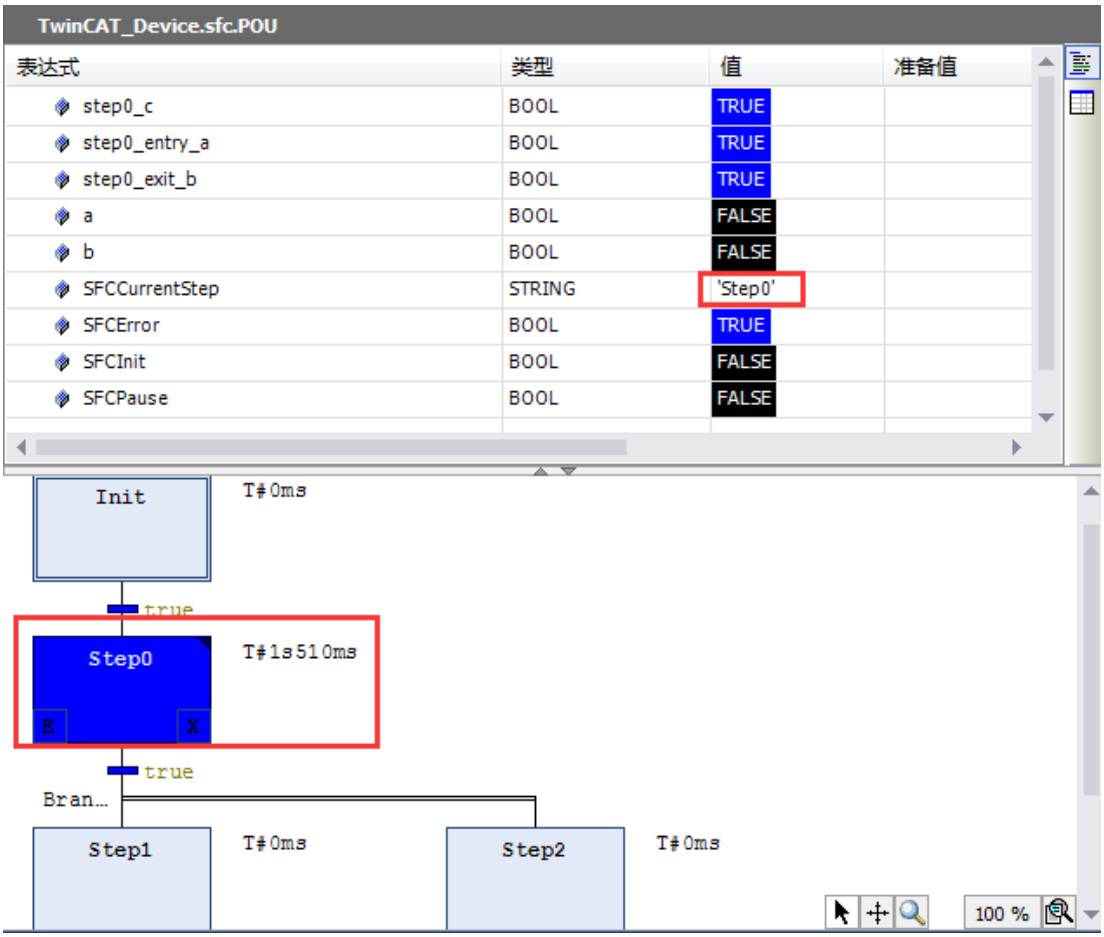
一旦标志位为 True，SFC 图表的触发就被停止。

例如下图所示：当触发程序运行过程中该标志位置为 True，则该程序停止；置为 False，则该程序继续运行。



（3） SFCCurrentStep (String)

该标志位储存一个当前为活动状态的步的名字，例如下图所示：



注意：

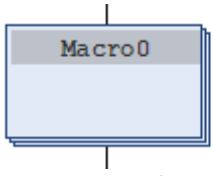
TwinCAT 2 软件中 SFC 标志位需要用户在本地或者全局变量自行申明，并变量名与标志位拼写保持一致才能使用 SFC 标志位；

TwinCAT 3 软件中默认已经申明，需要在 PLC 属性中进行勾选启用。

4. 宏（Macro）

基本概念：

宏程序用黑色粗体框表示，并包含宏程序的名字。如下图所示：



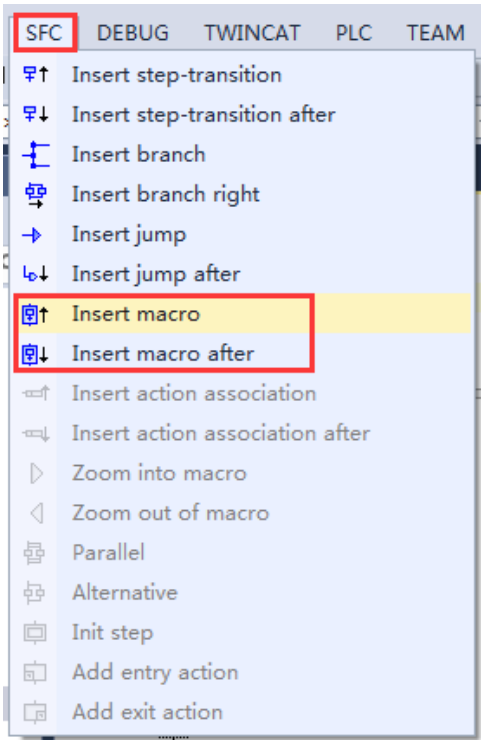
宏包括了部分的 SFC 图表，这部分图表在主程序编辑区不直接出现，进程流不会因使用宏程序被影响。

应用场合：

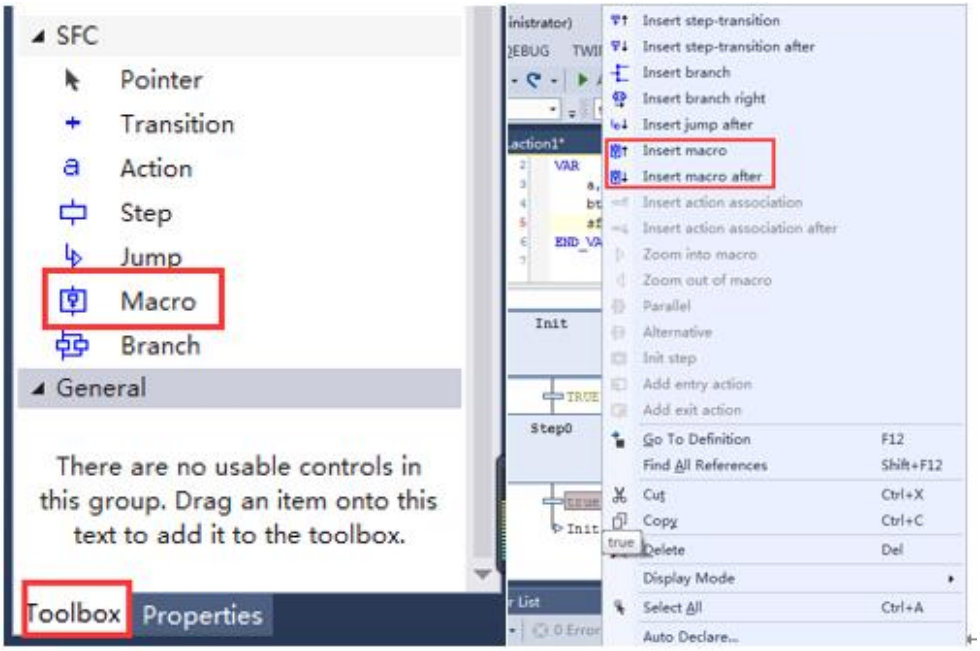
需要对 SFC 若干个连续的 Step 步和 Transition 转移条件进行封装，将冗长的程序进行隐藏，使得程序更简明。

创建步骤：

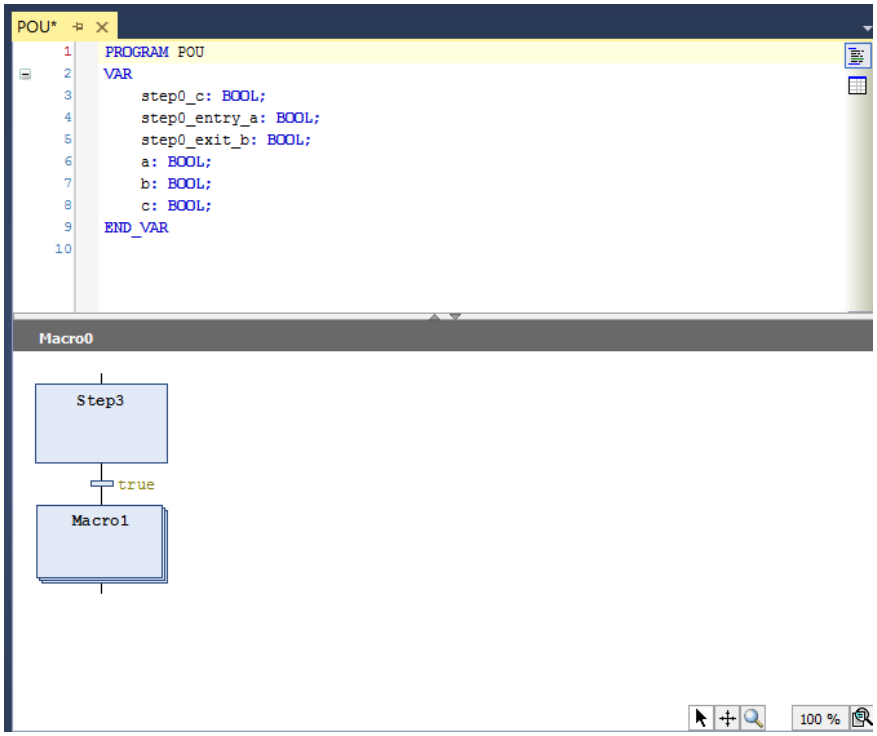
（1） 选中需要创建 macro 相邻的 Step 或者 Transition，然后点击菜单栏 SFC，找到 Insert macro 或者 Insert macro after；



也可以从右侧的 Toolbox 或右键进行添加；

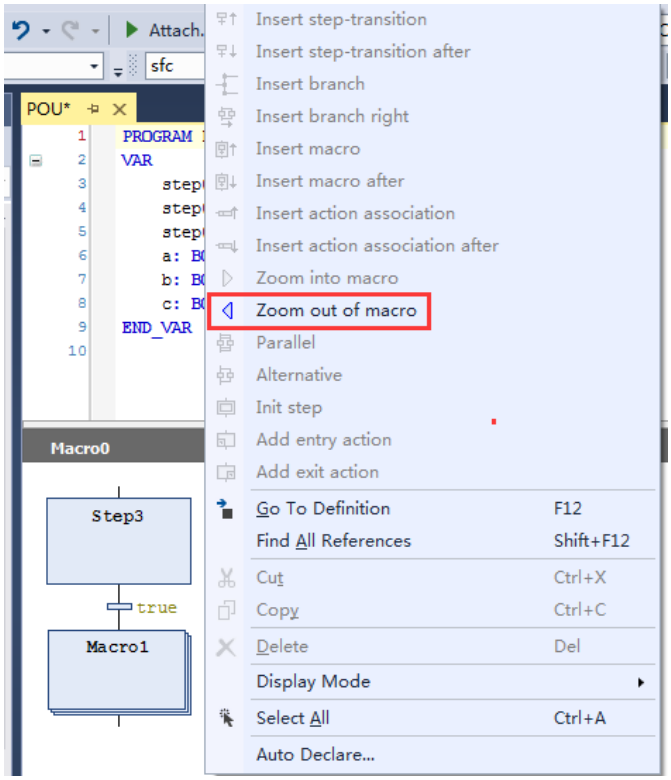


（2） 选中 Macro，找到 SFC 菜单栏，在下拉框中找到“Zoom into macro”单击。进入 Macro 编程界面：



（3） 在宏中可以添加与主程序相同的元素，例如 Step 步、Transition 转移条件及各种分支结构等

（4） 完成编辑后，如需退出 Macro，在编辑区域空白区域右击“Zoom out of macro”退出 macro 编辑。



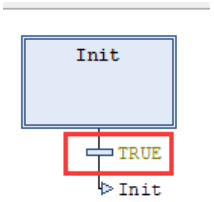
（5） 这样就完成了宏命令的编写

五、SFC 综合使用举例

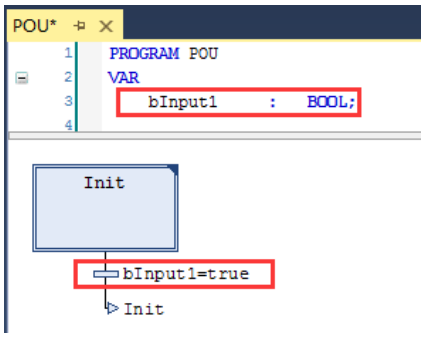
本举例可以在虚拟学院上下载，地址：
http://tr.beckhoff.com.cn/mod/folder/view.php?id=955

1. 完成一个简单程序的编写，将 bInput1 赋值给 bOutput1，即：bOutput1:= bInput1。（利用 step 和 transition）

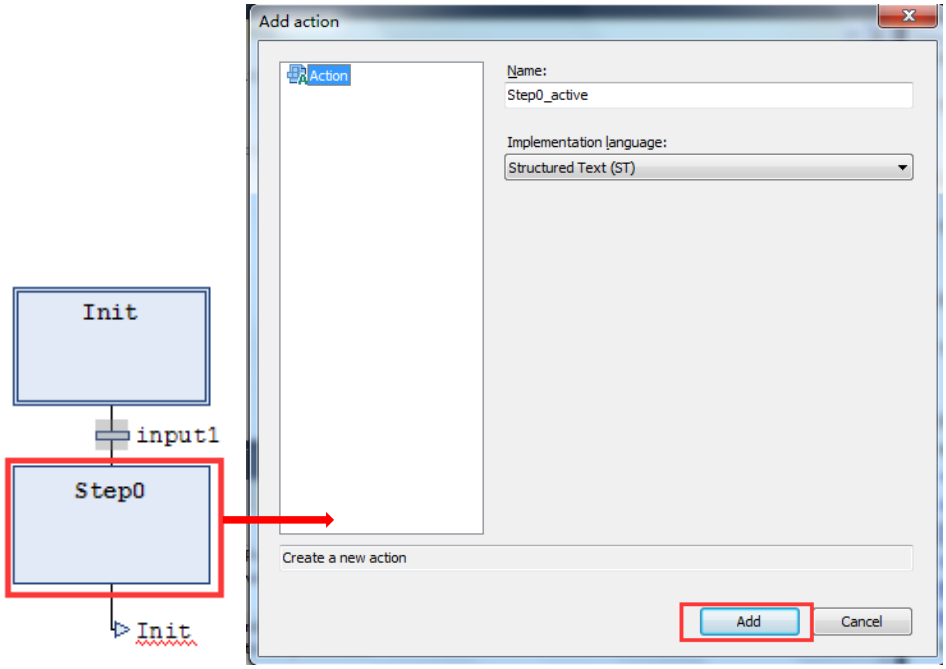
（1） 新建项目。



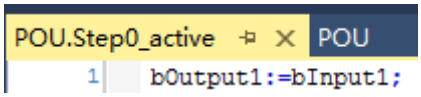
（2） 将 True 更改成 bInput1 变量，并将类型设置成 bool 型。



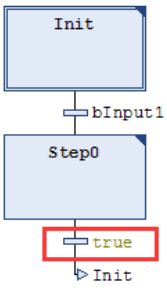
（3） 创建一个普通步：从右侧 Toolbox 中拖一个 step 在转移条件 bInput1 下，生成 step0 并双击，弹出对话框单击 Add。



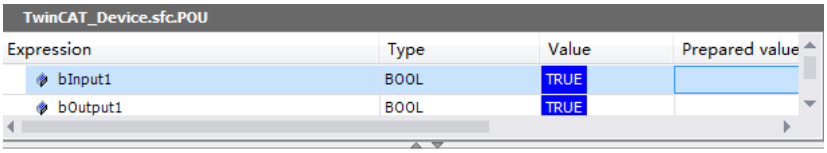
（4） 在程序编辑空白界面中输入：bOutput1:= bInput1；完成后回车，将 bOutput1 变量设置成为 bool 类型。

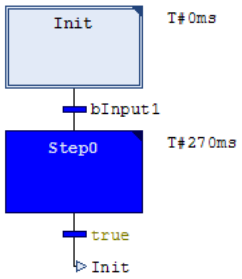


（5） 在 step0 下创建一个转移，并将其赋值为 true。



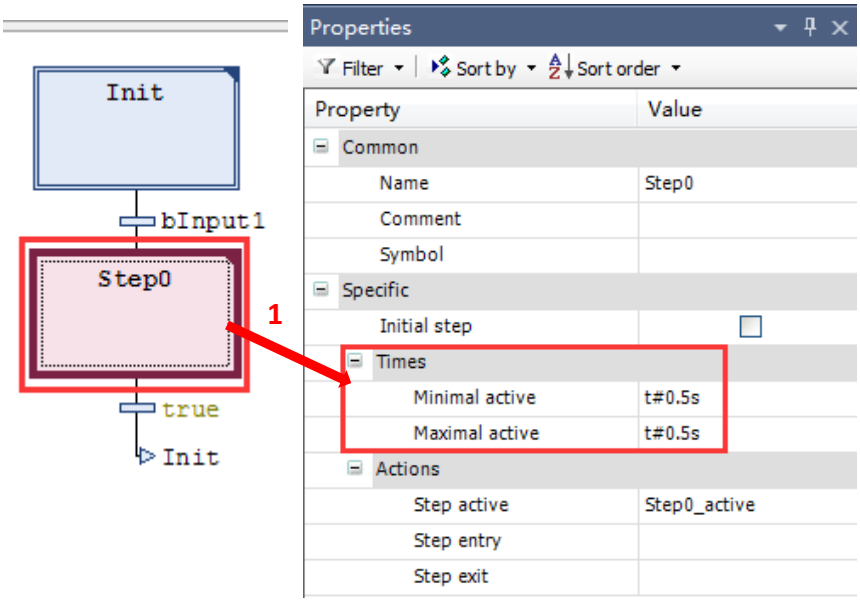
（6） 以上就完成了简单程序的编写，编译查看是否报错，如无报错，登入并运行程序。将 bInput1 置为 true 后，bOutput1 也立即也置为 true。



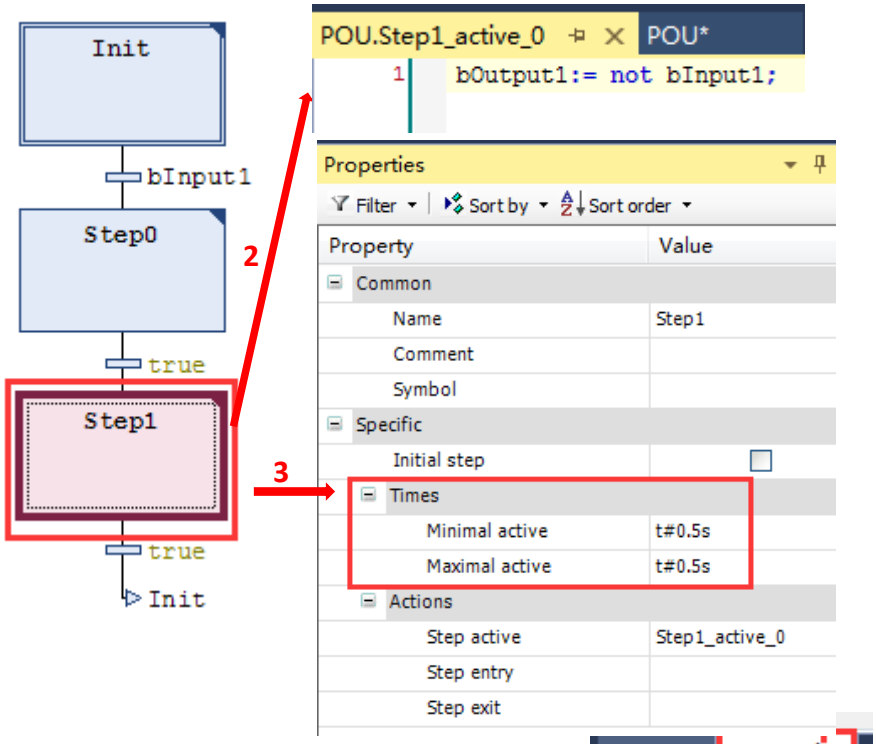


2. 实现一个周期为 1s 的 50%占空比脉冲，即：将 bOutput1 变量设置成以周期为 1s 的 50%占空比脉冲。（利用 step、transition 以及定时功能）

（1） 选中 step0，找到右侧的 Toolbox Properties 选项卡，将 Times 的 Minimal active 和 Maximal active 都设置为 T#0.5S；

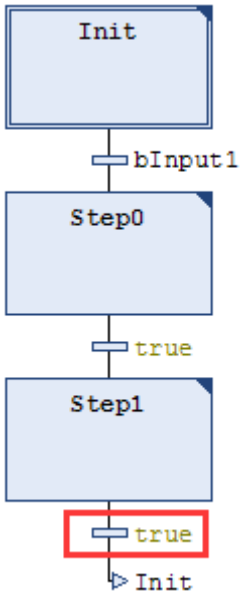


（2） 创建一个普通步 step1，并双击；弹出对话框单击 add；在程序编辑空白区中输入 bOutput1:= not bInput1;

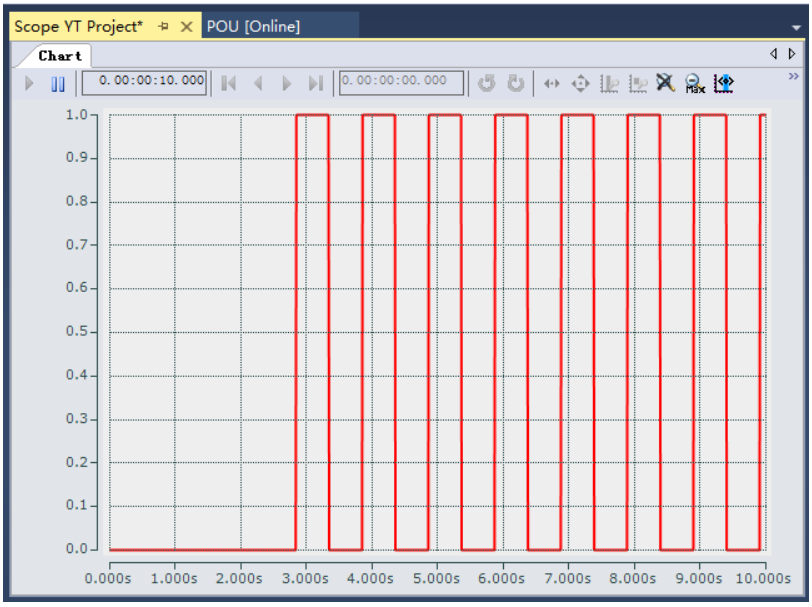


（3） 回到主界面，选中 step1，找到右侧的 Toolbox Properties 选项卡，将 Times 的 Minimal active 和 Maximal active 都设置为 T#0.5S；

（4） 插入转移条件，并赋为 true；

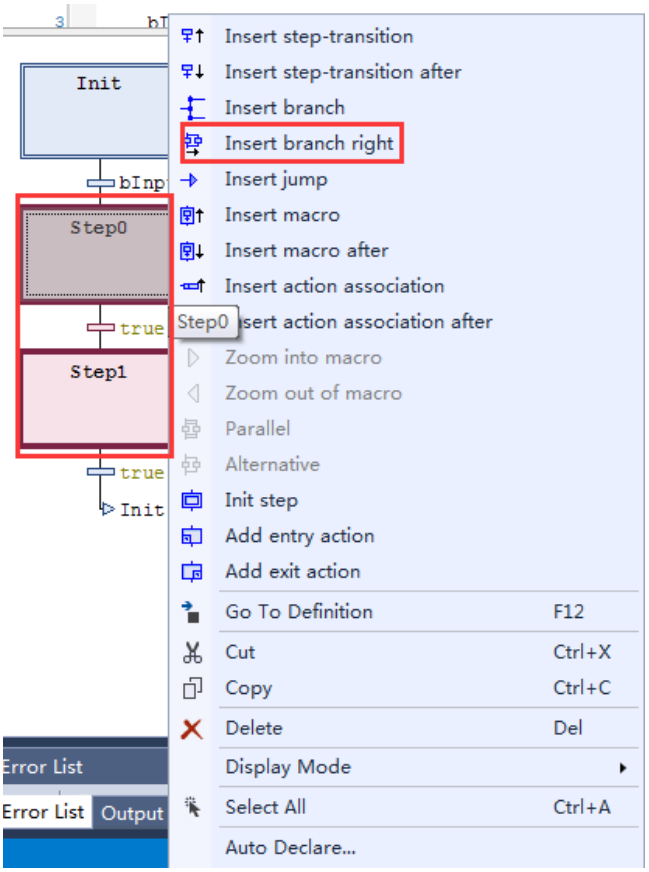


（5） 以上就完成了简单程序的编写，编译查看是否报错，如无报错，登入并运行程序。使用 scope view 来监控 bOutput1 值的变化。



3. 实现同时输出周期为 1s 的 50%占空比脉冲和余弦函数脉冲，即：余弦函数脉冲 lOutput_cos:=COS(pi*0.2*lInput_t);;

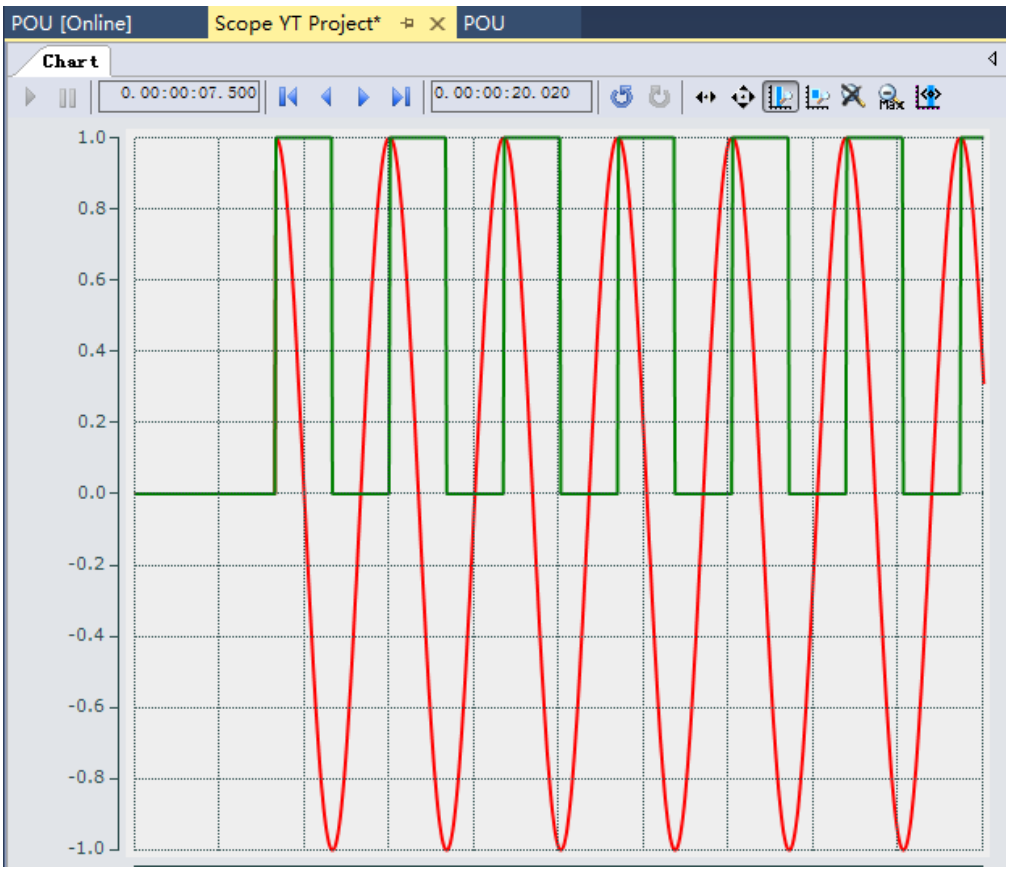
（1） 选中 step0，step1，及两步之间的“true”，然后右击，单击 inset branch right；



（2） 双击创建出的空白 step2，弹出对话框单击 Add，并在程序编辑区输入
lOutput_cos:=COS(pi*0.2*lInput_t);
lInput_t:=lInput_t+0.1;
并将 lOutput_cos 和 lInput_t 两个变量设置成为 LREAL 类型。

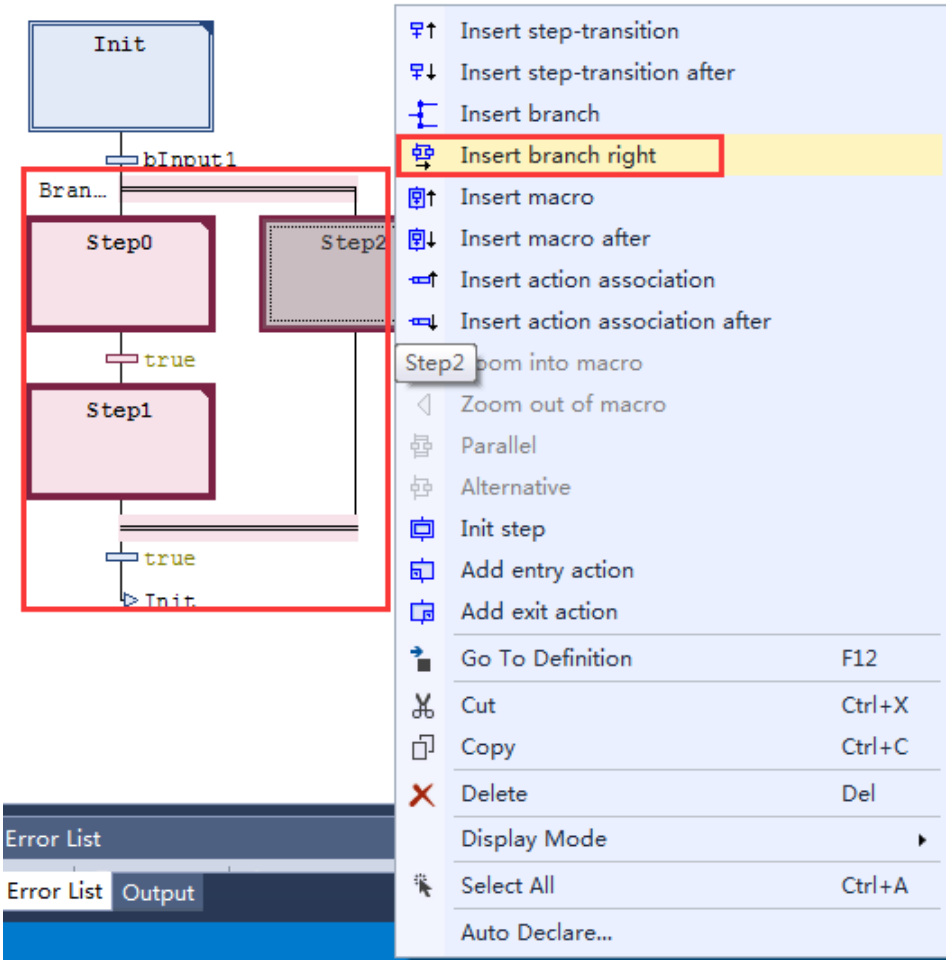
```
POU.Step2_active    POU*
1   lOutput_cos:=COS(pi*0.2*lInput_t);
2   lInput_t:=lInput_t+0.1;
```

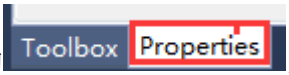
（3） 以上就完成了简单程序的编写，编译查看是否报错，如无报错，登入并运行程序。使用 scope view 来监控 bOutput1 值和 lOutput_cos 值的变化。



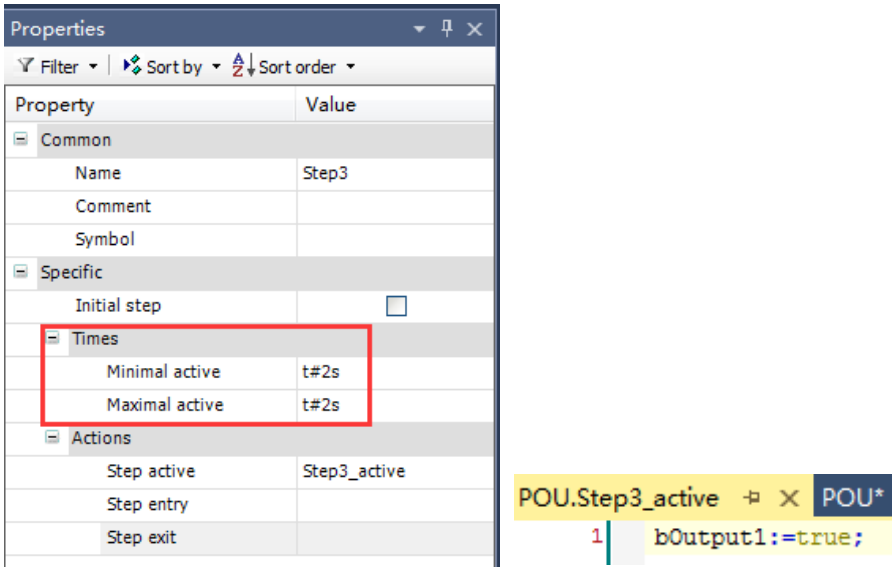
4. 实现利用变量的不同状态来选择执行相应的子程序。例如利用 bInput1 的 true 和 false，选择 bOutput1 不同的输出状态。当 bInput1 为 true 时，状态为原先的 50%占空比脉冲；当 bInput1 为 false 时，状态输出周期为 2s 的高电平脉冲。

（1） 选中 bInput1,step0,true,step1,step2，然后右击，单击 insert branch right;

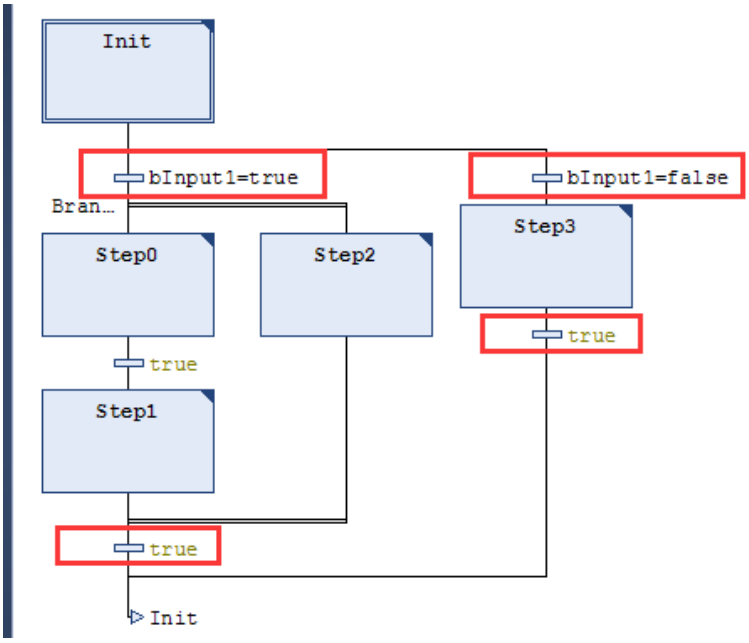


（2） 选中 step1，找到右侧的 

选项卡，将 Times 的 Minimal active 和 Maximal active 都设置为 T#2S；双击新创建出的 step3，弹出的对话框单击 Add，并在程序编辑区输入 bOutput1:=true;



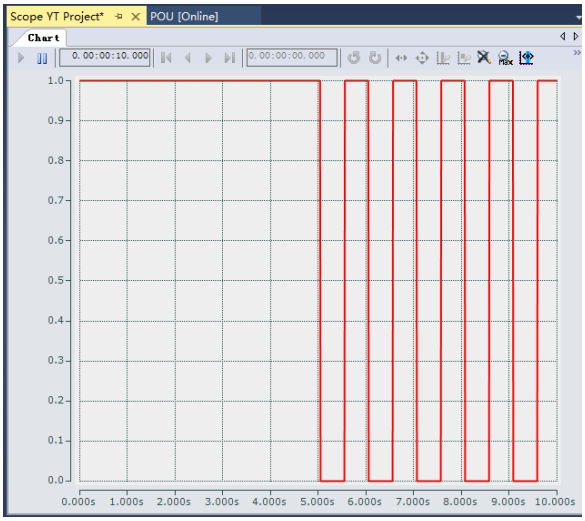
（3） 修改相应的转移条件，如下图所示；



（4） 以上就完成了简单程序的编写，编译查看是否报错，如无报错，登入并运行程序。使用 scope view 来监控 bOutput1 值的变化。

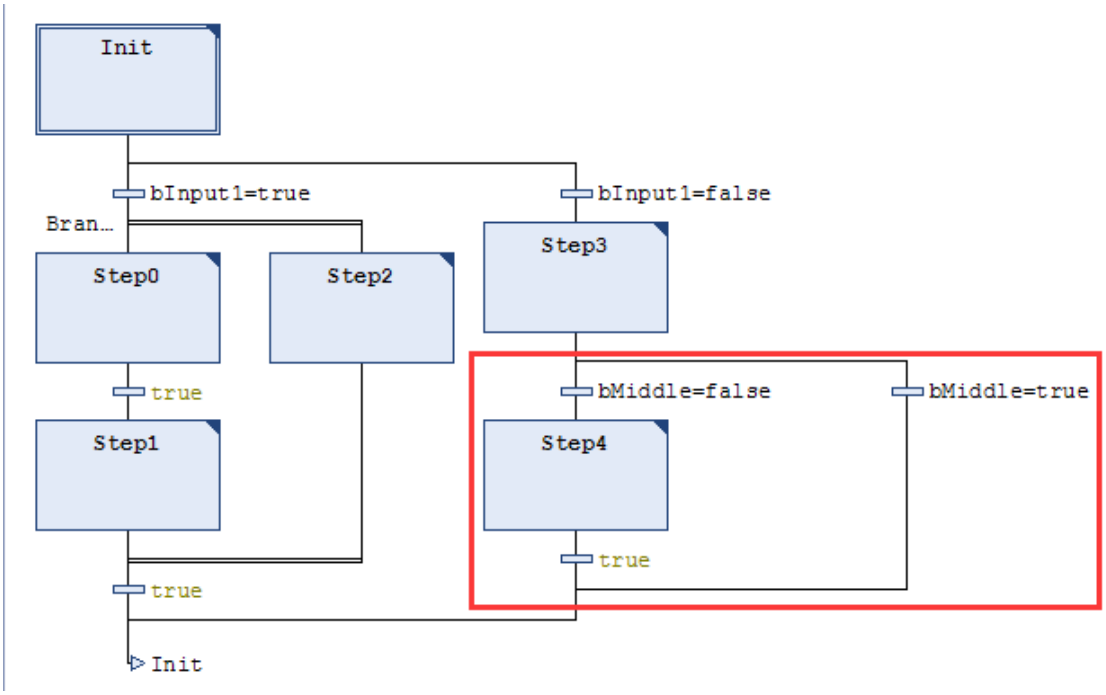
1) 运行程序后，初始 bInput1 为 false，选择分支进入右分支结构中。通过 scope view 监控到 bOutput1 值，如下图所示：

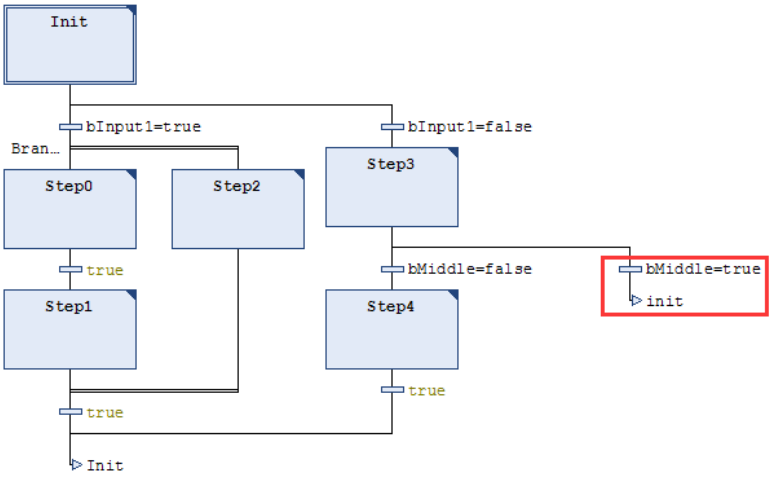
2) 当将 bInput1 置为 true 后，在运行到下一个任务周期时，选择分支进入左分支结构中。通过 scope view 监控到 bOutput1 值，如下图所示：



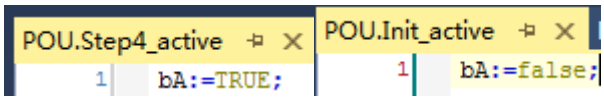
5. 实现控制不同的条状条件控制输出结果不同。在上述基础上，bInput1 选择分支（右分支）即：当 bInput1 为 false 时，bOutput1 状态输出周期为 2s 的高电平脉冲后；变量 a 受到转移条件 bMiddle 的影响，当 middle 为 false 时，a 会输出周期为 1s 的高电平；当 middle 为 true 时，跳转回 init 步。

（1） 将 step3 下方的 true 删去，并在添加在 step4，以及对 step4 创建右分支，同时添加下图示转移（bMiddle 变量为 bool 类型）。选中转移“bMiddle=true”插入后跳转，并设置跳转到 Init



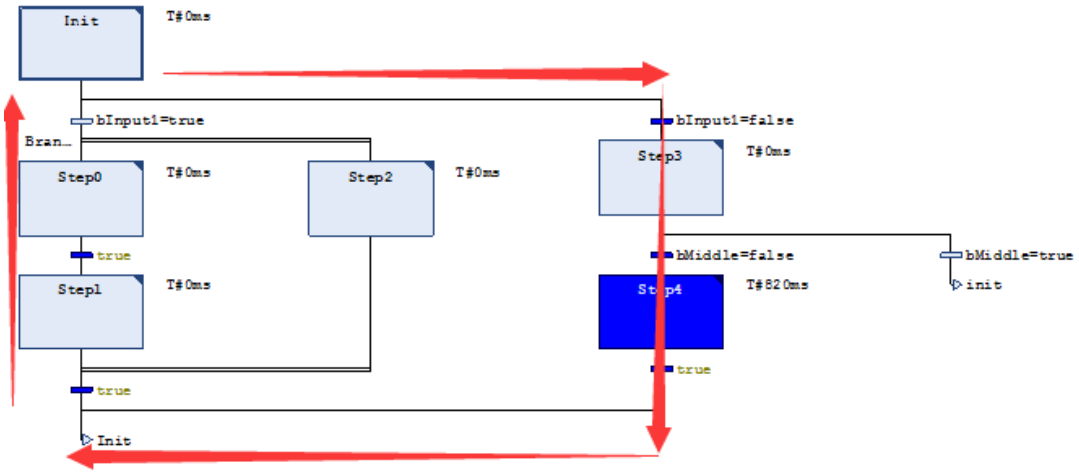


（2） 双击 step4，进入程序编辑区输入 bA:=true；双击 Init，进入程序编辑区输入 bA:=false;

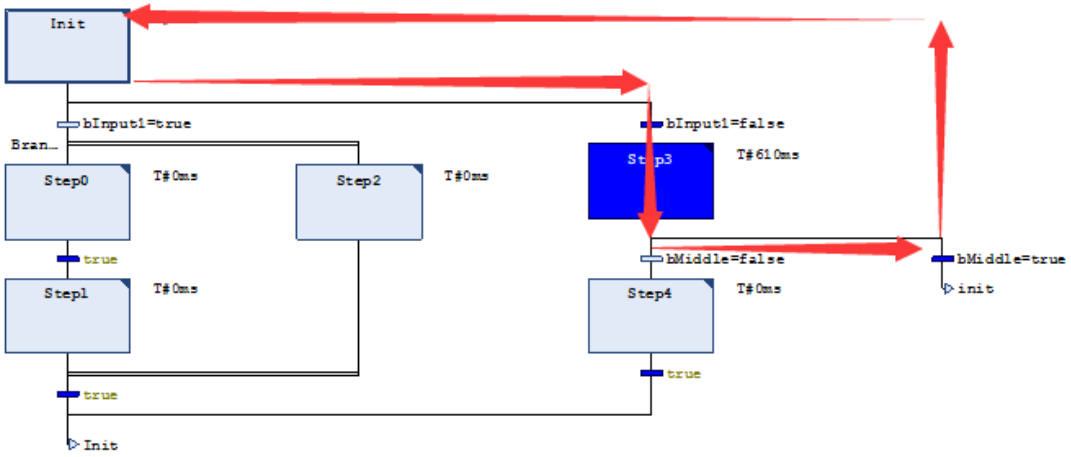


（3） 以上就完成了简单程序的编写，编译查看是否报错，如无报错，登入并运行程序。

1) 当 bInput1 为 false，bMiddle 为 false 时，程序运行方向如下：



2) 当 bInput1 为 false，bMiddle 为 true 时，程序运行方向如下：

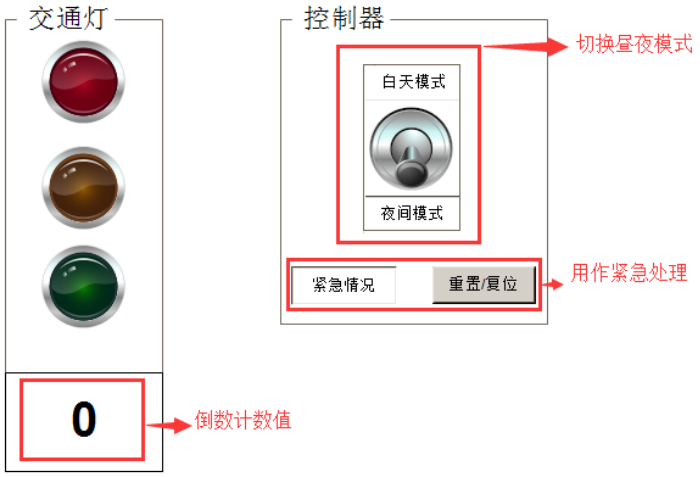


六、交通灯实例说明

本举例可以在 FTP 服务器上下载，地址：
http://tr.beckhoff.com.cn/mod/folder/view.php?id=955
实例要求：设计一款交通灯，实现昼夜两种模式显示，同时在出现紧急情况下，要求能瞬间切入黄灯闪烁模式。

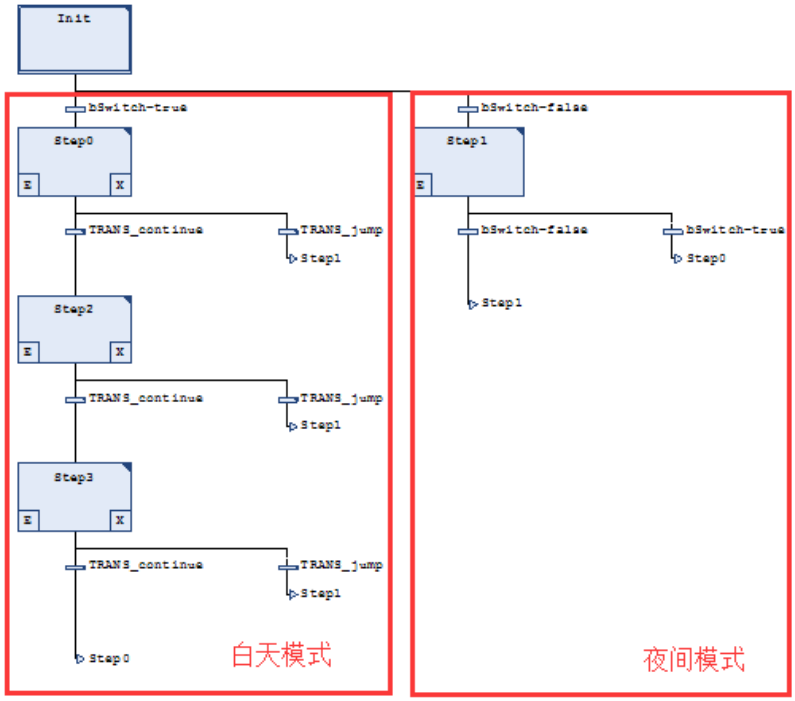
白天模式：在此模式下，按照“绿灯常亮 8s→黄灯间隔闪烁亮 5s→红灯常亮 13s→…”循环显示，并能显示出倒数计数值；

夜间模式：在此模式下，保证正常结束白天模式，例如当红灯亮时，以红灯常亮 13s 作为结束，然后黄灯持续间隔闪烁；



编写提示：

1. 采用分支结构将整个框架分配成白天和夜间两个分支；



2. 在初始步内分别对绿灯、红灯、黄灯，以及昼夜转换按钮和倒数计时值进行复位和清零。
3. 白天模式时候，循环是在“step0→step2→step3→…”；夜间模式，循环是在“step1→step1→…”。
4. step0, step2,step3 分别是对绿灯 8s、黄灯 5s、红灯 13s 进行操作，运行时长利用“property”中“times”进行设置，在如下图所示。复位、赋初值、清零等功能在输入输步的完成。

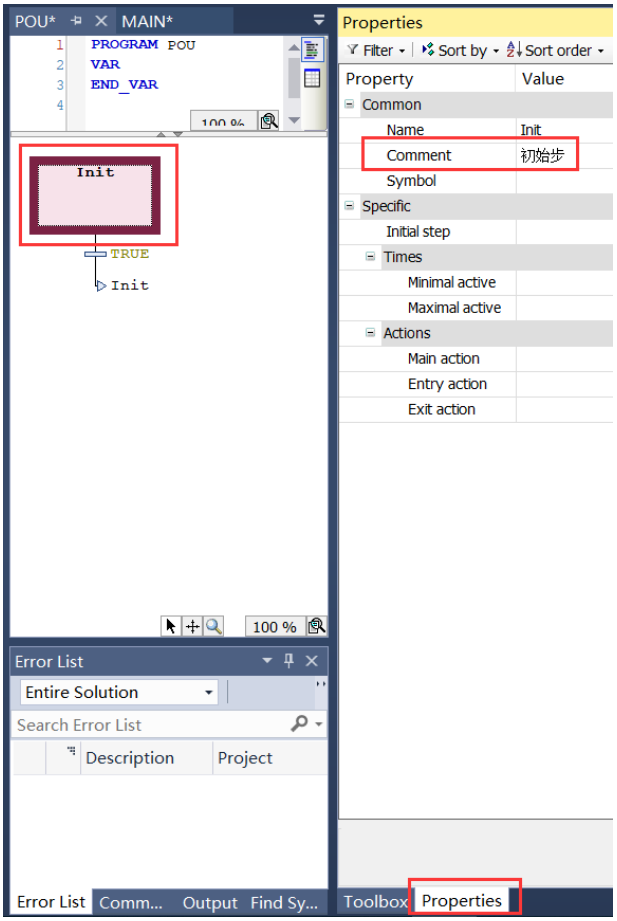
5. step1 用来做夜间模式的黄灯闪烁，也是切入紧急状态后进行黄灯闪烁。

注：具体提示及编程结构可参看样例程序中的注释。

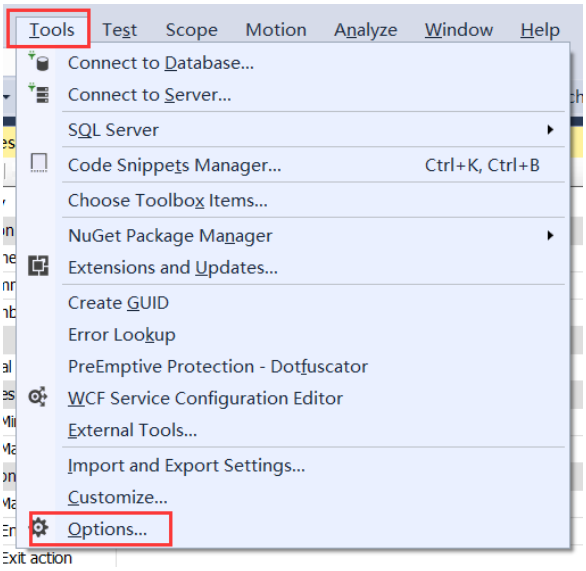
七、常见问答

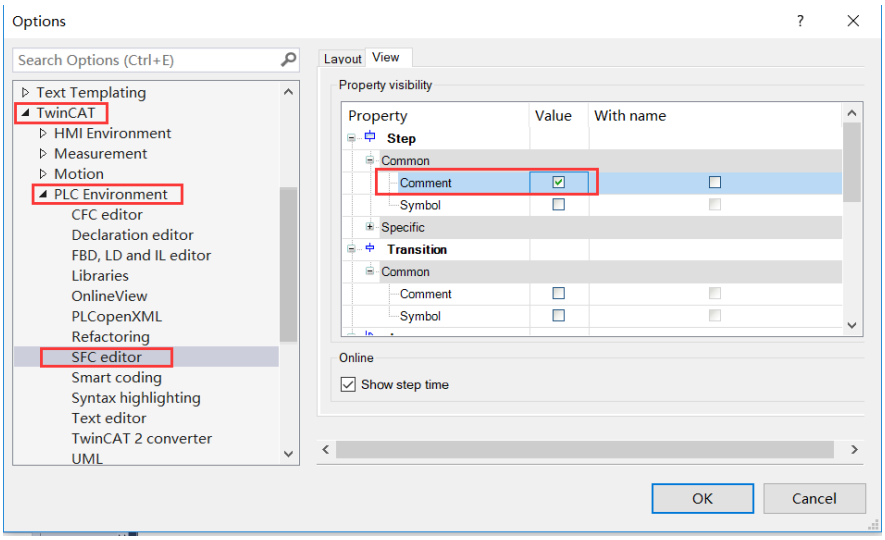
1. SFC 中的备注如何添加和显示？

以 step 为例，比如说需要在 Init 步需要进行备注，备注信息为“初始步”。方法如下：首先选中 init 步，然后在 properties 中 comment 写入备注信息；

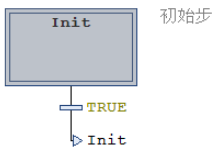


写好后，发现 comment 信息未在程序区显示，需要显示出来的话，需要进入 Tools-options 中调用出来的，





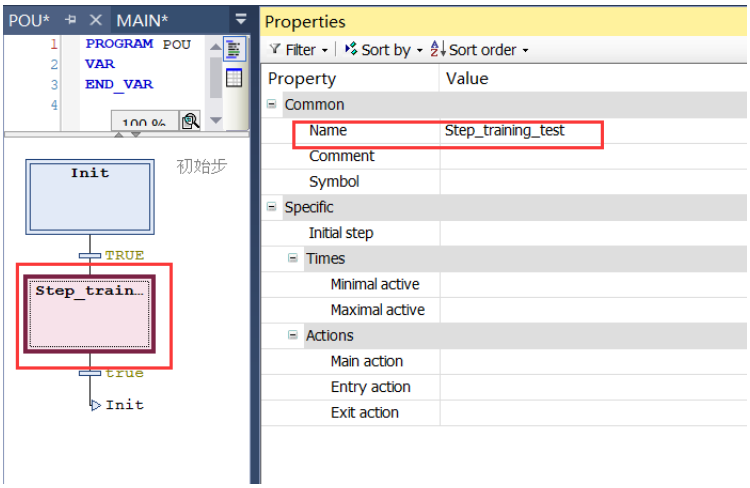
此时程序区可以看见 comment 信息已经显示出来



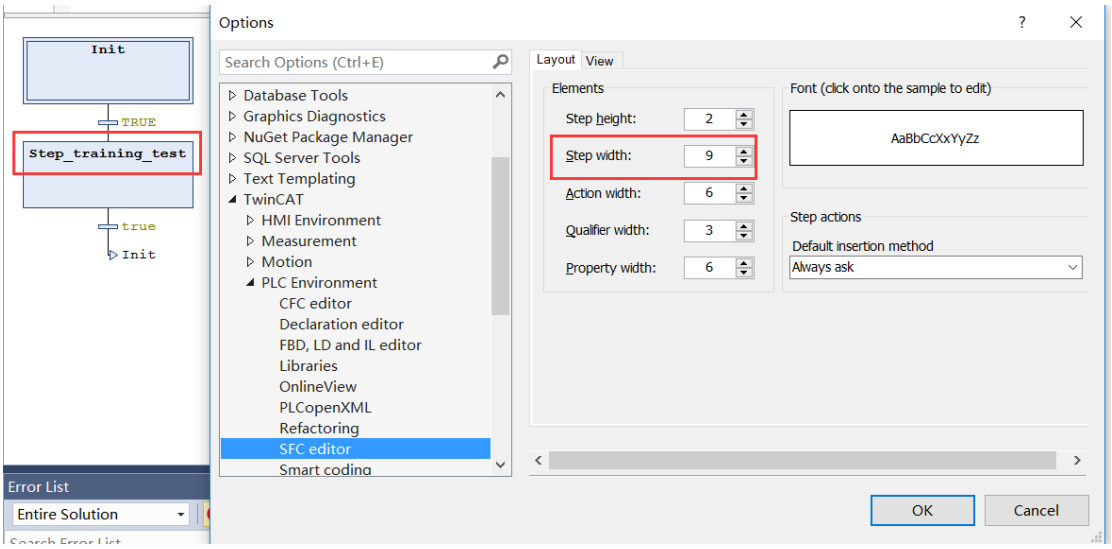
同理，如果需要显示 trans，jump,macro 等的备注也去 options 中勾上相应的 comment

2. 在编程时，对 step 的步名做修改后，步名过长，无法完全显示怎么办？

假设遇见下图情况：



可以通过修改 Tools-options 中的 step width 步宽，如下图：



八、结束语

SFC 作为一种较新型的编程语言，市面上少有相关的编程学习书籍。在众多同事和客户的强烈要求下，倍福技术支持部编写了这本《TwinCAT SFC 编程入门》供各位读者学习参考。

本书基于 IEC61131-3 编程语言标准进行编写，使用倍福 TwinCAT 3 软件，可以作为有关电气控制技术人员学习 SFC 编程语言的入门级教材使用。本书编写过程中广泛听取多方意见，并吸取多项实际项目的优秀经验，经过反复斟酌后编写而成。

本书中的多个实例，配合相关说明，由浅入深地帮助各位读者学习掌握 SFC 编程语言。参考案例已经上传至对应网站供读者下载。在边学边做的过程中，读者可以充分体会到编程的乐趣。

由于编者水平所限，误漏欠妥之处在所难免，竭诚欢迎各位同行和读者批评指正，联系邮箱: Support@Beckhoff.com.cn

编者 梁霄 张立文
2016 年 3 月

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