

1. 前言

本文可以作为 S7-GRAPH 编程语言的使用参考资料，希望读者通过对本章的阅读，能够更快更好地学习 S7-GRAPH 编程语言。西门子提供了 S7-GRAPH 编程语言的详尽手册，在安装 S7-GRAPH 软件包后，通过点击 Windows 菜单 开始->Simatic->Documentation->English 可以阅读到名称为“S7-GRAPH - Programming Sequential Control Systems”的 PDF 手册。此手册共分为 15 个章节，其详细地讲解了 S7-GRAPH 编程语言。一切关于 S7-GRAPH 使用的问题请以此手册为准。

相对于其它编程语言，S7-GRAPH 有着它本身一些特性，一些独特的概念也比较难于理解。本文中将对 S7-GRAPH 手册中的部分内容作出额外强调与讲解，虽然这些讲解与手册有重叠之处，但希望通过这些讲解能够使用户更快地掌握 S7-GRAPH 的使用。在讲解内容之后，再列举出一些用户在实际使用当中经常会遇到的问题，希望能够对用户有所帮助。

相关手册地址连接：

S7-300 和 S7-400 的梯形图 (LAD) 编程

<http://support.automation.siemens.com/CN/view/zh/18654395>

S7-300 和 S7-400 的语句表 (STL) 编程

<http://support.automation.siemens.com/CN/view/zh/18653496>

使用 STEP 7 V5.3 编程

<http://support.automation.siemens.com/CN/view/zh/18652056>

S7-GRAPH V5.3 for S7-300/400

<http://support.automation.siemens.com/CN/view/zh/1137630>

2. 软件的基本信息

2.1. S7-GRAPH 简介

S7-GRAPH 具有以下特点：

- ✧ 适用于顺序控制程序
- ✧ 符合国际标准 IEC 61131-3
- ✧ PLCopen 基础级认证
- ✧ 适用于 SIMATIC S7-300 (推荐用于 CPU314 以上 CPU), S7-400, C7 and WinAC

S7-GRAPH 针对顺序控程序做了相应优化处理，它不仅仅具有 PLC 典型的元素（例如 输入 / 输出，定时器，计数器），而且增加了如下概念：

- ✧ 多个顺控器（最多8个）
- ✧ 步骤（每个顺控器最多250个）
- ✧ 每个步骤的动作（每步最多100个）

3. S7-GRAPH 基本概念与基本使用

3.1. S7 程序构成

作为 STEP7 的选项包，S7-GRAPH 软件在安装后，将被集成在 STEP7 中使用。S7-GRAPH 编程界面为图形界面，包含若干个顺控器。当编译 S7-GRAPH 程序时，其生成的块以 FB 的形势出现，此 FB 可以被其它程序调用，例如 OB1。

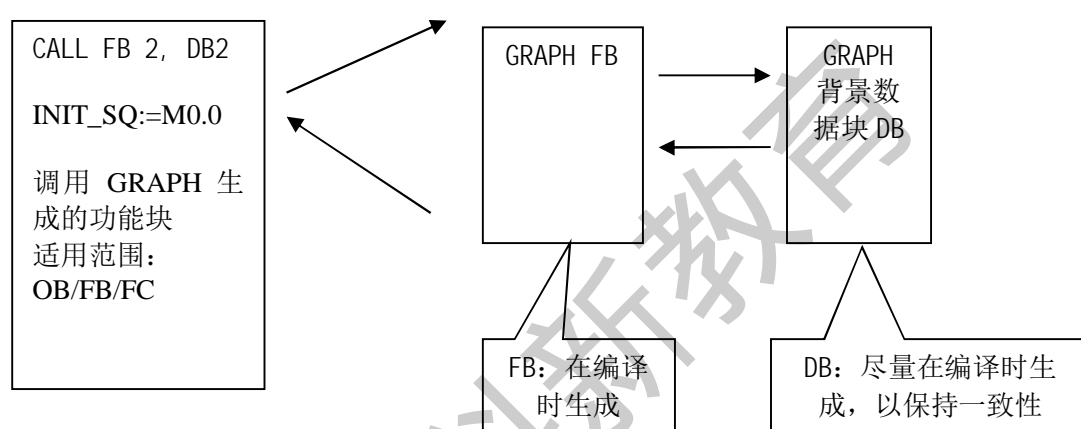


图 3-1: S7 程序构成

3.2. 用户界面

3.2.1. 生成新 S7-GRAPH 程序

首先，在 STEP7 当中生成一个新项目，用右键点击 Sourc 文件夹，插入一个新的 GRAPH Source, 如下图：

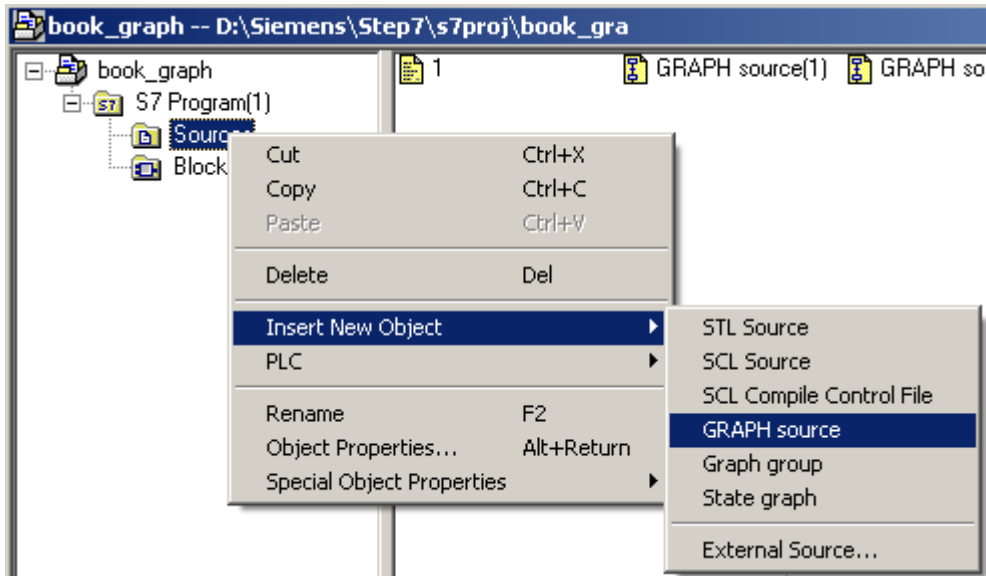


图 3-2: 生成新 S7-GRAPH 程序

注意：用户也可以在 Block 文件夹插入一个 GRAPH 格式的 FB，与插入源文件的区别，将在本文的后续部分详细讲解。

双击新生成的 GRAPH source 后，可以打开用户界面，如下图：

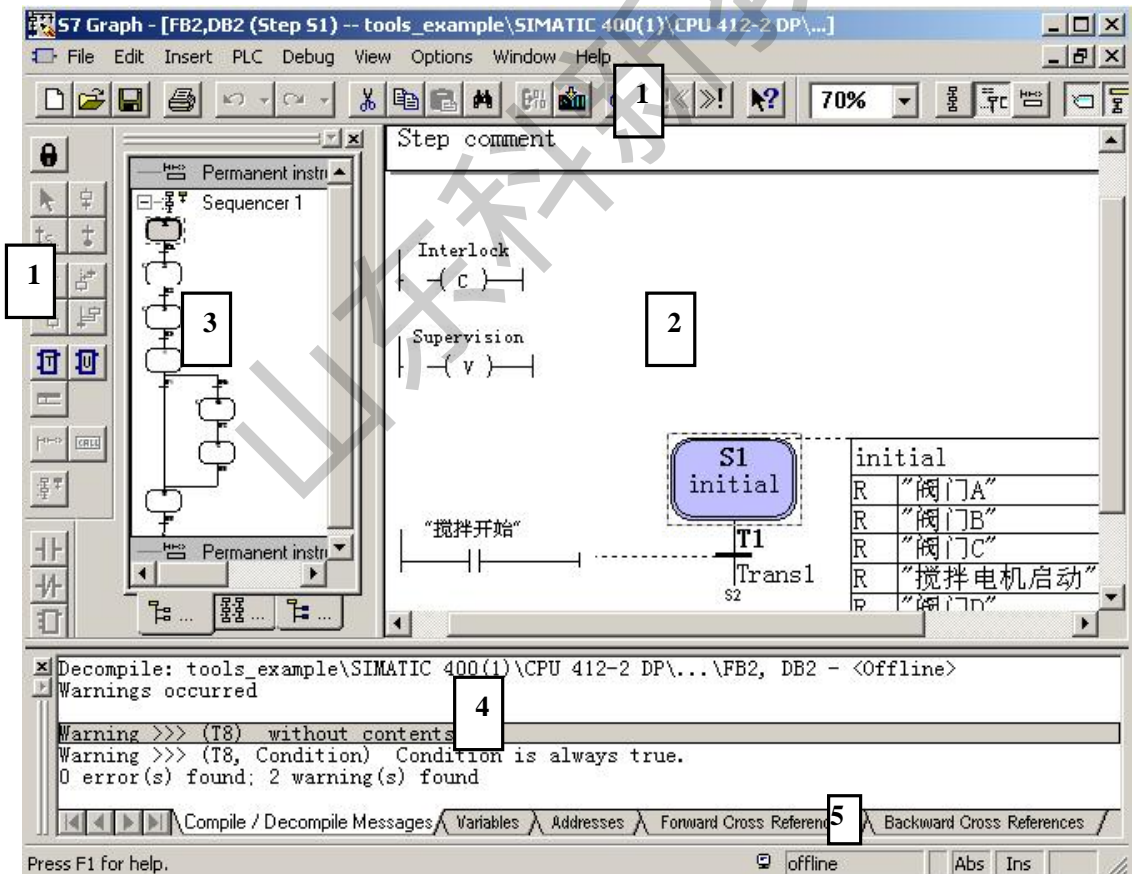


图 3-3: S7-GRAPH 用户界面

1. 工具条
2. 工作区

- 3. 概览窗口
- 4. 详细窗口
- 5. 状态条

3.2.2. 工具条

工具条中可以分为4类功能，

标准功能：主要为文件功能，例如打开，保存，复制，粘贴等功能

视图功能：调整显示作用，如是否显示符号名等

顺控器：包含顺控器元素，如分支，跳转，步等。

LAD/FBD：可以为每步添加LAD/FBD指令。

3.2.3. 工作区

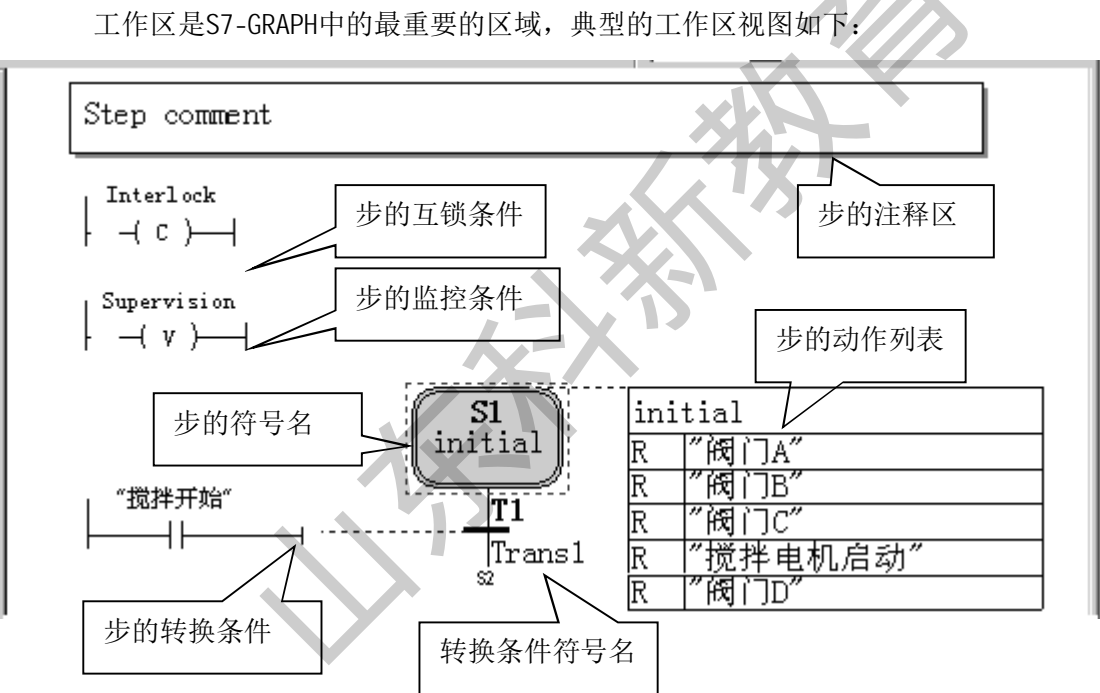


图3-4: S7-GRAPH 工作区视图

互锁条件（Interlock）：

Interlock 是每步的一个可编程条件，它将影响每个单独步的执行情况。

- 如果互锁条件（Interlock）满足，则与互锁条件组合的指令将被执行（在S7-GRAPH中有专门与Interlock 状态相关的指令）
- 如果互锁条件（Interlock）不满足，则：
 - 与互锁条件组合的指令将不被执行
 - 互锁错误信号将为1（事件 event L1）

每个互锁条件（Interlock）最多可以容纳32个LAD/FBD元素，在工作区中用字母“ C” 来表示。如果每个互锁条件（Interlock）为空，即没有编程，系统则认为互锁条件满足。

例子：下图中的监测温度超过了90度的设定值，即互锁条件（Interlock）不满足，则监控时互锁错误信号被触发。

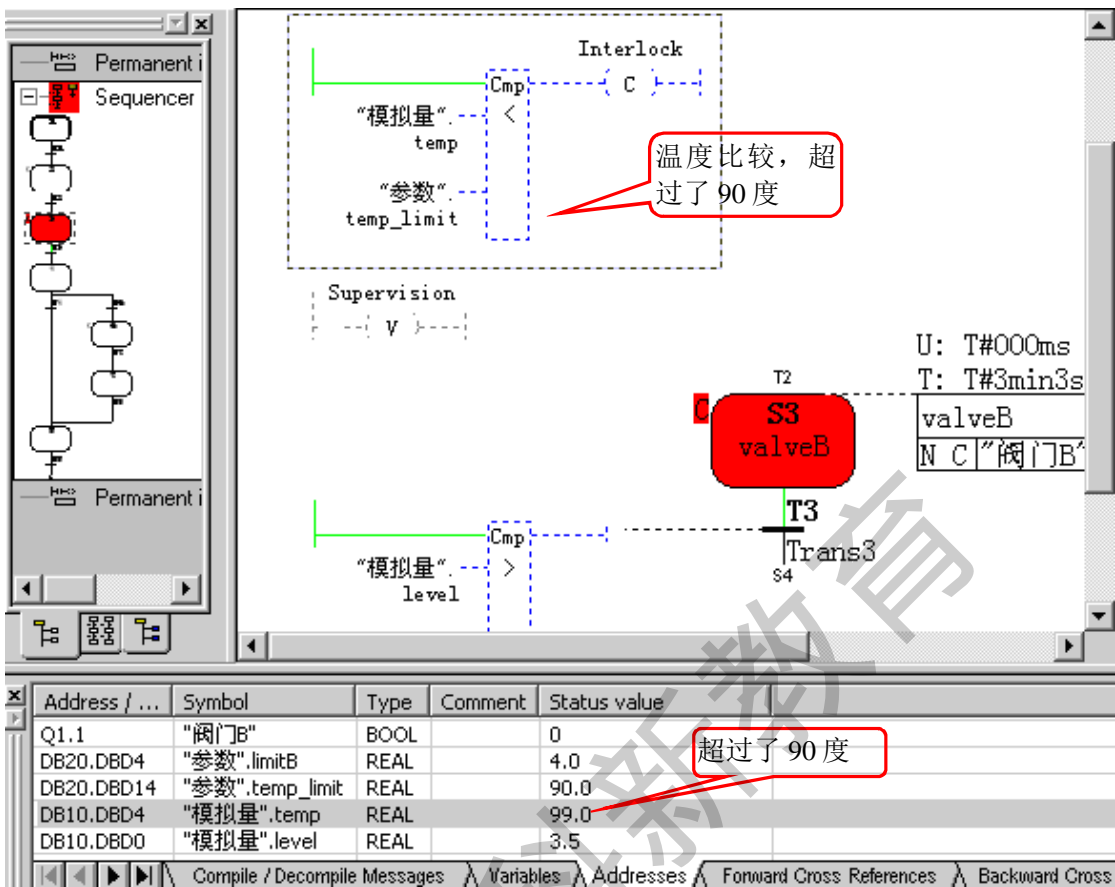


图3-5: 互锁条件使用

监控条件（Supervision）：

Supervision是每步的一个可编程条件，它将影响每个单独步向下一步转换的执行情况。

- 如果监控条件（Supervision）满足，则事件 V1 发生，顺控器不再转换到下一步，当前步保持激活，步的激活时间 Si.U 停止。
- 如果监控条件（Supervision）不满足，并且当前步向下一步的转换条件满足，顺控器将转换到下一步。

每个监控条件（Supervision）最多可以容纳32个LAD/FBD元素，在工作区中用字母“ V” 来表示。如果每监控条件（Supervision）为空，即没有编程，系统则认为监控条件不满足。

3.2.4. 概览窗口

概览窗口的Graphic视图：

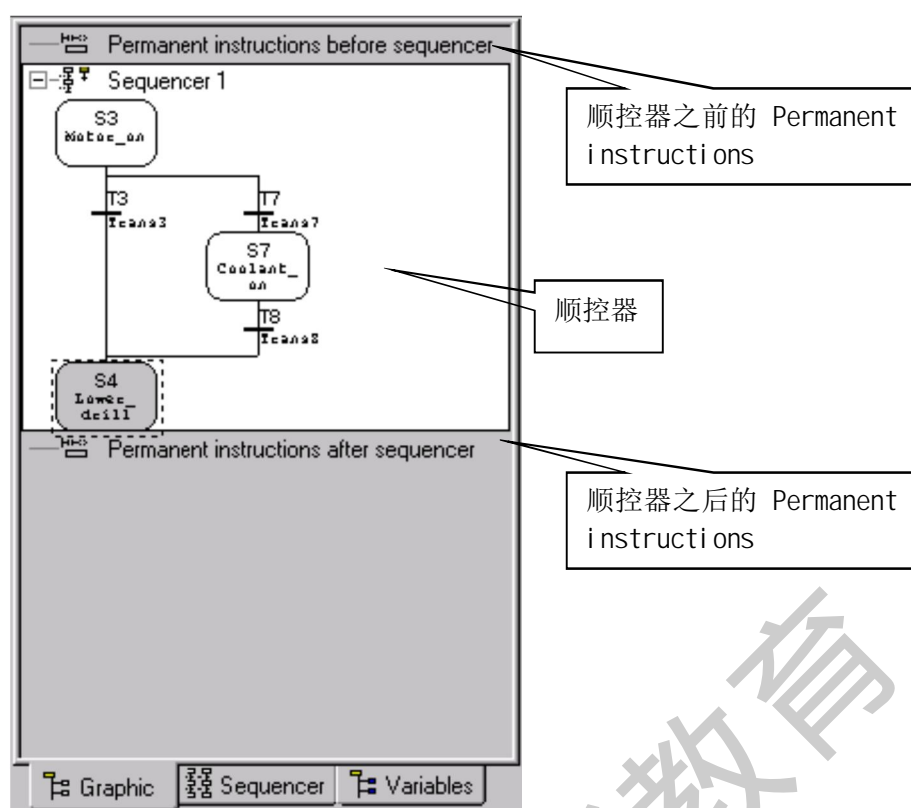


图 3-6: 概览窗口的 Graphic 视图

Permanent instructions: 英文原意为永久的，固定的指令。可以通过右键点击图中的区域，在提示 Insert New Element->Permanent Instruction 下，选择 Condition 或者 Call。不论顺控器的状态如何，这些指令可以在每次扫描时被执行。可以在 S7-GRAPH 的 FB 中编写任意多的指令（资源足够的情况下）。

注意：Permanent instructions 只支持两种格式，即：

- LAD/FBD 格式的 AND, OR, 比较指令（每个 NETWORK 最多可以容纳 32 个）
- CALL 指令

如果希望 Permanent instructions 使用 LAD/FBD 其它格式指令，请将这些指令编写为 FB, FC, 然后用 CALL 格式调用。

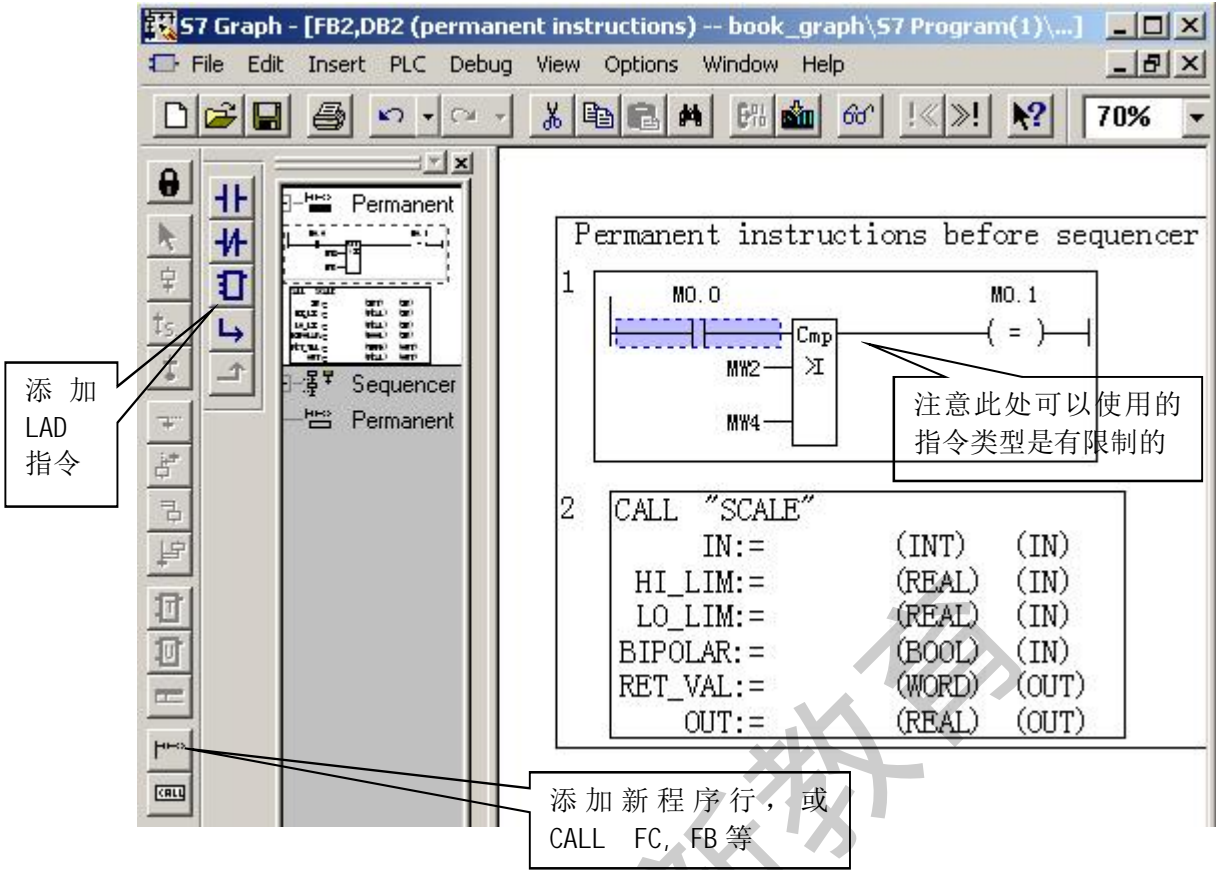


图 3-7: Permanent instructions 使用

概览窗口的Sequencer视图:

对于每个S7-GRAPH的FB程序，可以包括多个Sequencer(顺控器)，如果希望对多个顺控器概览察看，可以选择概览窗口的Sequencer视图

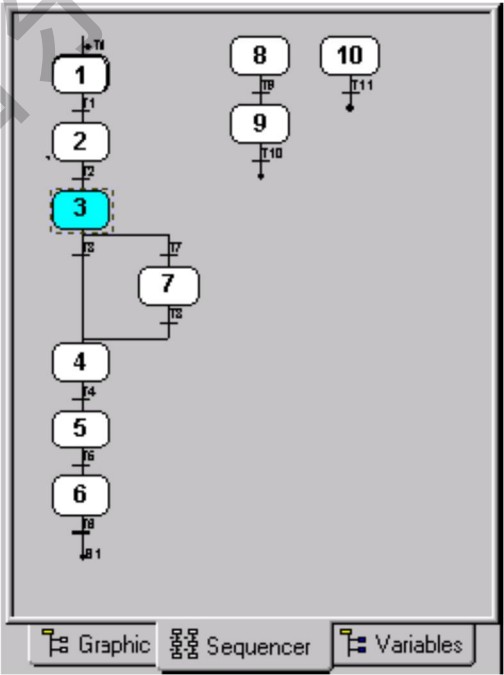


图 3-8: 概览窗口的 Sequencer 视图

概览窗口的Variables视图:

在S7-GRAPH的FB程序中，可以通过概览窗口的Variables视图，察看并编辑如下资源

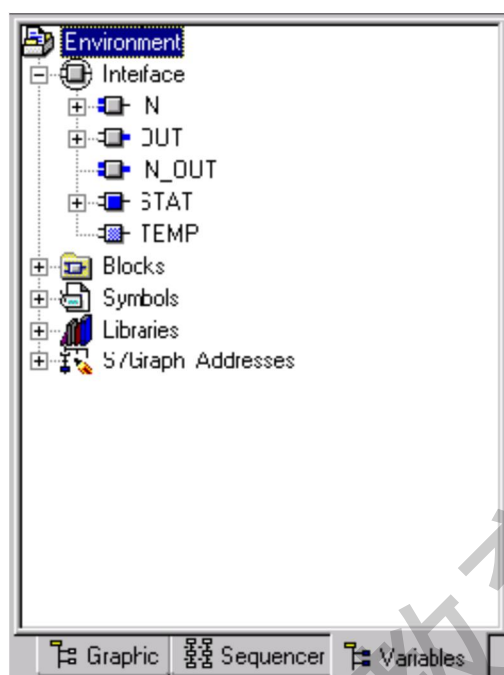


图3-9: 概览窗口的Variables视图

3.2.5. 详细窗口

可以查看编译信息，变量监控，交叉参考等
状态条

显示在线，离线，CPU 状态等。

3.2.6. S7-GRAPH 格式的 FB 与 Source 文件

在生成 STEP7 项目后，可以通过两种方式生成 S7-GRAPH 程序：

- S7-GRAPH 格式的 FB，右键点击项目中的 Block 文件夹，Insert New Object->Function Block->Created in Language 选择 GRAPH
- S7-GRAPH 格式的 Source 文件，右键点击项目中的 Source 文件夹，Insert New Object->选择 GRAPH Source

两者互相转换：S7-GRAPH 格式的 FB 可以通过菜单 File->Generate Source File 生成 S7-GRAPH 格式的 Source 文件；S7-GRAPH 格式的 Source 文件也可以通过菜单 File->Compile 生成 S7-GRAPH 格式的 FB。

两者的区别：S7-GRAPH 格式的 FB 保存时会自动检查语法错误，如果有语法错误，则无法保存；S7-GRAPH 格式的 Source 文件可以随时保存，不检查语法错误。FB 保存后即为可执行代码，可以下载调试；S7-GRAPH 格式的 Source 文件只能用于记录用户程序，无法下载调试。因此，编程者如果不能保证一次编写完成语法正确的程序，建议的使用 Source 文件格式。

4. 顺控器规则

S7-GRAPH 格式的 FB 程序是这样工作的：

- 每个 S7-GRAPH 格式的 FB，都可以作为一个普通 FB 被其它程序调用
- 每个 S7-GRAPH 格式的 FB，都被分配一个背景数据块，此数据块用来存储 FB 参数设置，当前状态等等
- 每个 S7-GRAPH 格式的 FB，都包括三个主要部分：顺控器之前的 Permanent instructions，一个或多个顺控器，顺控器之后的 Permanent instructions

4.1. 顺控器执行规则：

请注意下面描述中的英文关键字，这些关键字对于学习实用 S7-GRAPH 非常重要

1. 每个顺控器都以如下情况开始：

- 一个初始步或者
- 多个位于顺控器任意位置的初始步

只要某个步的某个动作（**action**）被执行，则认为此步被激活（**active**），如果多个步被同时执行，则认为是多个步被激活（**active**）

2. 一个激活的步在如下情况退出：

- 任意激活的干扰（**active disturbances**），例如互锁条件或监控条件的消除或确认
- 并且至后续步的转换条件（**transition**）满足

3. 满足转换条件的后续步被激活

4. 在顺控器的结束位置如果有：

- 一个跳转指令（**jump**），指向本顺控器的任意步，或者 FB 的其它顺控器。此指令可以实现顺控器的循环操作
- 分支停止指令。顺控器的步将停止

步（Step）

在 S7-GRAPH 程序中，控制任务被分为多个独立的步。在这些步中将声明一些动作，这些动作将在某些状态下被控制器执行（例如控制输出，激活或非激活某些步）

激活的步（Active Step）

激活的步是一个当前自身的动作正在被执行的步。

一个步在如下任意情况下，都可被激活：

- 当某步前面的转换条件满足
- 当某步被定义为初始步（**initial step**），并且顺控器被初始化

- 当某步被其它基于事件的动作调用（event-dependent action）

4.2. 顺控器结构

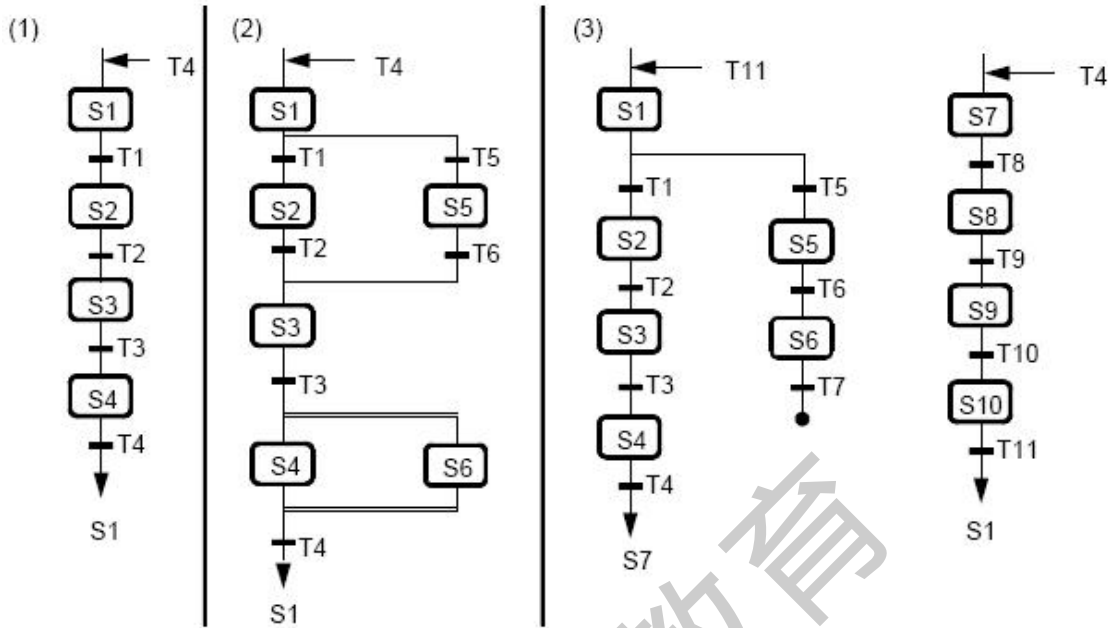


图4-1: 顺控器结构

由上图示例中可以归纳出三点：

1. S7-GRAPH 的 FB 可以是简单的线性结构顺控器
2. S7-GRAPH 的 FB 可以是包括选择结构及并行结构顺控器
3. S7-GRAPH 的 FB 可以包括多个顺控器

4.3. 顺控器元素

在工具栏中有如下顺控器元素，用户应当熟练使用：



Step + Transition（步+转换条件）

在新建的 S7-GRAPH FB 中，默认会有一个步及转换条件，用户可以在此基础上增加新的步及转换条件。步+转换条件的形式总是组合出现的，它们会被系统自动分配一个编号，此编号可以被任意修改。

初始步：当一个 S7-GRAPH FB 被调用时，顺控器中的初始步将被无条件执行，此步不一定是顺控器中编号第一的步。顺控器由 FB 的参数 INIT_SQ=1 被初始化，由初始步开始执行。



Jump（跳转指令）



Open Alternative Branch（打开选择分支）



Close Alternative Branch（关闭选择分支）



Open Simultaneous Branch（打开并行分支）



Close Simultaneous Branch (关闭并行分支)



Branch Stop (分支停止)



Insert Sequencer (插入顺控器)

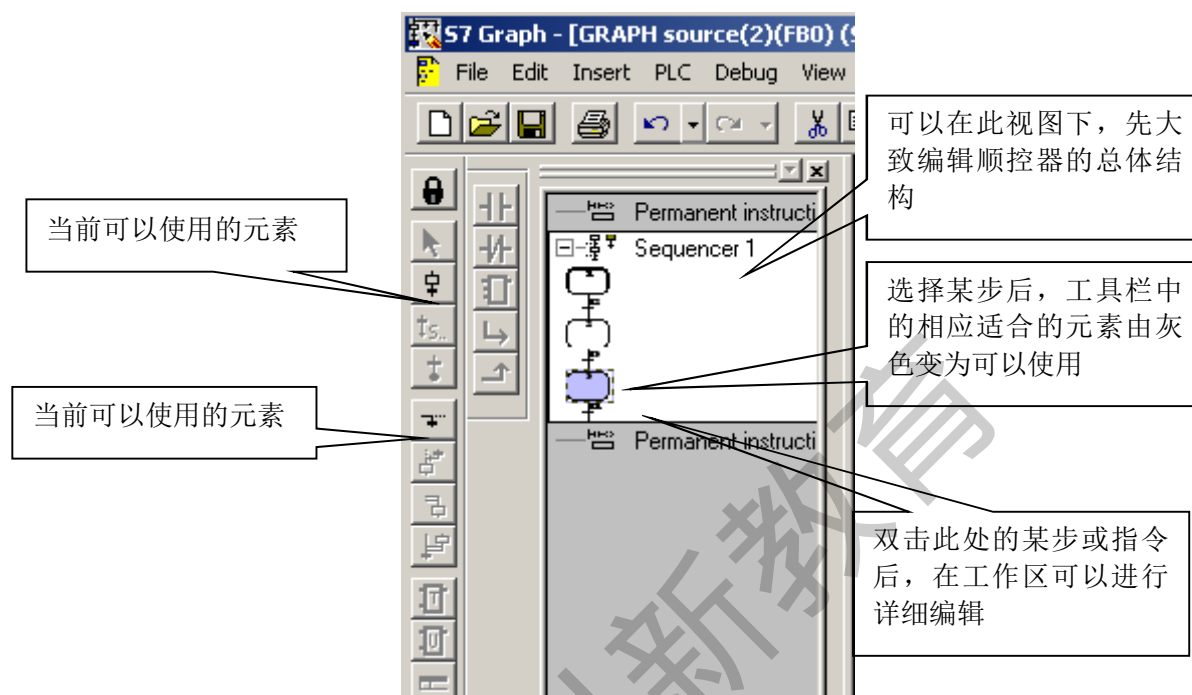


图4-2: 顺控器元素

4.4. 条件与动作的编程

4.4.1. 步的构成及属性

双击顺控器的结构视图的某步后，在工作区可以对每步进行详细编辑，见下图：

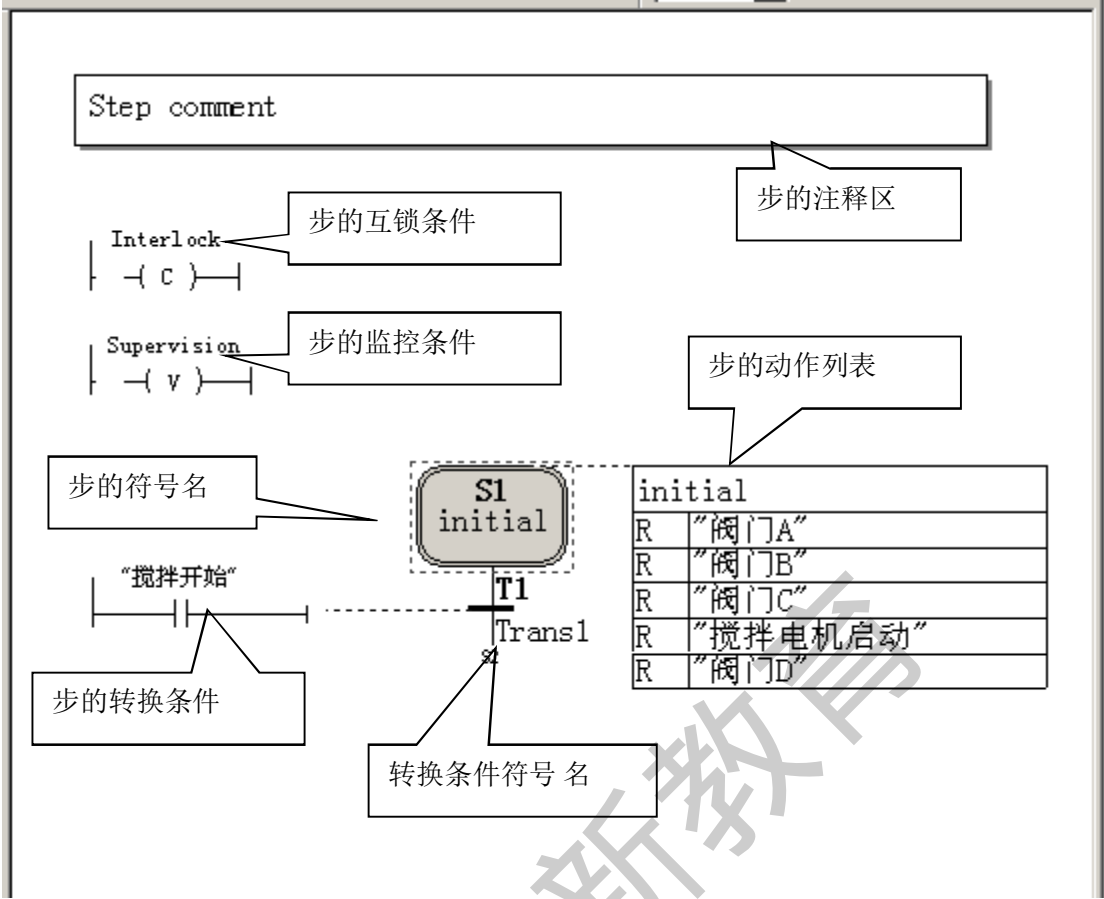


图4-3: 步的构成

步的属性：右键点击步的图标，选择 Object Properties, 注意初始步属性的选择。



图4-4: 步的属性

4.4.2. 步的动作 (action)

添加动作

步的动作（action）在 S7-GRAPH 的 FB 中占有重要位置，用户大部分控制任务要由步的动作来完成，编程者应当熟练掌握所有的动作（action）指令。

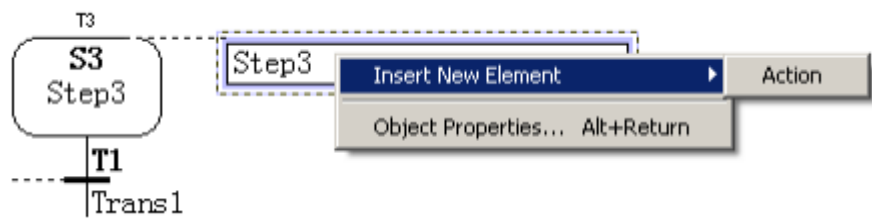


图 4-5: 添加动作

可以通过右键点击步的动作图标，添加新的动作。

动作执行顺序

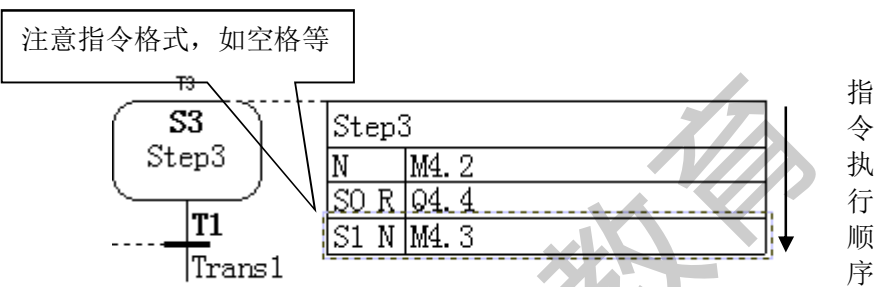


图 4-6: 指令执行顺序

4.4.3. 动作（action）分类:

动作可以分为以下几类:

- 标准动作
 - 与 interlock 有关或无关
- 基于事件的动作
 - 与 interlock 有关或无关
 - 与步的激活或非激活有关
- 计数器，定时器及算术指令的动作

4.4.4. 没有动作的步

顺控器中可以有没有动作的步，顺序执行到这些步后，此步激活，并直接进入后续的转换条件判断部分。

4.4.5. 动作的指令

动作（action）的具体指令描述在 S7-GRAPH 的手册中占有大量篇幅，编程人员应当像学习 LAD 中的基本指令一样，熟练地掌握动作指令。限于篇幅，本文不再讲解。

4.5. S7-GRAPH 特有地址

在 S7-GRAPH FB 中存在一些特有地址，用户可以像使用普通 PLC 地址一样来使用这些地址，具体含义见下表：

地址	含义	使用方式
Si.T	步 i 的当前或上次的激活时间	比较，赋值
Si.U	步 i 的没有干扰的总的激活的时间	比较，赋值
Si.X	显示步 i 是否被激活	常开/常闭触点
Transi.TT	显示转换条件 i 是否满足	常开/常闭触点

表 4-1 S7-GRAPH 特有地址

S7-GRAPH 特有地址的引用格式见下图：

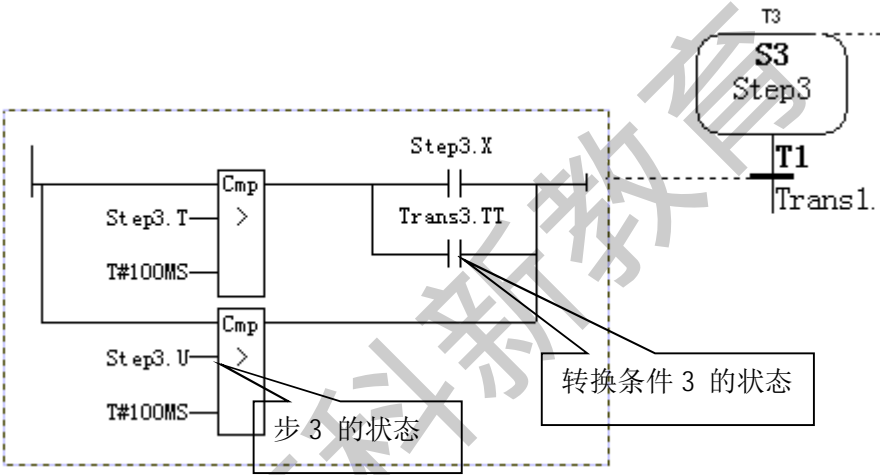


图 4-7: S7-GRAPH 特有地址的引用格式

4.6. S7-GRAPH FB 参数设置

注意：由于 S7-GRAPH FB 是在其它程序中被调用的，所以其接口参数的设置及编译设置对程序的后续调试及执行有着重大的影响，在此加以强调。

在 S7-GRAPH 编辑器菜单中选择 Options->Block Settings... 可以进行 FB 设置，如下图：

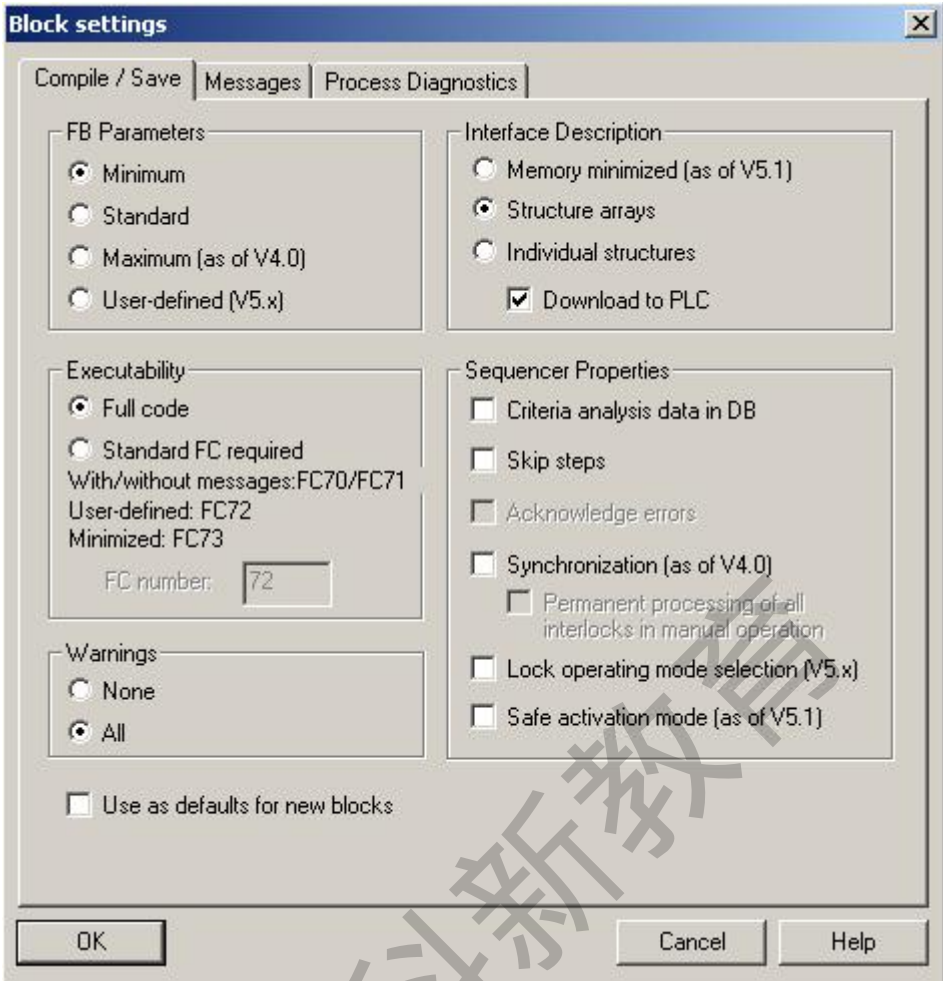


图 4-8: S7-GRAPH FB 参数设置

4.6.1. S7-GRAPH FB 参数设置:

S7-GRAPH FB 参数设置包括 4 种类型:

- 1. Minimum, FB 只包括 SQ_INIT 启动参数, 如果用户的程序仅仅会运行在自动模式, 并且不需要其它的控制及监控功能, 可以选择此模式
- 2. Standard, FB 包括默认参数, 如果用户希望程序运行在各种模式, 并提供反馈及确认消息功能。可以选择此模式。
- 3. Maximum (S7-GRAPH 版本=> V4. x) FB 包括默认参数, 扩展参数, 提供更多的控制, 监控参数
- 4. User-defined (S7-GRAPH 版本=> V5. x) 包括默认参数, 扩展参数, 提供更多的控制, 监控参数

注意: 选择 FB 参数设置后, 可以手动对接口参数进行修改, 例如删除一些不关心的部分接口参数。

下表是对部分参数的解释, 希望有利于用户调试使用。

FB 参数 (上升沿有效)	内部变量 (静态数据区 名称)	顺序控制器 (S7-GRAPH 名称)	含义
ACK_EF	MOP. ACK	"Acknowledge"	故障信息得到确认
INIT_SQ	MOP. INIT	"Initialize"	激活初始步 (顺控器复位)
OFF_SQ	MOP. OFF	"Disable"	停止顺控器, 例如使所有步失效
SW_AUTO	MOP. AUTO	"Automatic (Auto)"	模式选择: 自动模式
SW_MAN	MOP. MAN	"Manual mode (MAN)"	模式选择: 手动模式
SW_TAP	MOP. TAP	"Inching mode (TAP)"	模式选择: 单步调节
SW_TOP	MOP. TOP	"Automatic or switch to next (TOP)"	模式选择: 自动或切换到下一个
S_SEL	-	"Step number"	选择, 激活/去使能在 手动模式 S_ON / S_OFF 在 S_NO 步数
S_ON	-	"Activate"	手动模式: 激活步显示。
S_OFF	-	"Deactivate"	手动模式: 去使能步显示。
T_PUSH	MOP. T_PUSH	"Continue"	单步调节模式: 如果传送条件满足, 上升沿可以触发连续程序的传送。
SQ_FLAGS. ERROR	-	"Error display: Interlock"	错误显示: “互锁”
SQ_FLAGS. FAULT	-	"Error display: Supervision"	错误显示: “监视”
EN_SSKIP	MOP. SSKIP	"Skip steps"	激活步的跳转。
EN_ACKREQ	MOP. ACKREQ	"Acknowledge errors"	使能确认需求
HALT_SQ	MOP. HALT	"Stop sequencer"	停止程序顺序并且重新激活
HALT_TM	MOP. TMS_HALT	"Stop timers"	停止所有步的激活运行时间和块运行和重新激活临界时间
-	MOP. IL_PERM	"Always process interlocks"	“执行互锁”
-	MOP. T_PERM	"Always process transitions"	“执行程序传送”
ZERO_OP	MOP. OPS_ZERO	"Actions active"	复位所有在激活步 N, D, L 操作到 0, 在激活或不激活操作数中不执行 CALL 操作。
EN_IL	MOP. SUP	"Supervision active"	复位/重新使能步互锁
EN_SV	MOP. LOCK	"Interlocks active"	复位/重新使能步监视

表 4-2 S7-GRAPH FB 参数

关于选择 FB 参数的详细解释, 请参考 S7-GRAPH 手册中关于背景数据块结构描述的章节。

4.6.2. 接口描述设置:

在编译选项菜单中，Interface Description 及 Executability 两个选项是密切关联的，当选择占用内存最小（Memory minimized）时，必须选择 FC73。

- 1. Memory minimized 占用内存最小，但诊断及调试的功能受限制。
- 2. Structure arrays 步的描述以数组的方式存储在背景数据块中，不允许符号名方式访问步。
- 3. Individual structures 步的描述以独立的数据结构方式存储在背景数据块中，允许符号名方式访问步及转换条件。这种方式增加了背景数据块的大小。

4.6.3. Executability 选项

- 1. Full code 全代码，所有运行 S7-GRAPH FB 程序需要的代码都将被包括在 FB 自身当中，不需要额外的 FC。
- 2. Standard FC required ($\geq V4.x$) 标准的 FC，如果选择此项，则 S7-GRAPH 使用标准 FC (FC70, FC71, FC72, or FC73) 来支持所有的 S7-GRAPH FBs。填入 FC 的标号即可。

不同 Executability 选项下的区别:

	Full code	Standard FC70	Standard FC71	Standard FC72	Standard FC73
内存要求	取决于顺控器的大小	约 8200 字节	约 7750 字节	约 10700 字节	约 8150 字节
任务限制	无	在 S7-GRAPH V5.1 及更高版本中可以使用算术运算，定时器，计数器功能	在 S7-GRAPH V5.1 及更高版本中可以使用算术运算，定时器，计数器功能，无用户自定义参数	无	无诊断消息，无步激活时间 (.U)，无“跳跃步功能”选项，无法跳跃至用户指定的步及转换条件

表 4-3 S7-GRAPH 程序占用内存情况

4.6.4. 顺控器属性设置

- 1. Criteria analysis data in DB 当使用 SIMATIC 标准应用程序 ProTool /ProAgent 时，此选项用来将分析数据存储在背景数据块中。
- 2. Skip steps 激活步跳转功能（有利于调试）

3. Acknowledge errors 如果在顺控器执行时，发生了干扰（disturbance），此时必须在输入参数"ACK_EF"上确认，顺控器才会继续执行。此选项在把 FB 参数设置为 Minimum 时不可选。
4. Synchronization (as of V4.0) 同步（4.0 以上版本有效）允许将程序与过程同步
5. Permanent processing of all interlocks in manual operation 在手动模式下，如果选择此项，无论实际情况如何，每步的 interlocks 都将满足，并在监控的状态下显示出来。
6. Lock operating mode selection (V5.x) 锁定操作模式。如果选择此参数，则当程序运行时，无法通过编程设备改变顺控器运行模式（如手动/自动模式切换）
7. Safe activation mode (as of V5.1) 安全激活模式。此选项影响步的激活状态，当某步被激活时，系统同时将所有其它不能被激活的步去除激活。例如：在分支选择程序中的某步被激活时，另一个分支的所有步都将被去激活。

4.7. S7-GRAPH FB 程序的背景数据块

S7-GRAPH FB 程序的背景数据块保存着顺控器执行的所有信息，因此，S7-GRAPH FB 程序的背景数据块在调试及运行中有着重要的作用。编程人员要务必注意以下几点：

- 当修改顺控器程序后，下载时务必下载背景数据块
- PLC 断电后，背景数据块将保持，建议用户在供电恢复后，执行顺控器初始化操作
- 在其它用户程序中，不要随意改写 S7-GRAPH FB 程序背景数据块内容，否则将引起顺控器执行混乱，并对系统安全性构成威胁
- 在其它用户程序中，改写 S7-GRAPH FB 程序背景数据块内容，可以改变顺控器状态，编程者务必慎用。

注意：在下载 S7-GRAPH FB 程序时，建议选中图中所有项。



图 4-9: S7-GRAPH FB 下载参数设置

4.8. 背景数据块进阶使用

进阶使用背景数据块信息的意义：

在用户在其它程序中调用 S7-GRAPH FB 程序时，S7-GRAPH FB 提供给用户的接口资源是有限的，用户如果希望使用 S7-GRAPH 的更多、更高级的功能（例如当前步的状态，转换条件信息等），就需要对 FB 的背景数据块的结构有所了解。这样就可以达到灵活使用 S7-GRAPH 程序，S7-GRAPH 程序与其它程序无缝衔接的目的。

前面的讲解已经说明了关于 FB 的设置及分配问题，下面重点强调一下 FB 的背景数据块的结构，见下表：

区域	名称	长度
FB 参数	S7-GRAPH FB 输入/输出参数	依赖于参数设置 2 字节: minimum 10 字节: standard/maximum 不确定: user-define
保留工作区	G7T_0	16 字节
转换条件	转换条件名称 (例如 Trans1, Trans2)	转换条件个数*16 字节
保留工作区	G7S_0	32 字节
步	步的名称 (例如 Step1, Step2)	步个数*32 字节
顺控器状态	-	-
内部工作区	-	-

表 4-4 S7-GRAPH FB 背景数据块结构

对于 FB 的背景数据块中关于转换条件的结构, 请参考下表:

组件	说明	数据类型	内部读	内部写	外部读	外部写
TV	转换条件有效	BOOL	yes	no	yes	no
TT	转换条件满足	BOOL	yes	no	yes	no
TS	转换切换	BOOL	yes	no	yes	no
CF_IV	CRIT_FLT 条目不允许	BOOL	yes	no	yes	no
TNO	显示用户定义的转换条件数量	INT	no	no	yes	no
CRIT	当前处理周期中, 转换条件中 LAD/FBD 元素 (最多 32 个) 的状态	DWORD	yes	no	yes	no
CRIT_OLD	前一个处理周期中, 转换条件中 LAD/FBD 元素 (最多 32 个) 的状态	DWORD	yes	no	yes	no
CRIT_FLT	如果错误发生, 将复制 CRIT 状态	DWORD	yes	no	yes	no

表 4-5 S7-GRAPH FB 背景数据块中转换条件的结构

对于 FB 的背景数据块中关于步条件的结构, 请参考下表:

组件	说明	数据类型	内部读	内部写	外部读	外部写
S1	步激活进入事件	BOOL	yes	no	yes	no
L1	interlock 离开事件	BOOL	yes	no	yes	no
V1	Supervision 进入事件	BOOL	yes	no	yes	no
R1	保留	BOOL	no	no	no	no

A1	错误被确认	BOOL	yes	no	yes	no
S0	步激活离开事件	BOOL	yes	no	yes	no
L0	Interlock 进入事件	BOOL	yes	no	yes	no
V0	Supervision 离开事件	BOOL	yes	no	yes	no
X	步激活状态	BOOL	yes	no	yes	no
LA	Interlock 不满足	BOOL	yes	no	yes	no
VA	Supervision 满足	BOOL	yes	no	yes	no
RA	保留	BOOL	no	no	no	no
AA	保留	BOOL	no	no	no	no
SS	系统内部变量	BOOL	no	no	no	no
LS	interlock 状态	BOOL	yes	no	yes	no
VS	supervision 状态	BOOL	yes	no	yes	no
SNO	用户步的编号	INT	no	no	yes	no
T	步激活的所有时间	TIME	yes	no	yes	no
U	没有干扰情况下步激活的时间	TIME	yes	no	yes	no
CRIT_LOC	当前处理周期中，Interlock 中 LAD/FBD 元素（最多 32 个）的状态	DWORD	yes	no	yes	no
CRIT_LOC_ERR	当 interlock 离开时，将复制 CRIT_LOC 状态	DWORD	yes	no	yes	no
CRIT_SUP	当前处理周期中，Supervision 中 LAD/FBD 元素（最多 32 个）的状态	DWORD	yes	no	yes	no
SM	系统内部变量	BOOL	no	no	no	no
LP	系统内部变量	BOOL	no	no	no	no
LN	系统内部变量	BOOL	no	no	no	no
VP	系统内部变量	BOOL	no	no	no	no
VN	系统内部变量	BOOL	no	no	no	no
H_IL_ERR	系统内部变量	BYTE	no	no	no	no
H_SV_FLT	系统内部变量	BYTE	no	no	no	no
RESERVED	保留	DWORD	no	no	no	no

表 4-6 S7-GRAPH FB 背景数据块中步的结构

表格解释：

内部读：在 S7-GRAPH 程序内部，对此变量的读操作

内部写：在 S7-GRAPH 程序内部，对此变量的写操作

外部读：在 S7-GRAPH 程序外部，其它程序中对此变量的读操作

外部写：在 S7-GRAPH 程序外部，其它程序中对此变量的写操作

步激活进入事件：当步刚刚变为激活状态时（进入状态），此事件被认为是步激活进入事件，例如：对于 Step1.S1，此变量在 Step1 进入激活状态时，接通一个扫描周期。

步激活离开事件：当步刚刚变为非激活状态时（离开状态），此事件被认为是步激活离开事件，例如：对于 Step1.S0，此变量在 Step1 进入非激活状态时，接通一个扫描周期。

5. 程序调试

把程序下载后，可以进行调试工作。在菜单 Debug-Monitor 中，可以监视顺控程序的执行状态，在菜单 Debug-Control Sequencer 中，可以控制顺序执行的手动/自动模式，当前激活的步骤。

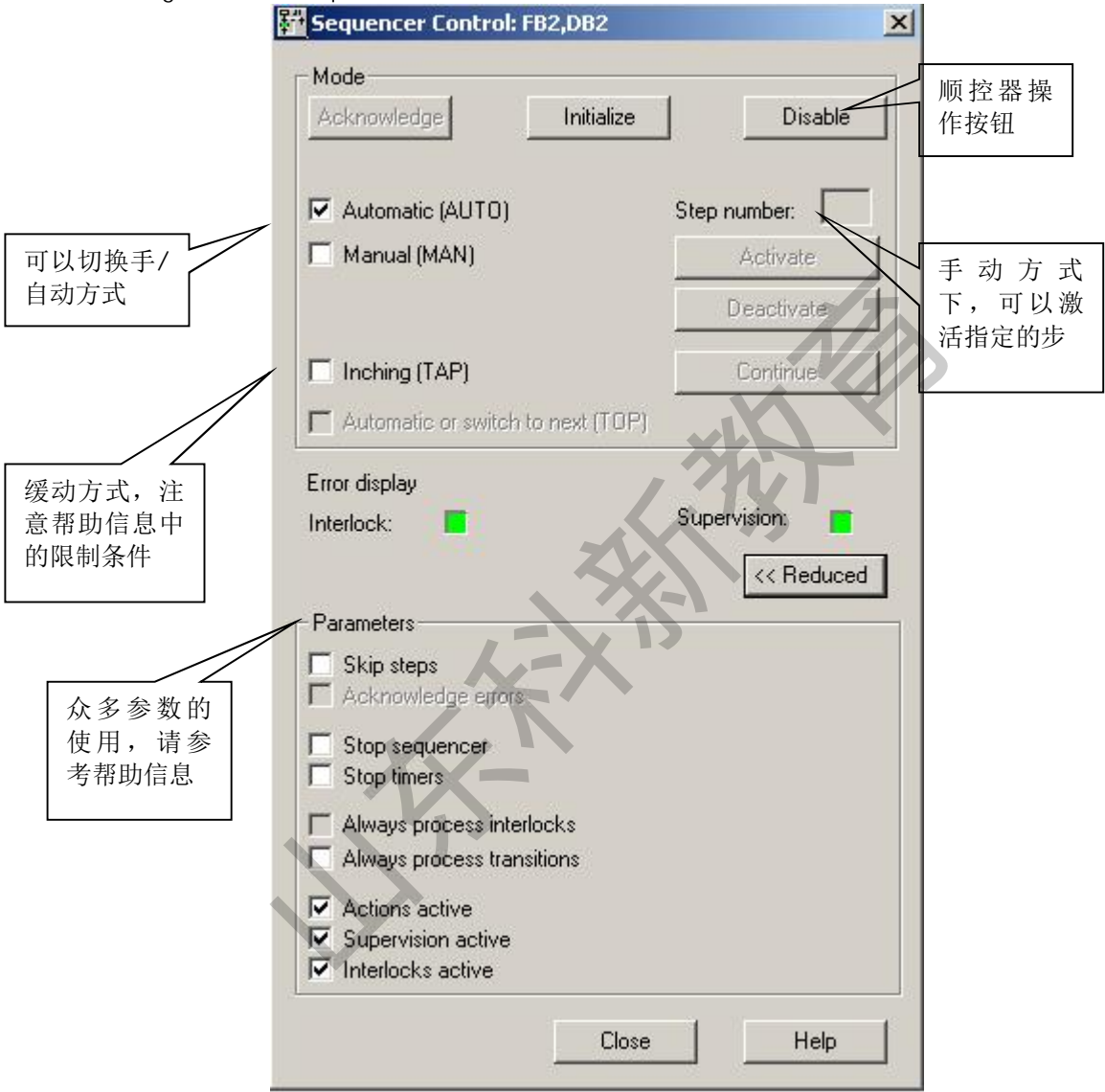


图 5-1: 顺序控制功能

菜单 Debug-Synchronization 可以同步顺控器，模式为手动模式，可以选择当前激活的步，如下图：

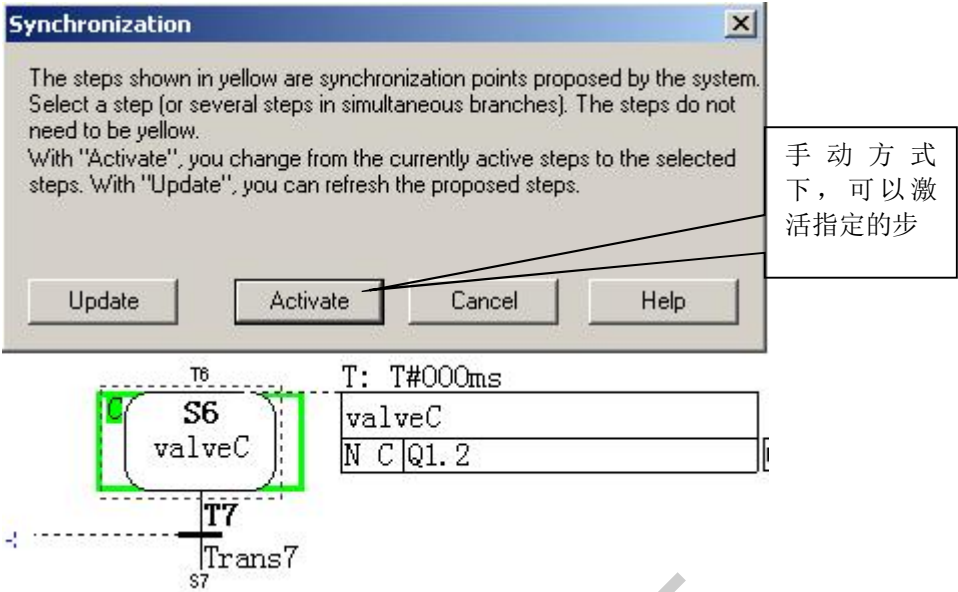


图 5-2: 手动调试模式

重要提示：调试功能可以更改当前程序的手动/自动模式，当前激活的步骤（手动模式下程序限定条件无效，并可以直接选择当前激活的步骤，例如直接从第一步跳转至最后一步）。这样的操作有可能跳过系统原有的保护，可能对人身或生产造成伤害及影响，所以在对工艺及 S7-GRAPH 不熟悉的情况下，请勿使用此功能。

至此，本章节用尽量少的篇幅强调了 S7-GRAPH 编程语言中需要注意的知识点，在后续的章节再列举出一些用户在实际使用当中经常会遇到的问题，希望能够对用户有所帮助。

6. S7-GRAPH 应用于虚拟工程

6.1. 虚拟工程工艺要求

下面将以的虚拟工程中的原料配比环节来举例说明 S7-GRAPH 的使用。

原料配比示意图：

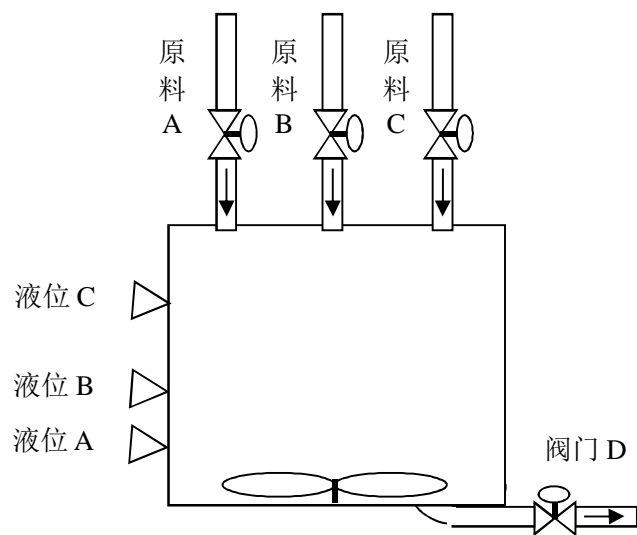


图 6-1: 原料配比示意图

原料配比环节需要将三种原料 A,B,C 按照一定的工艺要求进行混合，并且搅拌均匀后由阀门 D 送至下一工艺流程。并且在箱体流入液体过程中的任意时刻，如果温度超出了温度 X，则关闭入口阀门。输入输出符号定义：

- 阀门A Q1.0
- 阀门B Q1.1
- 阀门C Q1.2
- 搅拌电机 Q1.3
- 阀门D Q1.4
- 搅拌复位 I1.0
- 搅拌开始 I1.1

简单工艺描述：

初始化

打开阀门 A，当液体 A 到达限位 A 时，关闭阀门 A，
打开阀门 B，当液体 B 到达限位 B 时，关闭阀门 B，
启动搅拌电机，5 分钟后关闭搅拌电机，

如果附加工艺选择为“ 0”：

- 则打开阀门 D，流程结束。

如果附加工艺选择为“ 1”：

- 则打开阀门 C，当液体 C 到达限位 C 时，关闭阀门 C，
- 启动搅拌电机，10 分钟后关闭搅拌电机，，
- 打开阀门 D，流程结束。

下图为原料配比工艺流程图：

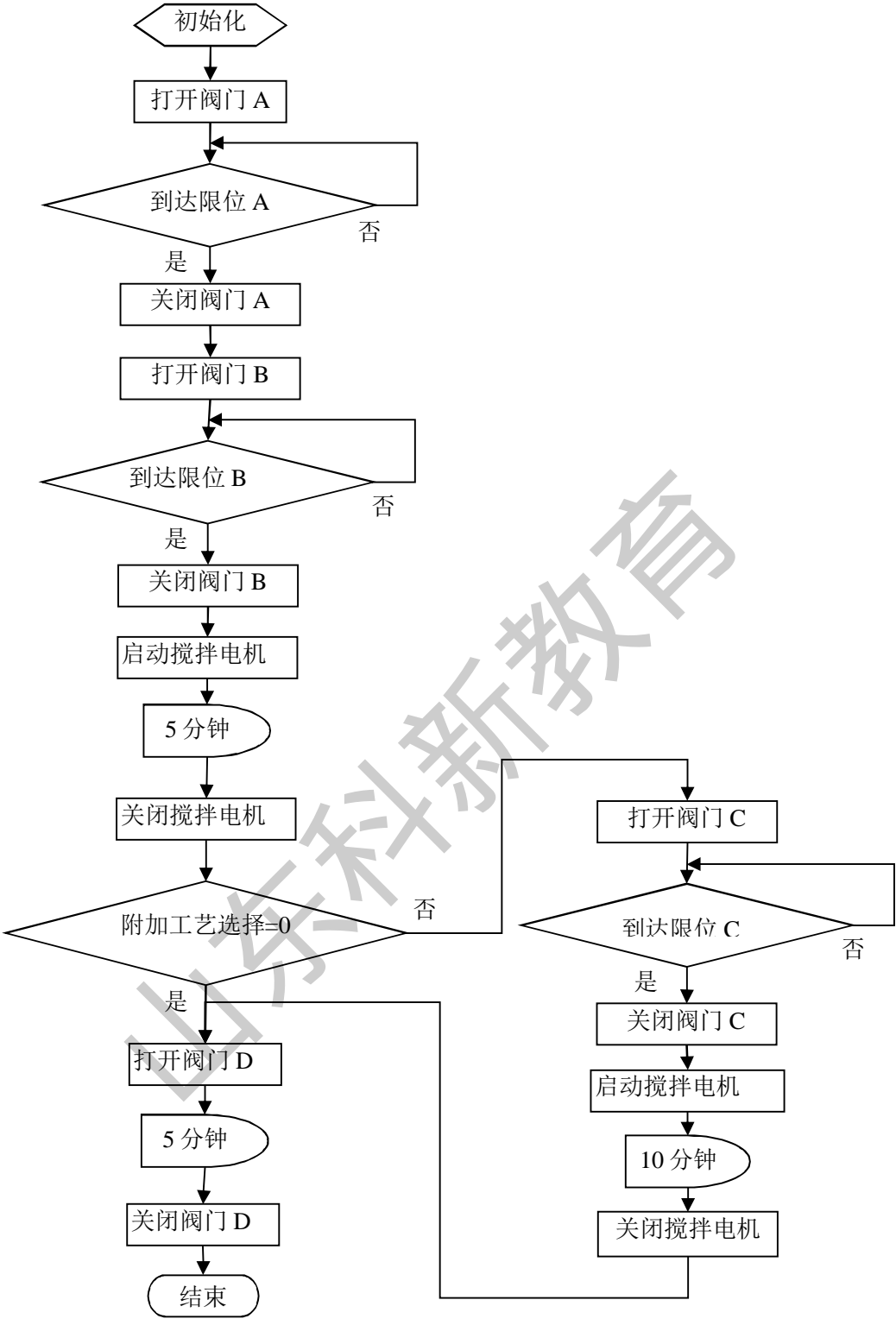


图 6-2: 原料配比工艺流程图

6.2. S7-GRAPH 简单示例

原料配比环节的控制任务属于非常典型的顺序控制流程，在下面的例子中，将使用 FB2 编写原料配比程序。

1) 添加功能块

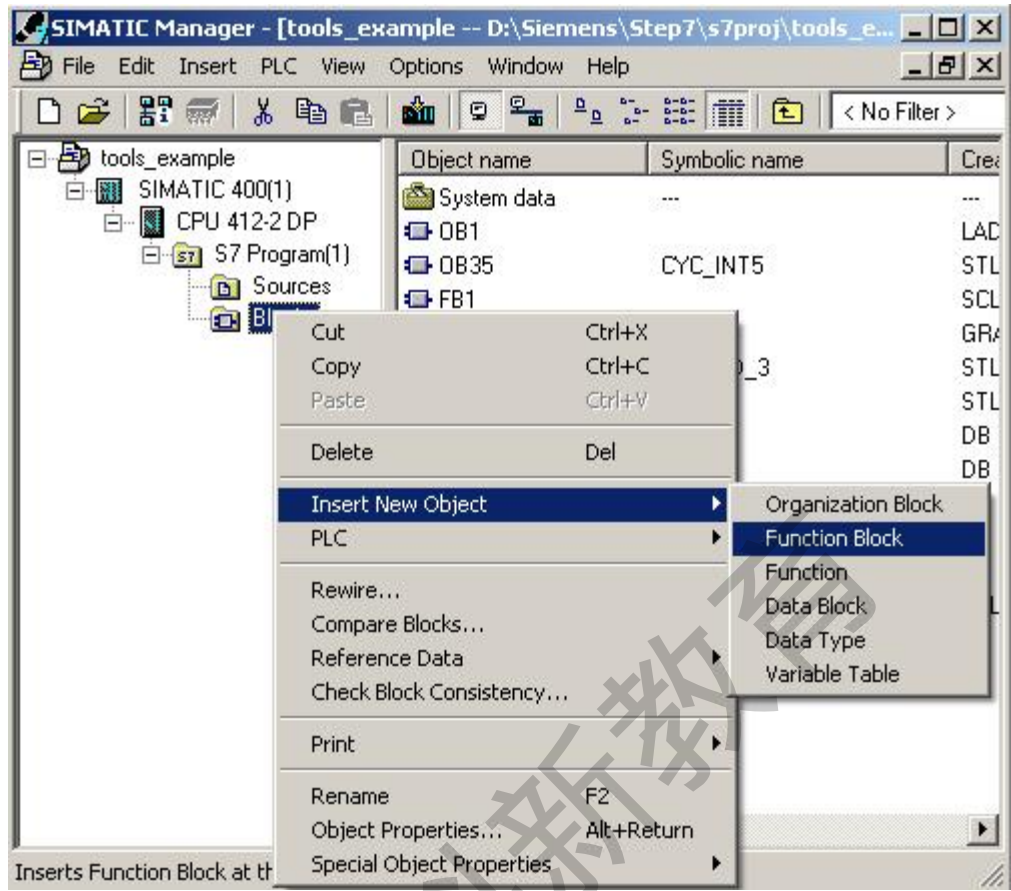


图 6-3: 添加功能块

2) 选择功能块类型

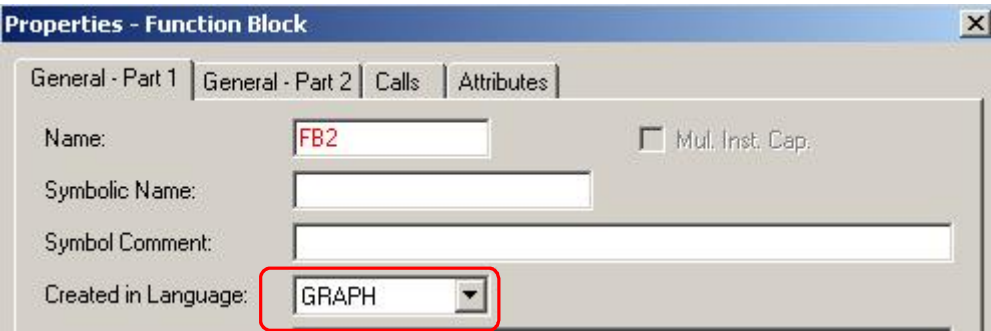


图 6-4：选择功能块类型

3) 编程元素

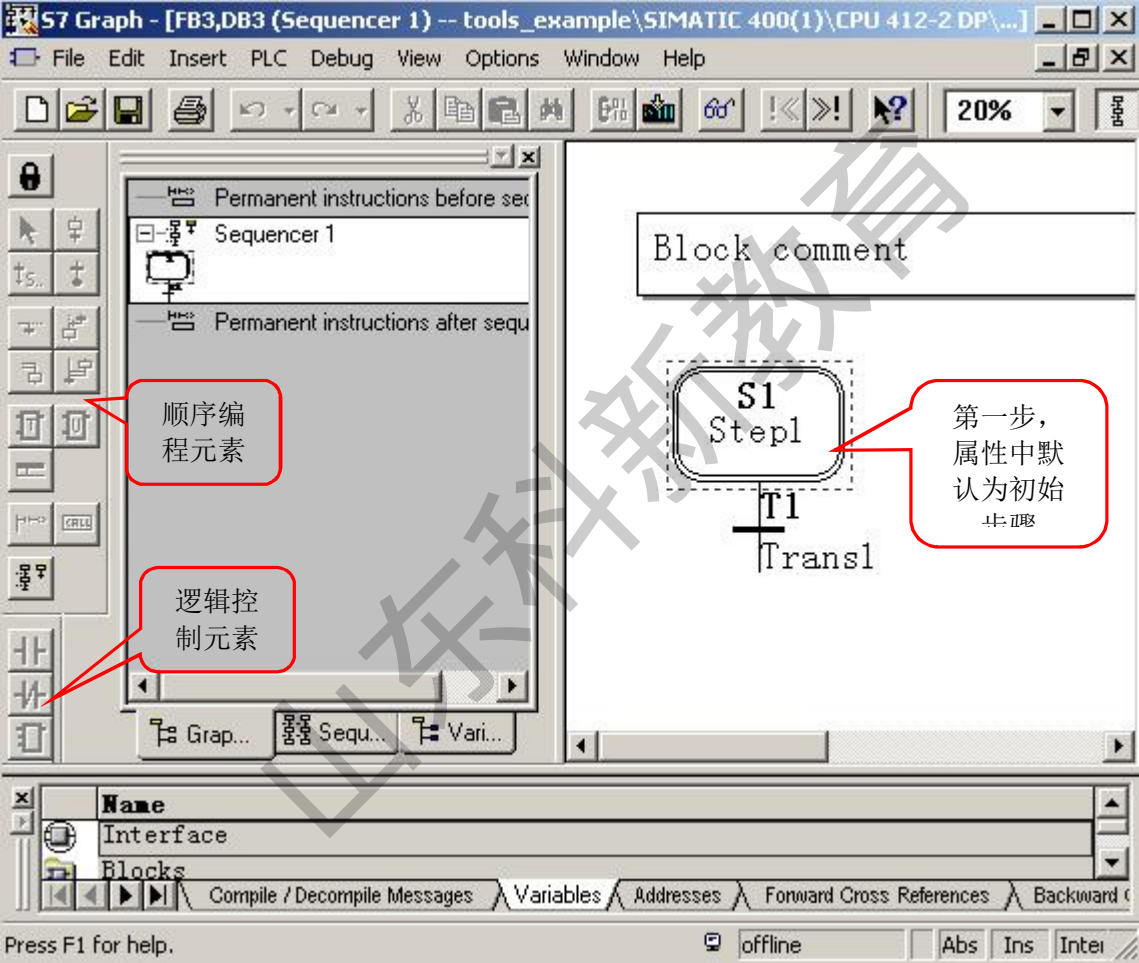


图 6-5：编程元素

4) 编程分支及流程

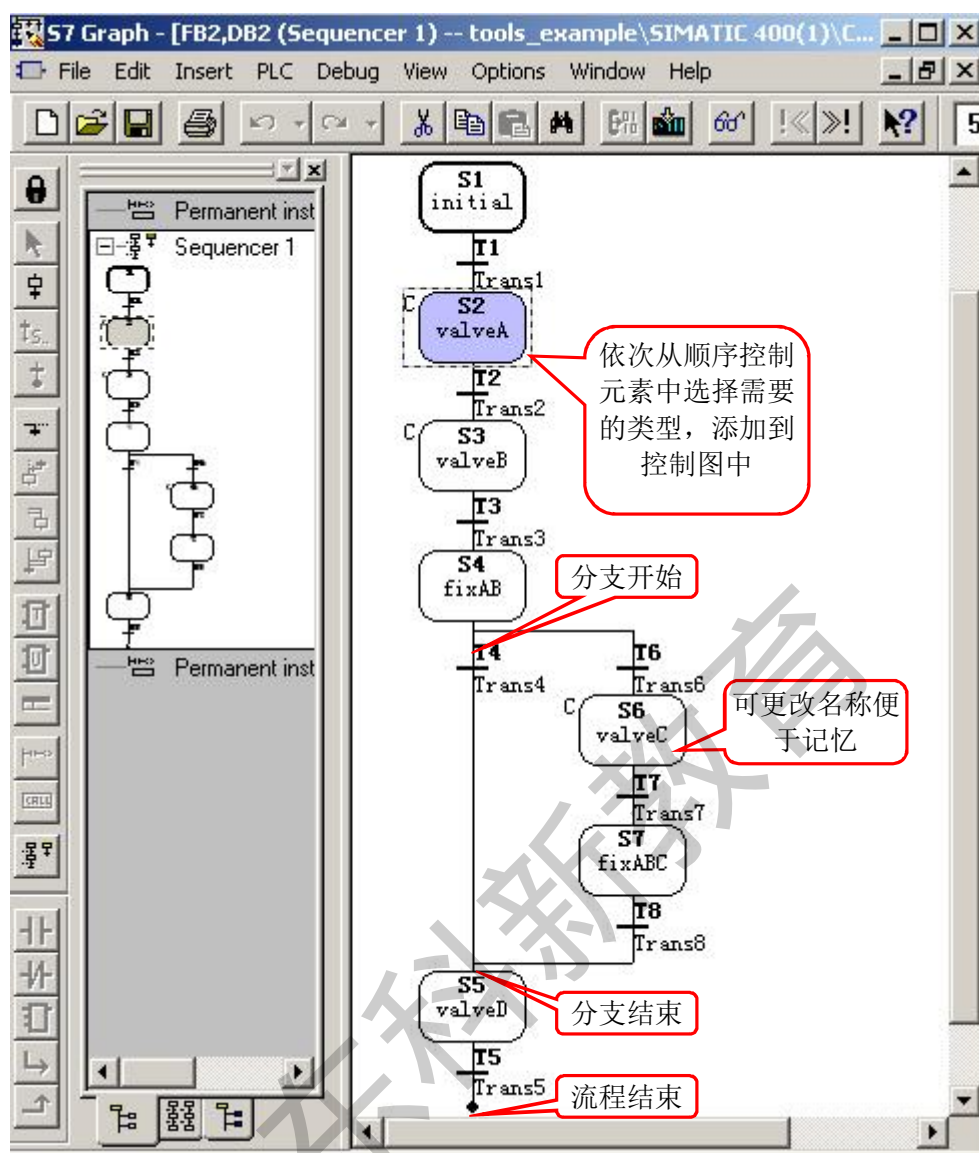


图 6-6: 编程分支及流程

5) 初始化处理

右键->Insert New Element->Action
关闭所有阀门

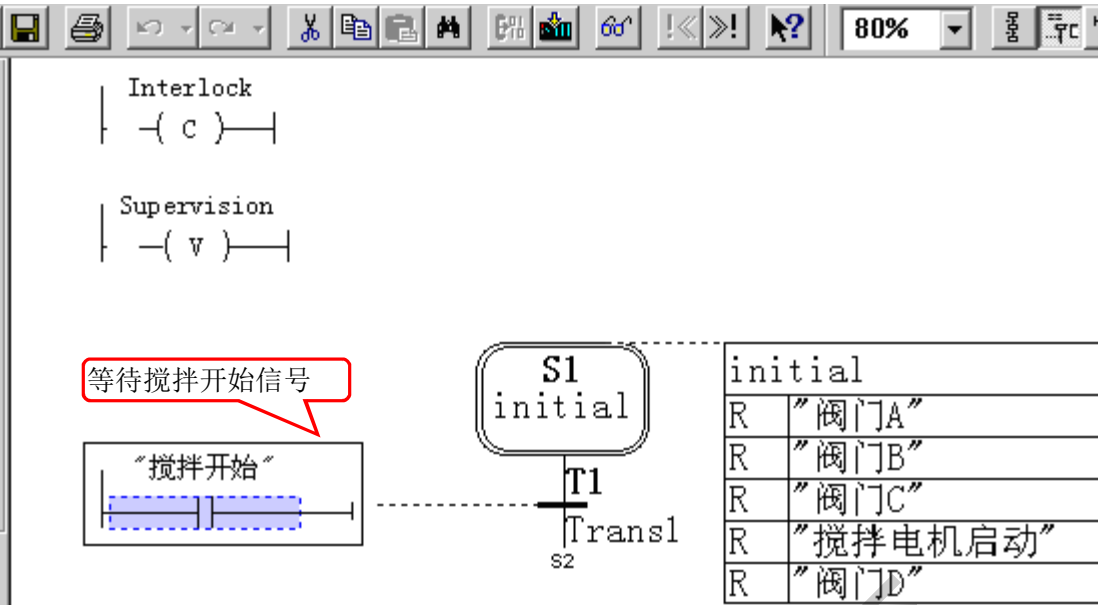


图 6-7：初始化

6) 互锁条件及指令

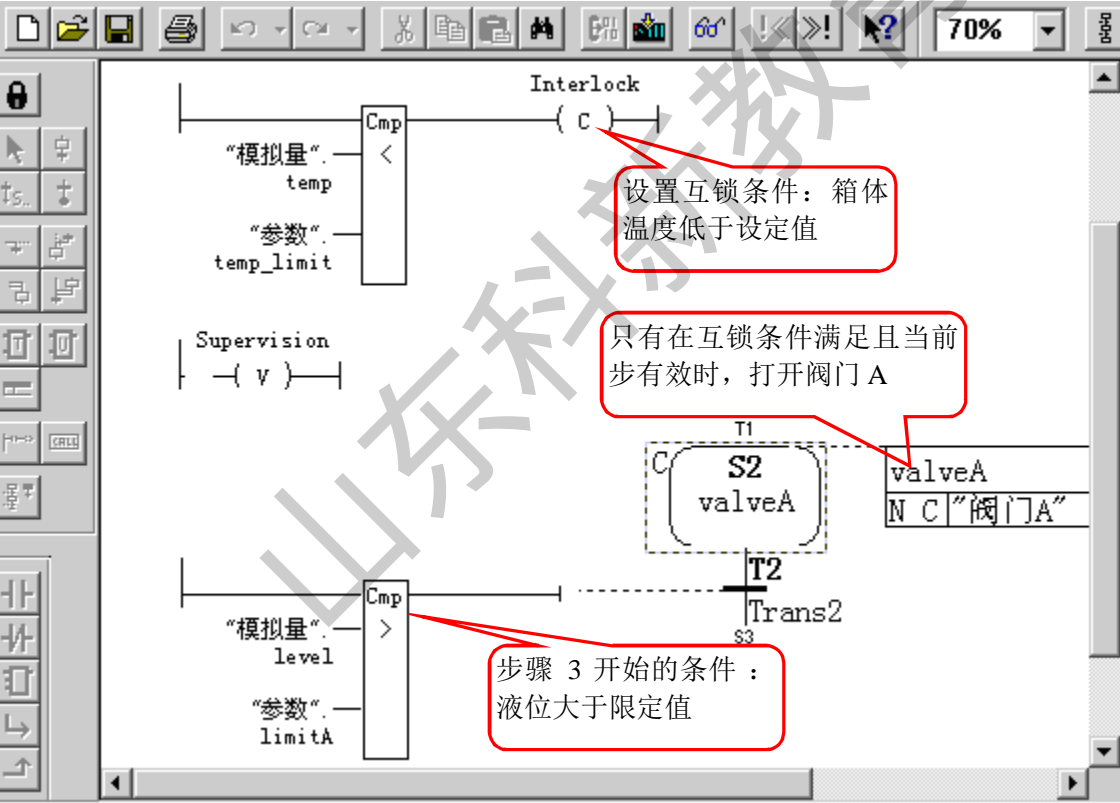


图 6-8：编程环境视图

7) 流程分支

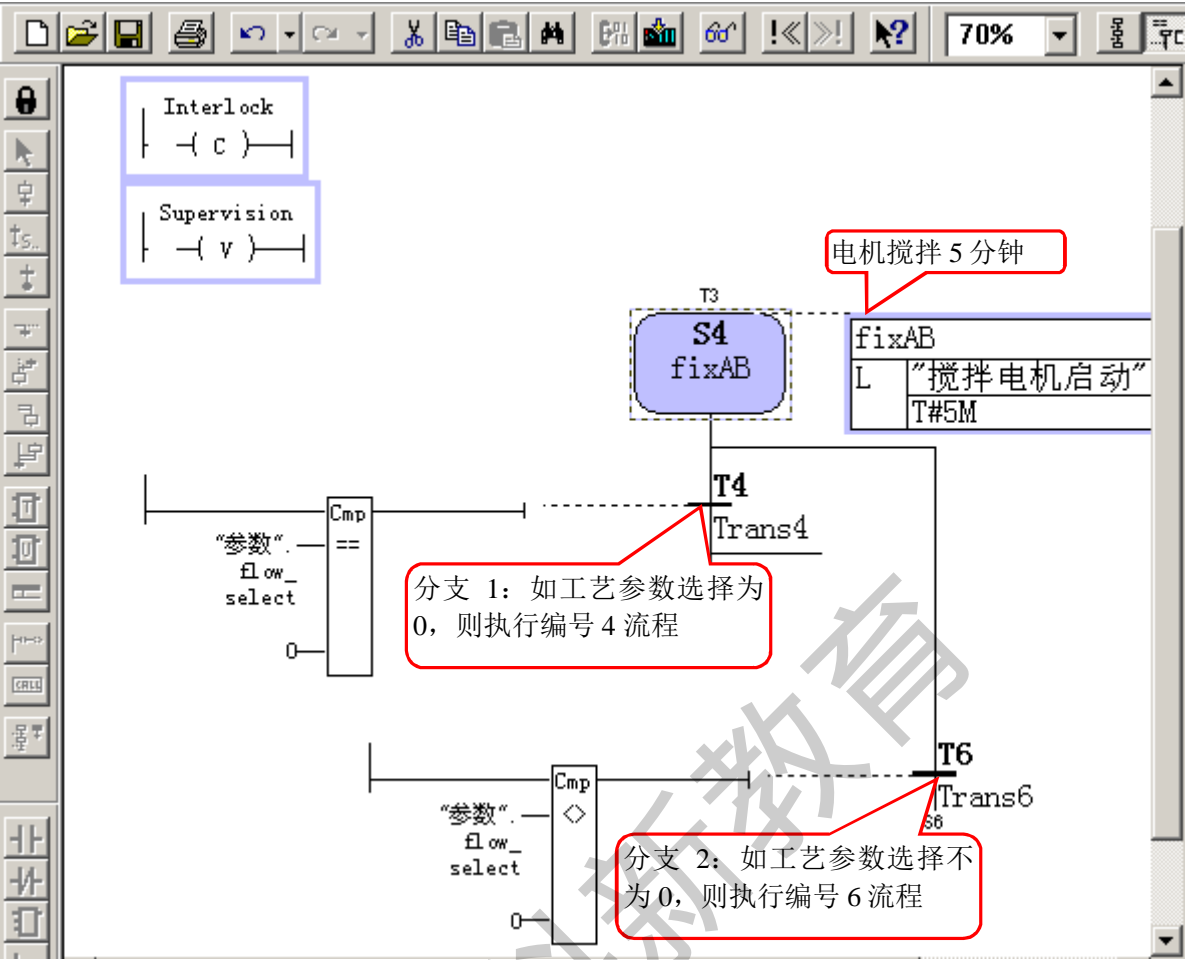


图 6-9: 流程分支

- 8) 省略步骤 6-7 具体程序
- 9) 步骤 5 结束

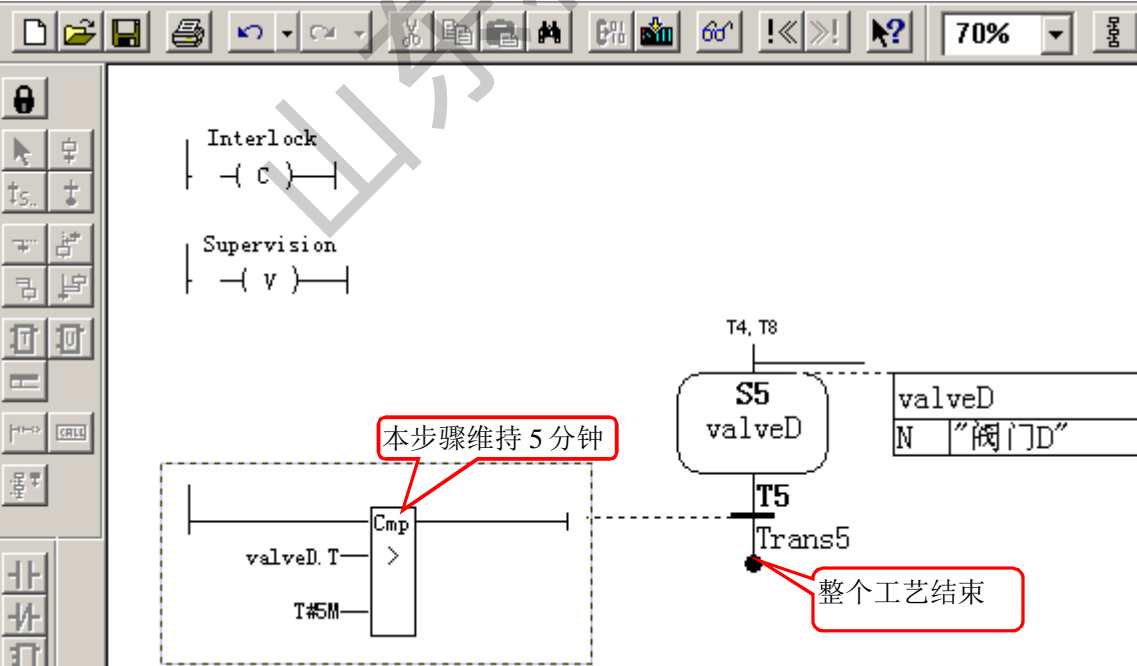


图 6-10: 流程结束

10) 菜单 Options-Application Setting 进行参数设置，本例选择了 FB 最小化参数及全代码，具体含义请参考菜单中的帮助说明

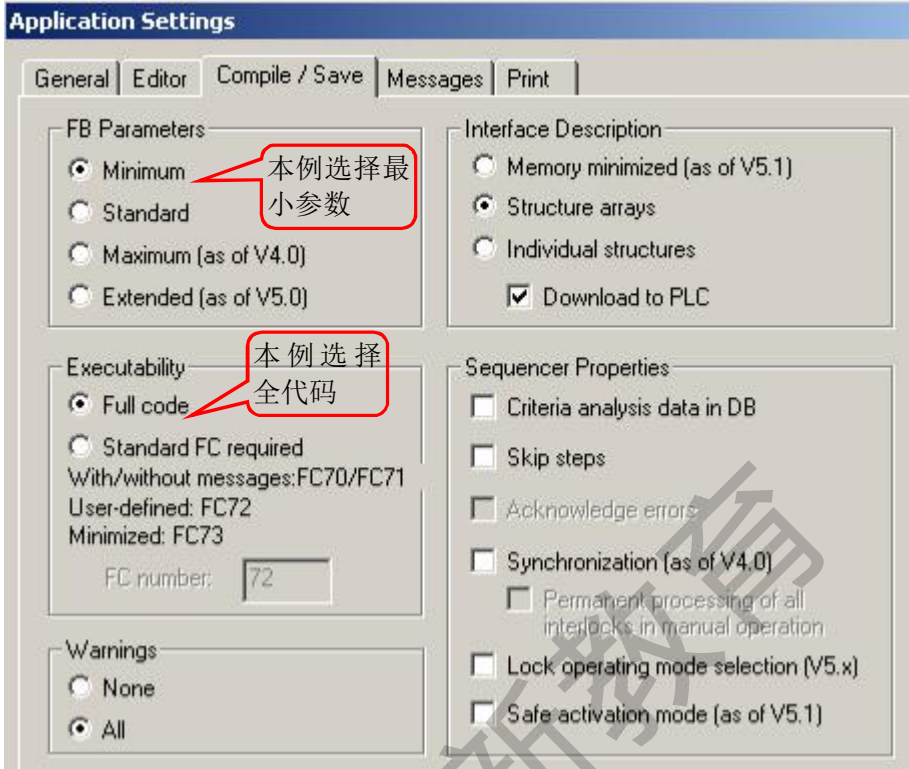


图 6-11: 参数设置

11) 存盘时程序会自动编译，在 OB1 中调用 FB2

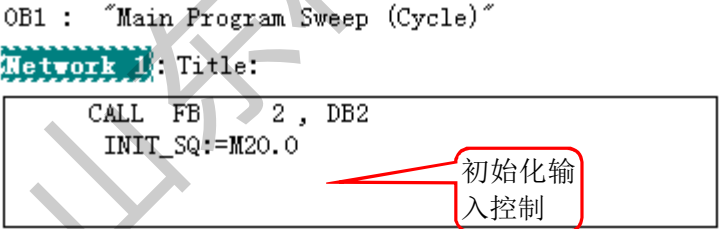


图 6-12: 调用 FB2

12) 下载后，监控 FB2

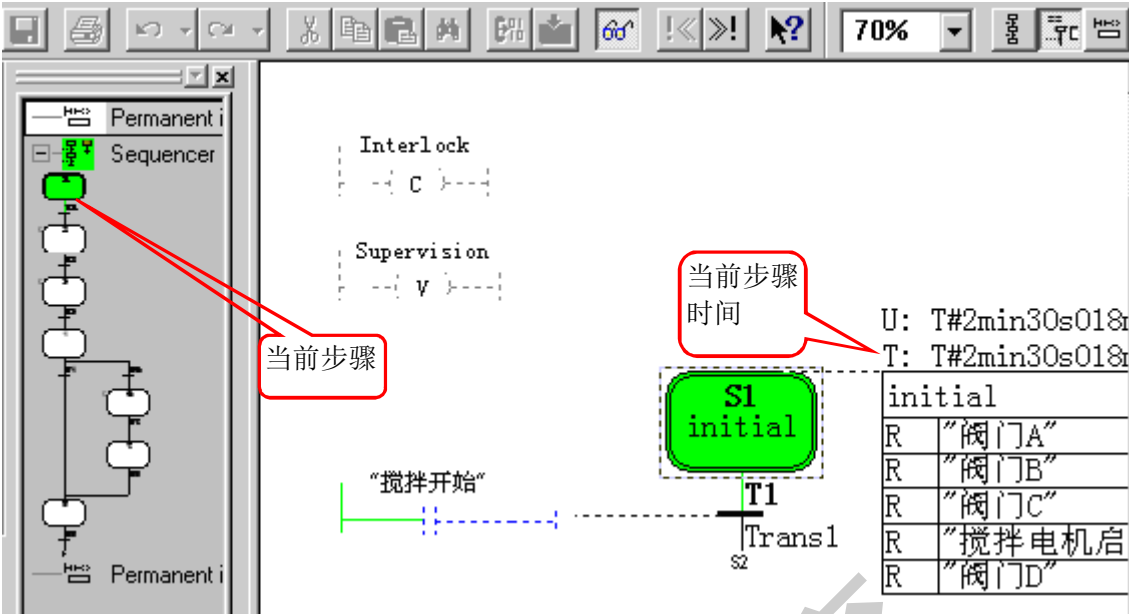


图 6-13: 监控 FB2

13) 输入“搅拌开始”为“1”后，阀门A打开，当液位高于设定值时，进入下一步

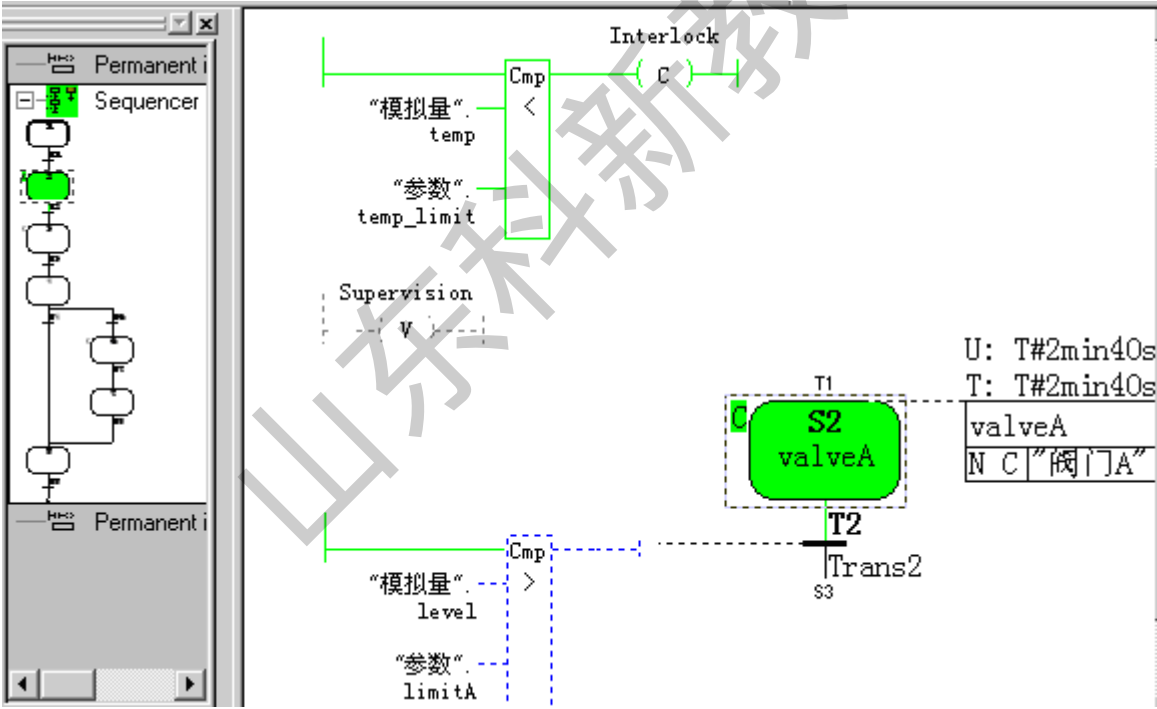


图 6-14: 测试 FB2

14) 温度超高时的监控画面，在 Addresses 中可以监控到程序中使用的变量值

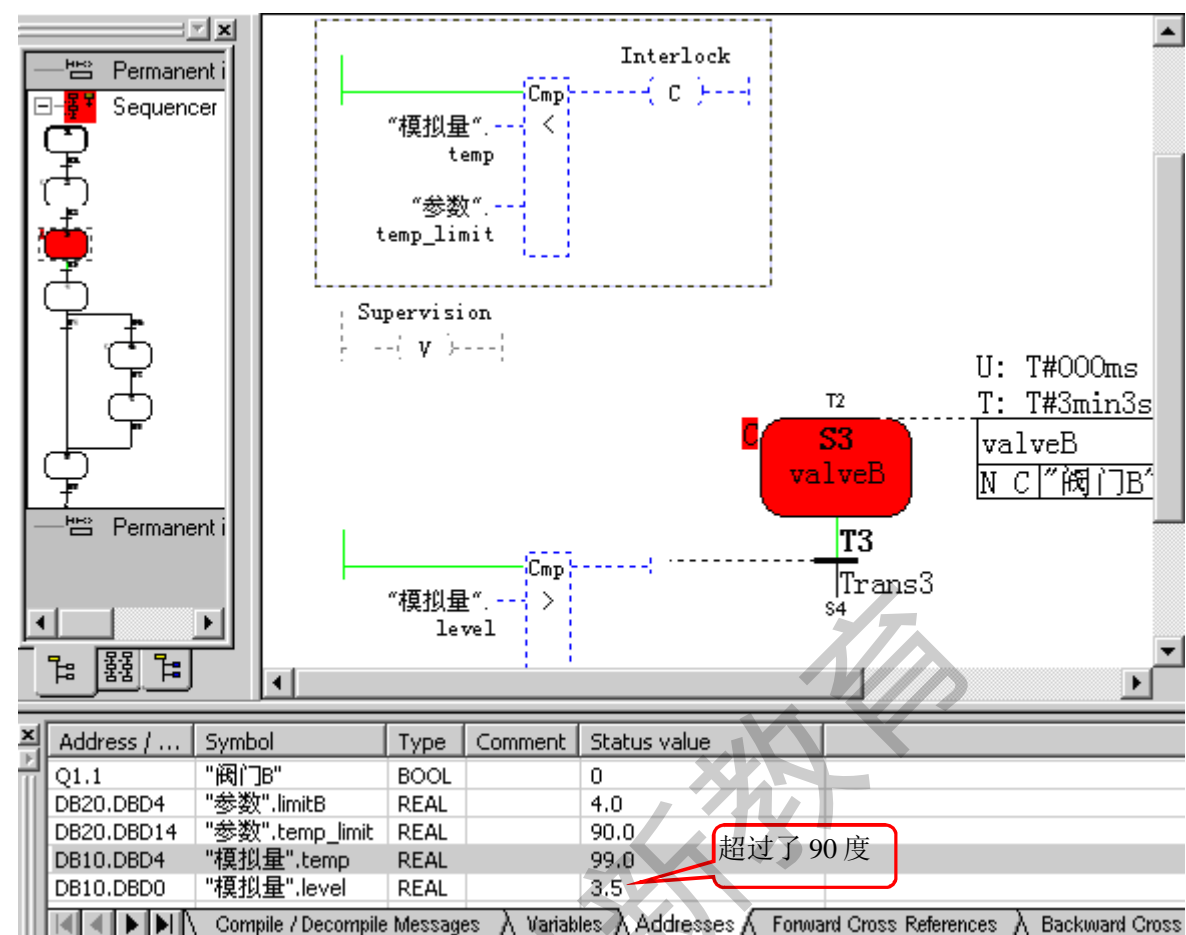


图 6-15: 温度超高互锁

15) 菜单 Debug-Control Sequencer 可以控制顺序执行的手动/自动模式，当前激活的步骤。

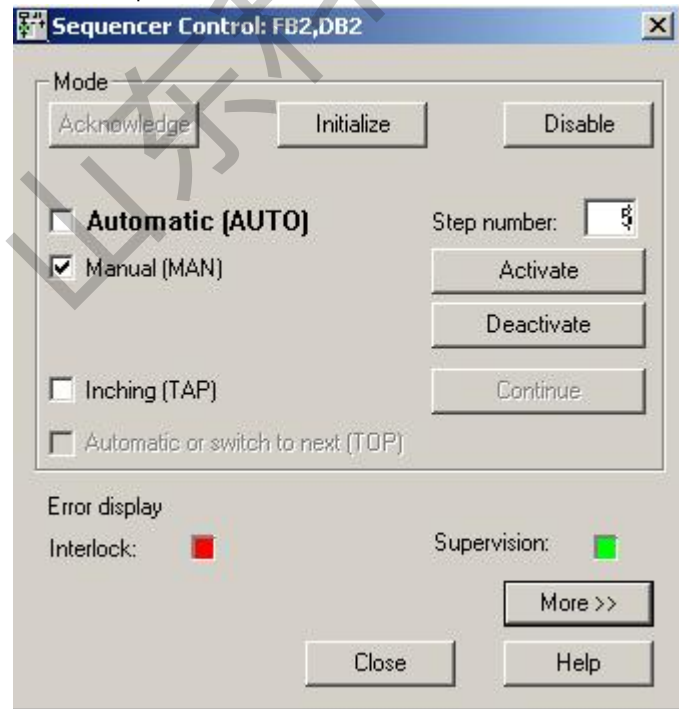


图 6-16: 顺序控制功能

重要提示：由于此功能可以更改当前程序的手动/自动模式，当前激活的步骤（手动模式下程序限定条件无效，并可以直接选择当前激活的步骤，例如直接从第一步跳转至最后一步）。这

样的操作有可能跳过系统原有的保护，可能对人身或生产造成伤害及影响，所以在对工艺及 S7-GRAPH 不熟悉的情况下，请勿使用此功能。

至此，一个简单的 S7-GRAPH 程序示例就结束了，本文中仅是对其非常简单的作了介绍。任何编程语言都有其复杂性，并非一朝一夕就可掌握，关于 S7-GRAPH 的具体使用，请按照本文提供的地址连接下载 S7-GRAPH 手册。

重要提示：

- ✧ 本文的虚拟工程与真实工程实例有重大差别，示例中并未遵循规范的工程设计流程进行编程，请读者切勿将其与工程实例相混淆。
- ✧ 由于此例子是免费的，任何用户可以免费复制或传播此程序例子。程序的作者对此程序不承担任何功能性或兼容性的责任，使用者风险自负。
- ✧ 西门子不提供此程序例子的错误更改或者热线支持。

7. S7-GRAPH 常见问题

7.1. 与监控调试相关问题

7.1.1. 问题：监控不完整

问题：为什么当顺控器超过 100 步时，监控程序时所有或部分步的状态和转换无法显示？

解答：这是由于 S7 CPU 系统资源有限，不能满足状态功能造成的。对于 S7-300 CPU (CPU318 除外)，当顺控步数超过 100 时，将会发生上述问题。而 CPU318 和 S7-400 模板相应的上限值将近 200 步。

解决方法：

- 在 S7-GRAPH 编辑器中，在 "Optionss > Application Settings... > 页签： General" 路径下，可修改 "maximum number of status jobs used" 选项中的监视任务数。此任务数增大，则监视程序将占用更多 CPU 的系统资源。
- 使用功能更强大的 CPU (如 CPU318 或 CPU400)。
- 减少步数和转换器数。

7.1.2. 问题：顺控器不切换

问题：为什么在自动模式下，即使转换条件已满足时顺控器也没有切换？

解答：如果 S7-GRAPH 程序的背景数据块没有被及时更新，可能会导致自动模式下的顺控器不继续

切换，比较常见的情况如下：

- 转换条件满足，但顺控器停在 Step 1 中
- 转换条件的状态不显示

解决方法：在 SIMATIC 管理器和 CPU 中删除背景数据块。然后在 S7-GRAPH 编辑器中再次生产背景 DB 并将其载入 CPU。

特别强调：在 S7-GRAPH 菜单中 "Optionss > Application Settings... > General > On Saving" 的选项中，应当选择 Include instance DB，这样当用户更改 FB 结构并存盘时，FB 的背景数据块也被更新。建议选择此项，利于保持项目数据一致性，很多调试时出现的问题都与背景数据块更新有关，此处不再依次列举。

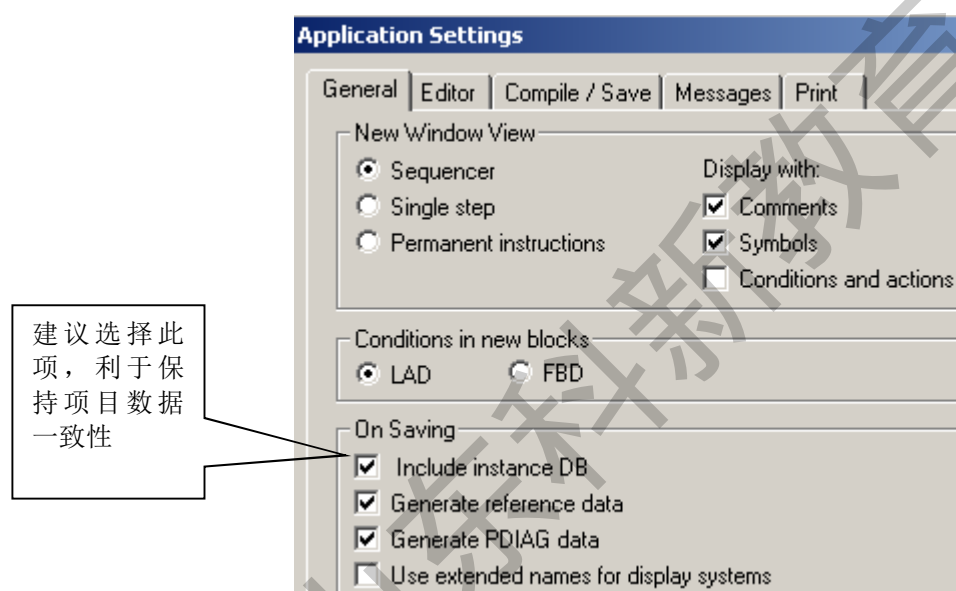


图 7-1: 数据块的刷新选项

7.1.3. 问题：无法监视顺控器

问题：上载 S7-GRAPH FB 之后，为何不能监视它。

解答：请检查“ Interface Descriptio” 选项，查看是否选择了“ Download to PLC”。如果没有，在菜单“ Options > Block Settings > Compile/Save” 下设置“ Download to PLC” 选项。然后重新保存块。 设置该选项后，下载 S7-GRAPH FB 时, 接口描述被下载到 PLC。这样，用户就可以下载监视它了。

7.1.4. 问题：顺控器初始化

问题：为什么即使在输入“ OFF_SQ” 为 TRUE 时 顺控器仍在输入“ INIT_SQ” 的上升沿被激活？

解答：输入“OFF_SQ”及“INIT_SQ”信号都是上升沿有效的，也就是说它们是否有效与它们的数值（TRUE 或 FALSE）没有关系。为了防止顺控器在初始化之后自动连续工作，在关闭顺控器之前可以通过输入“SW_MAN”的上升沿切换到手动模式。在重新初始化之后，顺控器保持在初始化步骤中直到再次切换到自动模式，例如通过参数“SW_TOP”。

参数说明：这些参数都是接收到上升沿有效

“OFF_SQ” 关闭顺控器

“INIT_SQ” 初始化顺控器

“SW_MAN” 模式选择：切换到手动模式

“SW_TOP” 模式选择：自动或切换到下一个

7.1.5. 问题：S7-GRAPH 程序占用多少内存

问题：S7-GRAPH 程序占用多大 CPU 内存？

解答：S7-GRAPH 程序代码量的大小与编辑器的设置有关，在编辑器的设置菜单下的块设定中，用户可以定义顺控程序的功能块号和所用内存大小。当使用标准功能块 FC70, FC71, FC72 或者 FC73 时，S7-GRAPH 的部分系统功能就会导入到这些功能块中，此种方式大约会使 S7-GRAPH 程序减少 5000 字节。但所用的这些标准功能块调用时同样占用内存空间。如果一个项目中有多个 S7-GRAPH FB 程序，并且这些 S7-GRAPH 程序被定义成使用标准功能块方式。那么，它们公用的系统功能都将通过重复使用标准功能来实现。而这些标准功能只需生成一次并下载即可，这样就节省了内存空间。

可以使用下列公式检测 S7-GRAPH FB 块和 S7-GRAPH DB 块所占内存的大小。

- FB 的内存需求：150 字节 + $n * 16$ 字节
- DB 的内存需求：236 字节 + $n * 26$ 字节

n: 步的数量

以下例子给出了如何确定两个各有 20 个步和转换条件的顺控程序在两种不同模式下所需内存的例子。例子只是粗略计算。

编 译 器 设 置 为 : “ Full Code ” 模 式
S7-GRAPH FB 以及它的背景数据块 (MC7 代码) 内存需求计算如下:

- FB 块的内存需求: 4900 字节 (管理共享) + 130 字节每步 + 转换条件字节数 (根据内容)

- DB 块的内存需求：270 字节（管理共享）+ 70 字节每步 + 转换条件字节数（根据内容）

计算结果：

- FB 块的内存需求约为： $4,900 + 20 \times 130 = 7,500$ 字节
- DB 块的内存需求约为： $270 + 20 \times 70 = 1,670$ 字节

编译器设置为：“标准功能块”（S7-GRAPH V4.0 或以上版本）

这时顺控程序所需内存计算如下：

- FB 块的内存需求约为： $150 \text{ 字节} + 20 \times 16 \text{ 字节} = 470 \text{ 字节}$
- DB 块的内存需求约为： $236 \text{ 字节} + 20 \times 26 \text{ 字节} = 756 \text{ 字节}$

管理共享功能被导入标准功能块 FC70, FC71, FC72 和 FC73。例如，FC71 被 FB7 调用和 FC73 被 S7-GRAPH FB 调用，这时 CPU 有额外的内存需要：

- 8150 字节（对于 FC73）
- 7750 字节（对于 FC71）

7.1.6. 问题：如何监控一系列步的时间

问题：如何通过同步分支的帮助来监控一系列步的时间？

解答：在下面的例子中，可以通过同步步 S5 来监控步 S2 及 S3 的时间。如果步 S2 及 S3 的过程时间超过了步 S5 的监控定义时间，步 S5 进入故障模式。

在同步分支处，转换条件使能步 S2 及 S5。步 S2 及 S3 组成的序列与同步步 S5 同时执行，并在转换条件 T3 处结束。为了防止达不到步 S5 中编程的监控时间，当同步步结束后，转换条件 T3 连接到下一个步 S4 上。

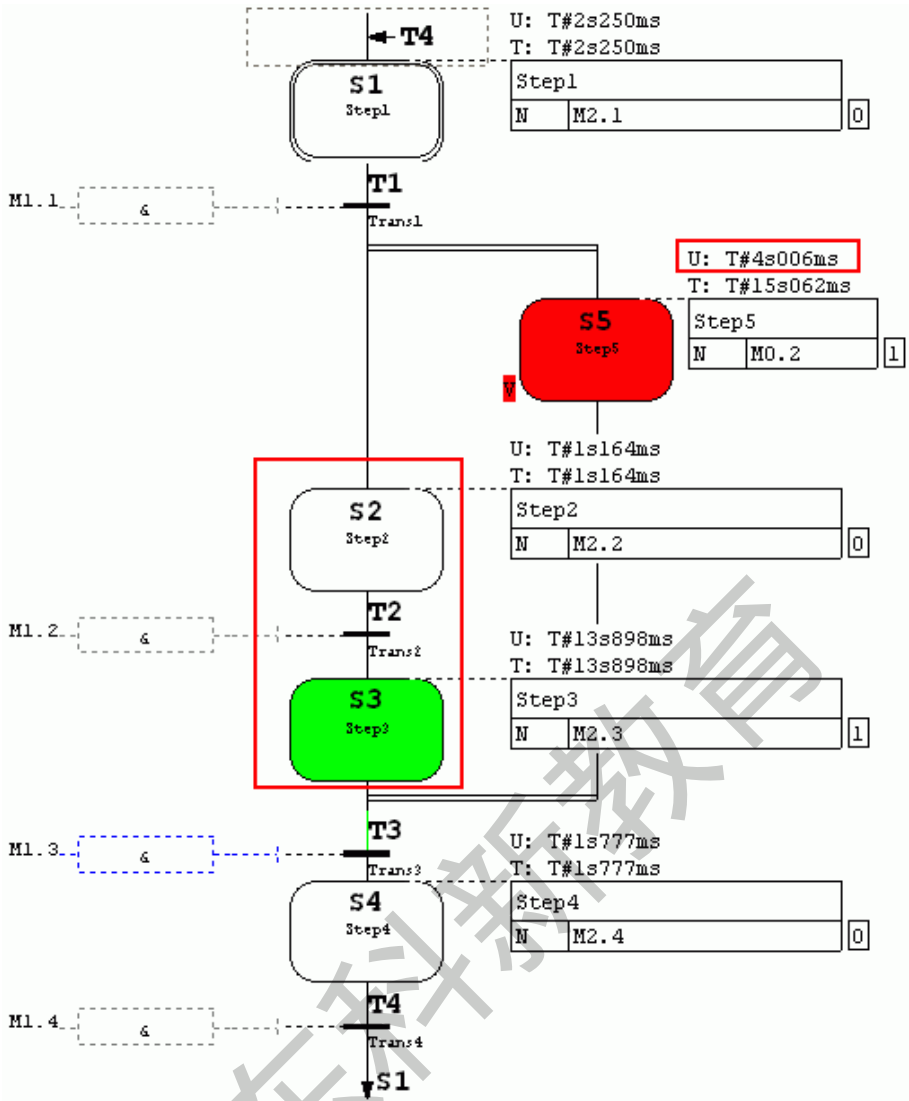


图 7-2: 同步分支用于监控时间

为了编程步 S2 及 S3 的监控时间总和，步 S5（同步分支）定义了监控条件。图中展示了步 S5 单步视图下“U”编程的监控时间，步 S5 的监控时间为 4s。输入参数“ACK_EF”可以用来确认错误，其被指定为标志 M0.2。

当监控时间及等待时间被编程后，可以改变默认的时间值来适应自动控制任务。参数“SiT”及“SiU”默认值可以在 S7-GRAPH V5.2 以上版本被指定。监控时间的默认值可以通过来菜单“Options > Application Settings... > Editor Tab”输入

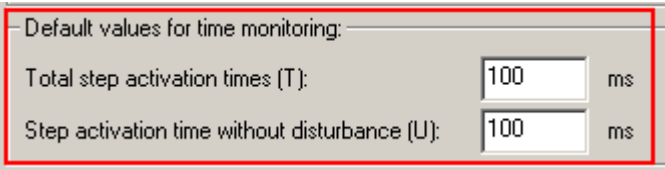


图 7-3: 更改监控时间设置

7.1.7. 问题：无法监控步的激活时间

问题：为什么无法监控步的激活时间(“ STEPi.U”)?

解答：如果无法监控步的激活时间(“ STEPi.U”)，那么很可能在“ Compile / Save” 栏中的块设置中选择了 FC73 而不是 FC72。虽然 FC73 可明显地减少 S7-GRAPH FB 的内存需求，但是这种方式生成的块不支持诊断，并且在监控运行系统控制时只有状态显示。因此，激活时间“ U” 也不可用。

解决方法：通过菜单 Options > Block settings >Compile / Save>Executability 下指定标准 FC72。就把接口描述从“ Memory minimized” 更改为“ Structure arrays” 并以 OK 确认。重新编译 S7-GRAPH FB 并将块加载到 CPU 中。

7.2. 与使用技巧相关问题

7.2.1. 问题：如何使用背景数据块内部的步数信息

问题：如何使用背景数据块内部的步数信息？

解答：顺控器中的每一步都会在 S7-GRAPH FB 的背景数据块中生成一个结构数据，步的信息被记录在此结构当中，此结构的内部数据可以通过符号名的方式进行访问，结构体中数据区的索引为步的号码，用户不要去修改此数值。

- G7S[1].X 用于显示步是否被激活
- G7S[1].SN0 用于显示用户步数

例如当步 1 被激活时，可以通过如下指令访问步 1 的信息：

A	FB2.G7S [1].X	
=	M0.0	//M0.0 存储当前步 1 是否激活的状态
L	FB2.G7S [1].SN0	
T	MW2	//MW2 储存当前激活步的序号

在菜单命令“Options > Block settings > Compile/Save”中，激活“ Structure arrays” 选项，如果此选项被激活，结构体将包含步的数组，并在存盘和编译后生成。

- 数据区名称: G7S
- 字节长度: (步的数量) * 32

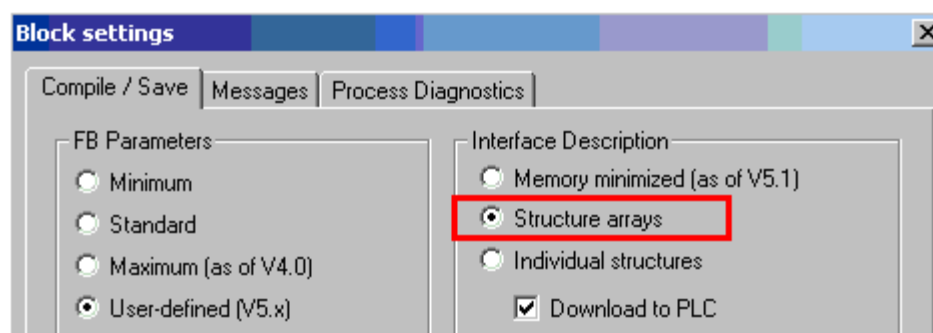


图 7-4: 激活“ Structure arrays” 选项

7.2.2. 问题：如何使用背景数据块内部的转换条件的信息

问题：如何使用背景数据块内部的转换条件的信息？

解答：在背景数据块中每个转换条件的信息都存储在一个结构当中。这一结构包括含转换条件的参数“ TV”，“ TT” 和“ TS”。只要相关步骤活动，这一转换条件传输数据就是当前的。

- TV：转换条件有效(类型：B00L)
- TT：转换条件为真(类型：B00L)
- TS：转换条件切换(类型：B00L)。

内部访问此结构(在 S7-GRAPH FB 内部)或从 STEP 7 块(LAD/STL/FBD 编辑器)进行外部访问格式如下：

- T002.TT
内部访问是通过转换条件编号及紧跟的结构单元名称进行的。
- G7T [2].TT
外部访问是通过转换条件编号及紧跟的结构单元名称进行的。

其它信息请参考问题：如何使用背景数据块内部的步数信息的解答。

7.2.3. 问题：如何在自动模式中重新激活上次的活动步

问题：在从手动模式切换到自动模式后，如何在自动模式中重新激活上次的活动步？

解答：顺控器的状态被保存在其分配的背景数据块中，通过对背景数据块的处理，可以实现以上要求。

1. 当前激活的步被存储于背景数据块的域(ARRAY) “ SAX” 中。为了保存当前激活的步的信息，必须在设置顺控器为手动模式前保存该数组。可以使用系统功能 SFC 20“ BLKMOV ” 来完成此功能

```
CALL "BLKMOV" //在 DB2 变量中标记 SAX field 并保存_SAX ARRAY[1..50]
```

```
SRCBLK                                     := "MY_DB1".SAX  
RET_VAL:=MW66  
DSTBLK := "MY_DB2".Save_SAX
```

2. 如果要重新激活保存的步，那么应当先将顺控器中所有的步去激活（deactive）。然后使用系统功能 SFC 20 把刚才保存的数据复制到背景数据块的“SONX”区域。

```
CALL      "BLKMOV"      //      复      位      标      记      步      骤  
SRCBLK:                                     := "MY_DB2".Save_SAX  
RET_VAL:=MW66  
DSTBLK:  := "MY_DB1".SONX
```

3. 在重新激活之前，使用“OFF_SQ”信号来停止依然活动的所有步。在保存活动步和随后的再激活之间的时间内不必对顺控器作任何改变。

注意：复制之后，原来的步就会被立即激活。

实例数据块中数组“SAX”和“SONX”的长度取决于顺控器的结构（并行步）。用于保存“SAX”域的数组必须足够大。

注意事项：当改变操作模式从自动到手动并且再次回到自动时，如果顺控器没有改变，那么顺控器会自动同步到先前的活动步。

7.2.4. 问题：如何实现忽略监控干扰而实现顺控器到下一步的转换

问题：如何实现忽略监控干扰（Supervision Disturbance）而实现顺控器到下一步的转换？

解答：如果在 S7-GRAPH 的操作中发生一个监控干扰（Supervision Disturbance），即 Supervision=TRUE。那么该监控干扰必须通过输入参数“ACK_EF”确认。确认后顺控器才会继续运行。下表解释了使用或不使用参数“ACK_EF”进行确认的要求。

在 S7-GRAPH 菜单中 “Options > Block Settings> Compile / Save>Sequencer Properties”，确定启用 “Acknowledge errors” 功能。

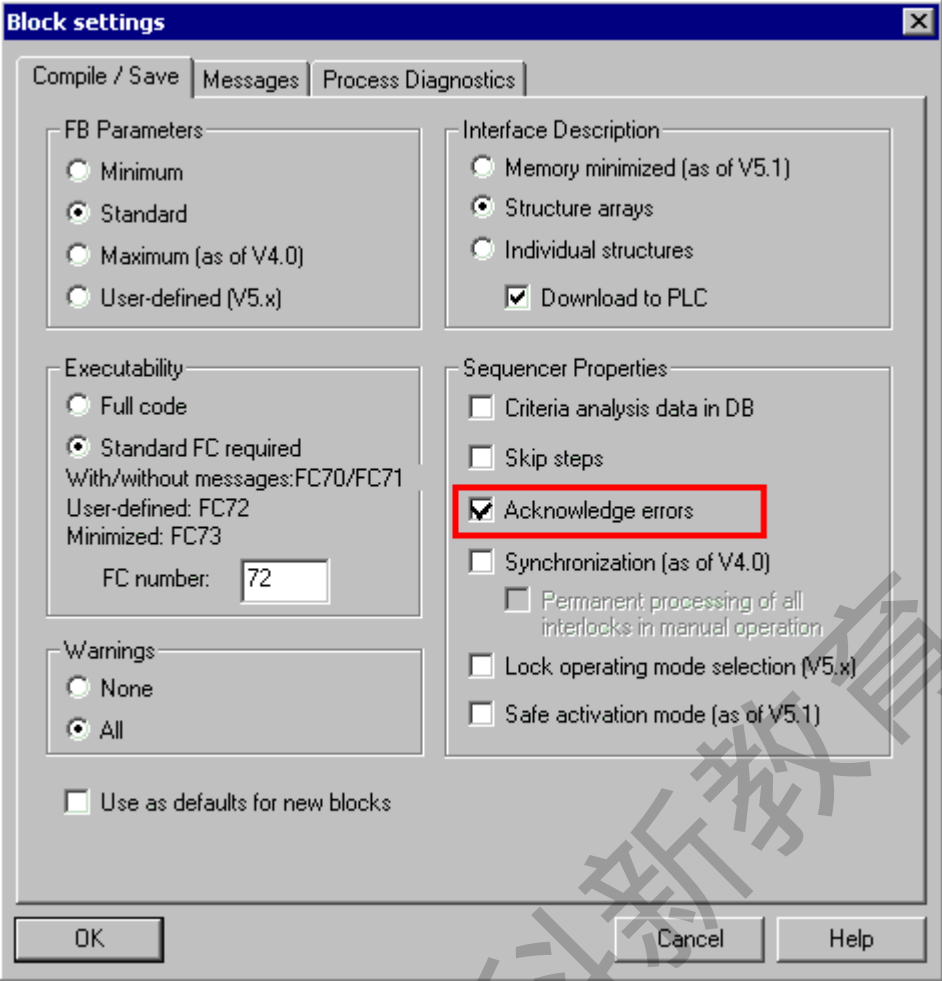


图 7-5: 启用 “ Acknowledge errors” 功能

但是，如果已经为“ FB Parameters” 启用了选项“ Mini mum” 就不能选择该功能。因此应首先为 “ FB Parameters” 启用“ Standard”，“ Maxi mum” 或“ User-defi ned”

如果不使用以上的方法，在下面的例子程序中，也可以通过程序逻辑屏蔽 Supervi si on 信号实现同样的功能。

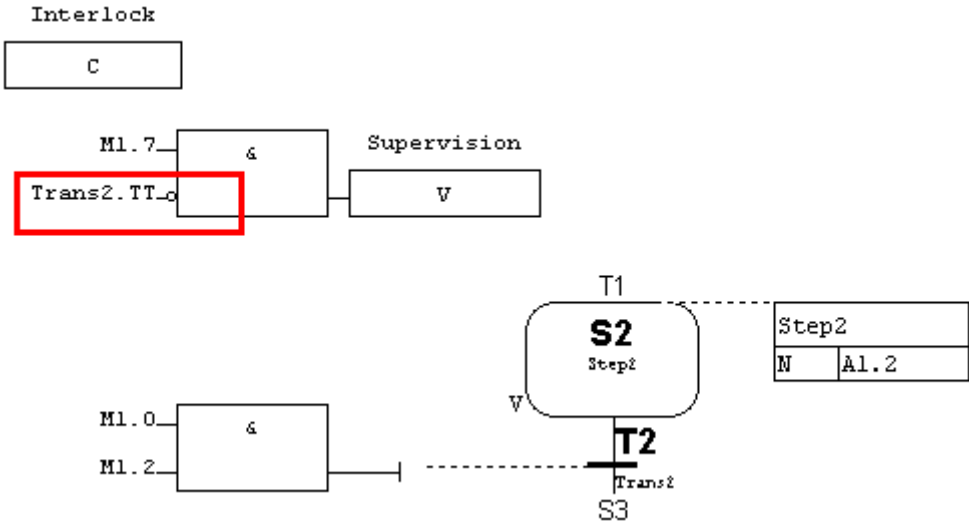


图 7-6: 程序逻辑屏蔽 Supervision 信号

程序分析:

当 M1.7=1 时，并且 M1.0 和 M1.2 也为“1”，此时，步可以向下一步转换，而监控干扰（Supervision Disturbance）条件不满足。然而请注意在该程序中，必须禁用“Acknowledge errors”功能。

7.2.5. 问题：如何确定 S7-GRAPH 顺控器的故障步

问题：如何确定 S7-GRAPH 顺控器的故障步？

解答：S7-GRAPH 顺控器的 FB 参数存储于实例 DB 中，而且 FB 参数的存储器要求取决于已经在表“Compile / Save”的菜单“Options > Block Settings”中的设置。这里设置“FB Parameters”为“User-defined (V5.x)”。

在对 S7-GRAPH 顺控器进行编译和保存后，就拥有了最大的参数集可用于在块 (FB, FC or OB1) 中调用 S7-GRAPH FB。现在，就可以通过一些参数来确定顺控器的状态或故障步。

输出参数:

- S_NO: 显示步编号
- S_MORE: 可在 S_NO 中显示更多的活动步
- S_ACTIVE: 显示步为活动
- SEF_DISP: 只显示故障步的列表
- ERROR: 任意步中的互锁错误
- FAULT: 任意步中的监控错误

输出参数:

- DISP_SEF: 只显示故障步

相关更多信息可以参考 S7-GRAPH 手册中关于 STEP Structure 章节。

7.2.6. 问题：为什么同时满足转换条件和监控的条件时，转换条件会切换到下一步

问题：为什么同时满足转换条件和监控的条件时，转换条件会切换到下一步？

解答：在 S7-GRAPH 当中，对监控条件（Supervision）的描述如下：

- 如果监控条件（Supervision）满足，则事件 V1 发生，顺控器不再转换的下一步，当前步保持激活，步的激活时间 Si.U 停止。
- 如果监控条件（Supervision）不满足，并且当前步向下一步的转换条件满足，顺控器将转换到下一步。

但需要注意的是：在一个周期中，首先进行的是切换过程，然后是状态中的保留处理。因而在一个周期中顺控器总是企图尽早地切换，也就是说，如果下图中的步 2 中的转换和监控同时满足，则立即进行步的转换。

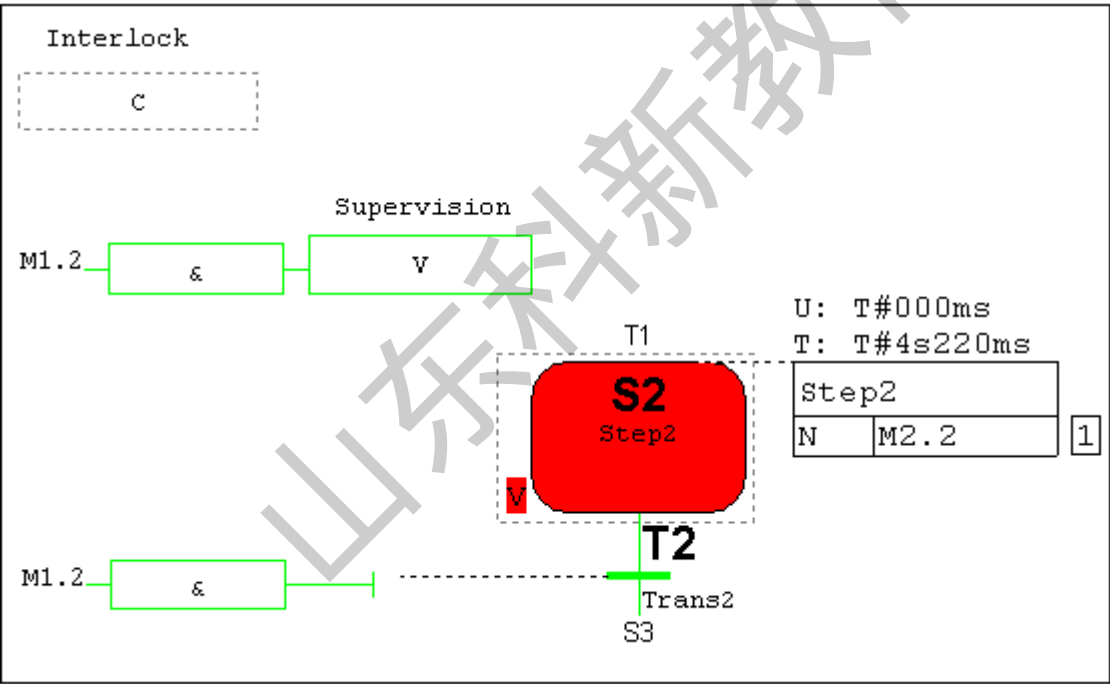


图 7-7: 步的转换

解决方法：使用不同的参数编程监控和接通，例如：

- M1.2 用于监控，并且
- M1.3 用于转换条件检测

7.2.7. 问题：为什么尽管互锁条件没有完成，在离开步的时候动作“ L0 S” 被执行？

问题：为什么尽管互锁条件没有完成，在离开步的时候动作“ L0 S” 被执行？

解答：在 S7-GRAPH 中，一个动作可以根据互锁状态信号的改变而被触发。在事件“ L0 S” 的情况下，此指令的操作数被置位为 1，并且一旦对于一个激活的步的互锁条件从信号状态 0 改变为 1 时，此动作即被触发。（互锁条件到来，错误离去）。

S7-GRAPH 的一个基本规则：没有激活的一个步不能引起任何错误。因此，如果当一个无法执行的互锁条件在离开步时被置为“完成的”（错误被强制确认），则当此步离开时，“ L0 S” 指令总是被执行。

7.2.8. 问题：如何将无条件同时分支改为条件分支时不需重新输入所有受影响的步

问题：如何将无条件同时分支改为条件分支时不需重新输入所有受影响的步？

解答：首先在菜单中 选择 “ View > Sequencer”，以察看整个顺控器。点击右键在弹出菜单中选择命令“ Insert New Element > Alternative Branch > Open”。这就可以插入一个条件分支。

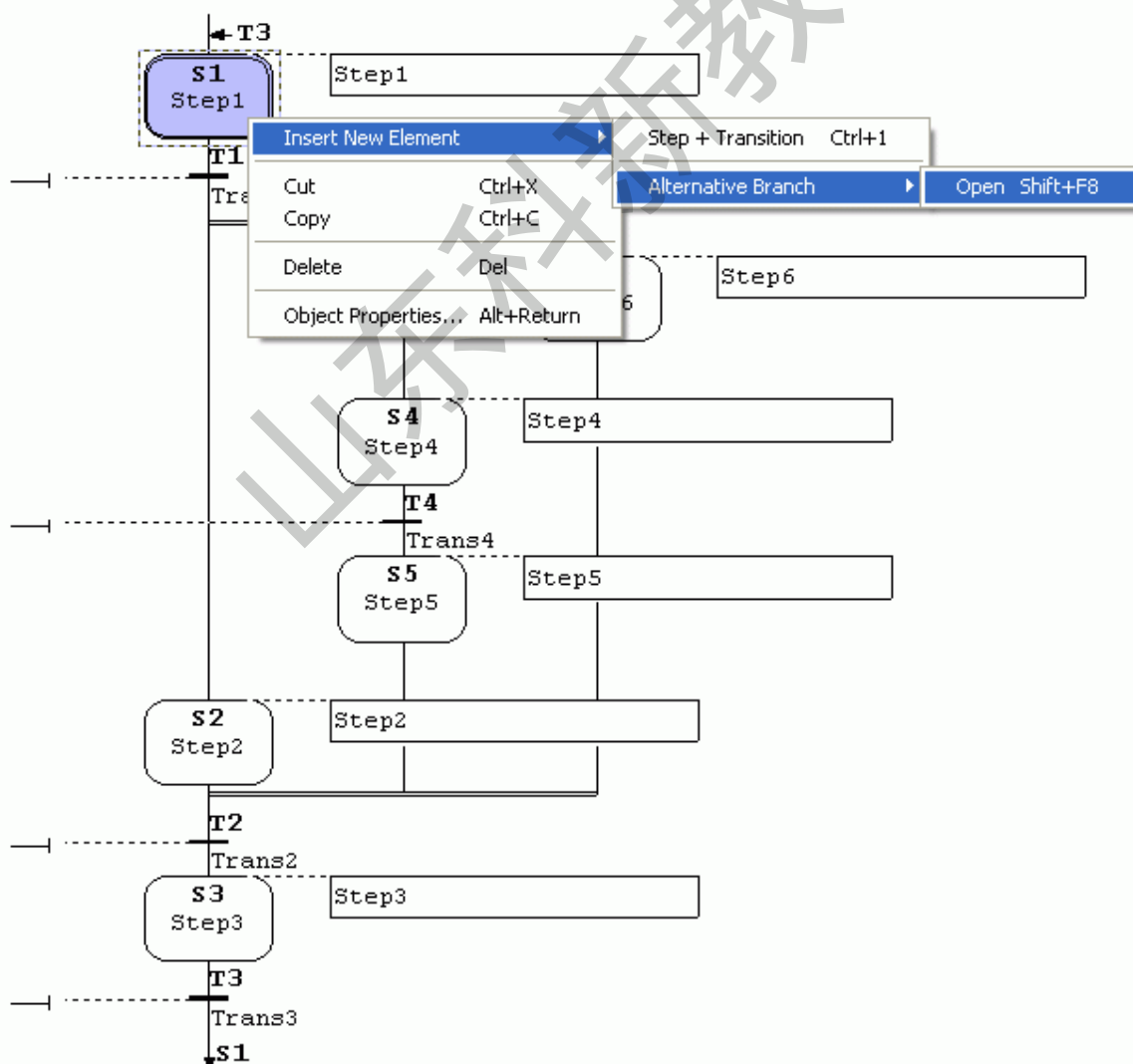


图 7-8: 插入一个条件分支

在刚创建的条件分支处插入选中的转换条件剪切的顺控器部分，并执行菜单命令“ Edit > Paste”。

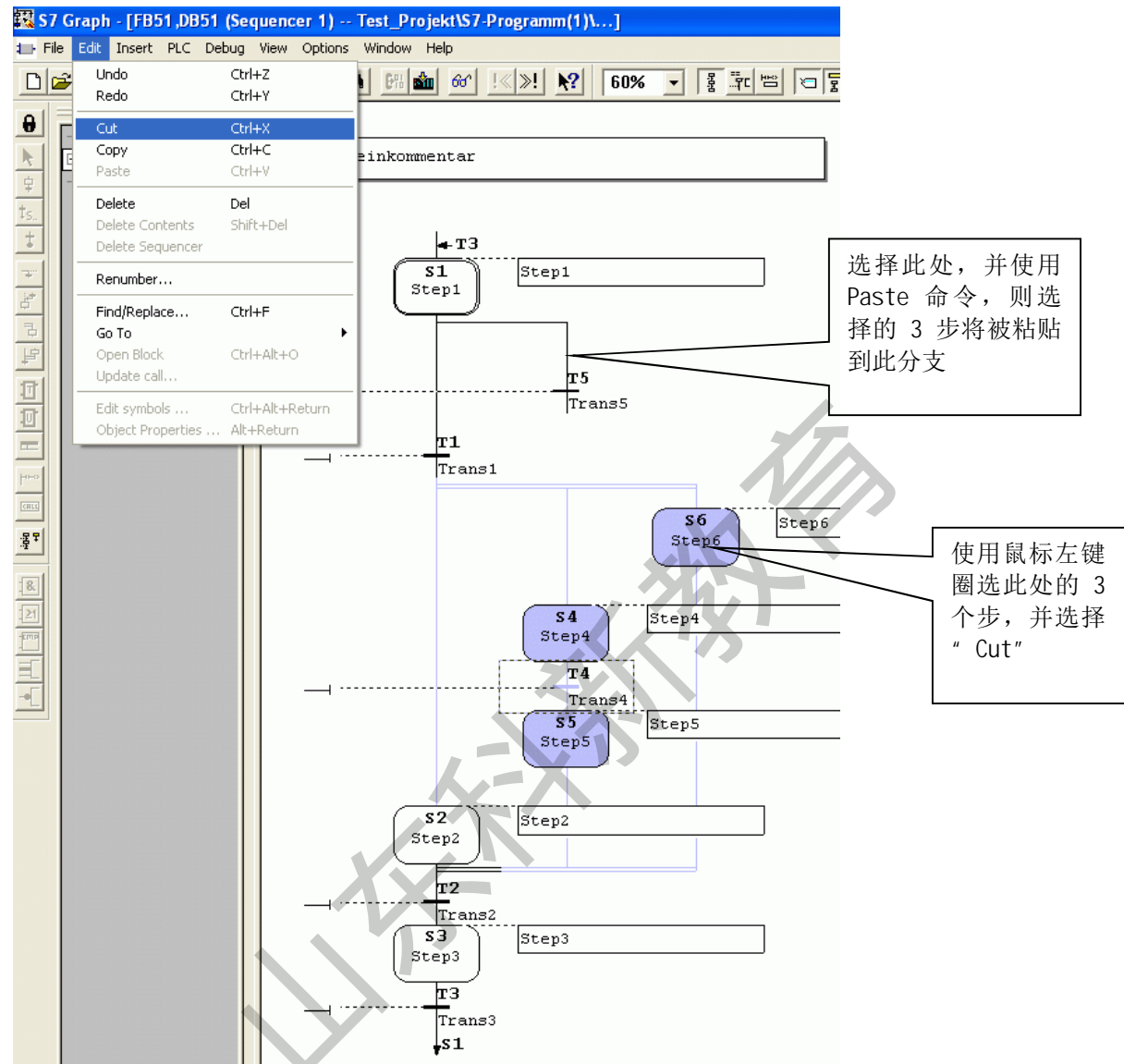


图 7-9: 粘贴一个条件分支

7.2.9. 问题：如何在第一个 OB1 周期中或第一次调用 S7-GRAPH FB 时关闭顺控器

问题：如何在第一个 OB1 周期中或第一次调用 S7-GRAPH FB 时关闭顺控器？

解答：通过参数 OFF_SQ 外部控制顺控器 FB 参数“ INIT_SQ” 和“ OFF_SQ” 可用来初始化和关闭顺控器，前提是确保在顺控器编译前以下路径下的：

“ Options > Block settings... > Compile/Save > FB Parameters 下的参数“ Standard” 、 “ Maximum” 或“ User-defined” 之中的一个已经激活。

下图显示了对 S7-GRAPH FB 的调用，调用时，可将 M0.0 的常闭点送至 “ OFF_SQ” ，这样当程序开始扫描时，“ OFF_SQ” 参数将收到一个上升沿，并关闭顺控器。

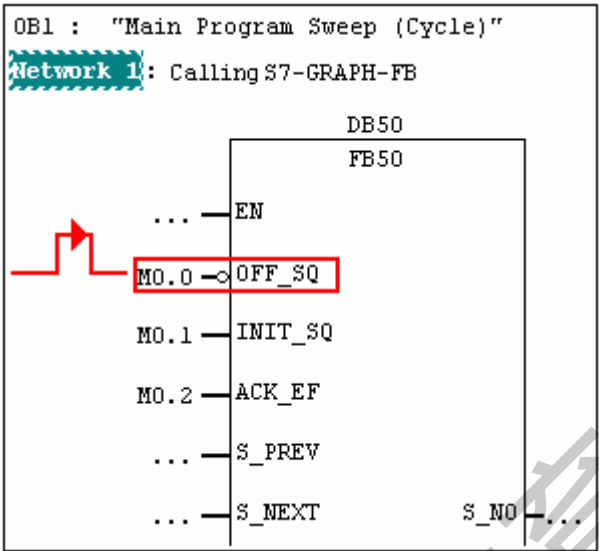


图 7-10: 关闭顺控器

通过 Supervision 来控制顺控器

下图显示了步 1(初始步) 中将 M10.3 的常闭点用于监控条件。如果 M10.3 为“ False”，则启动后，顺控器运行到监测步而且不会再切换。当在块设置中激活选项“ Acknowledge errors” 时，只需在 FB 输入参数“ ACK_EF” 进行确认后，顺控器将被继续执行。

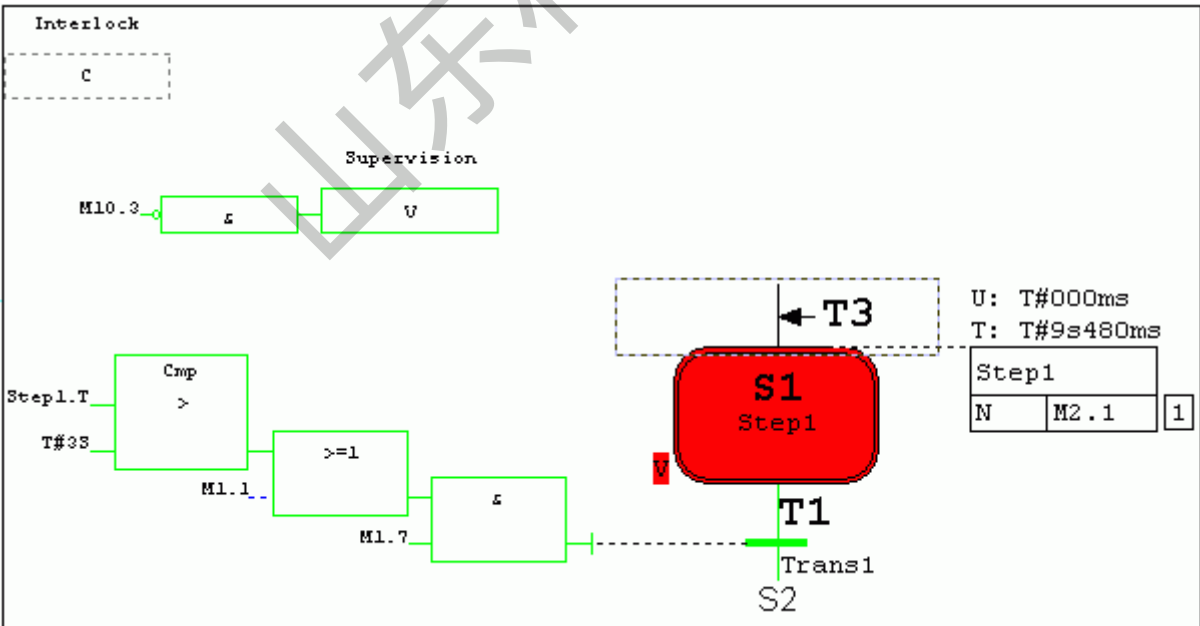


图 7-11: 通过 Supervision 来控制顺控器

7.2.10. 问题：为什么计数器比较功能产生一个错误结果

问题：为什么计数器比较功能产生一个错误结果？

解答：由于 S7-GRAPH 计数值是 BCD 码的格式, 那么输入的比较值也必须是 BCD 码的格式(3 位十进制 BCD), 并应当遵循下列语法格式: C#000 到 C#999。

7.2.11. 问题：为什么在顺控器中无法通过程序激活另一个步

问题：为什么在顺控器中无法通过程序激活另一个步？

解答：这是初学者经常犯一个错误, S7-GRAPH 中提供了一些指令, 可以在顺控器当中根据用户意图激活顺控器中的某一个步。但是务必要注意的是: 在程序调转到被激活步之前, 务必要增加关闭当前步的指令, 否则顺控器不会执行步的切换。

7.2.12. 问题：如何加密 S7-GRAPH 块

问题：如何加密 S7-GRAPH 块？

解答：在 STEP7 中, 只有 STL 源文件格式支持块保护功能, 仅仅有 S7-GRAPH 源文件无法实现块保护功能。因此可以通过由 S7-GRAPH 源文件生成 STL 格式源文件的方式实现块保护功能。

可以按照如下步骤实现：

1. 首先将 S7-GRAPH FB 备份为 S7-GRAPH FB 源文件
 - 打开 S7-GRAPH FB, 并通过“File > Generate Source File...”创建 S7-GRAPH 源文件。
 - 在打开的对话框中输入对象名称, 例如“Source_FB”。
 - 用 OK 键进行保存, 生成了 S7-GRAPH 源文件后, 关闭 S7-GRAPH FB。
2. 生成 STL 格式的 FB
 - 在 Windows 操作系统中, 通过“Start > SIMATIC> STEP 7 > LAD, STL, FBD-Programming S7 Blocks”打开 LAD/STL/FBD 编辑器。
 - 在 LAD/STL/FBD 编辑器中, 通过“File > Open”选择 STEP 7 项目, 打开在 S7-GRAPH 中创建的块(例如 FB51)。
 - 选择菜单中“File > Save”, 此时显示信息提示: 此块是在不同的编程语言中创建的, 保存后则不能再用其原始创建语言(S7-GRAPH)进行编辑。点击“ Yes”按钮。则在 S7-GRAPH 中创建的功能块, 现在已经变为 STL 语言格式了。
3. 生成 STL 格式的源文件
 - 在 LAD/STL/FBD 编辑器中通过“File > Generate Source...”创建 STL 源文件
 - 在打开的对话框中输入对象名称, 例如“Protect_FB”。
 - 选中要为其创建 STL 源文件的块(例如 FB51), 并将这个块移到右边的“Blocks Selected”窗口中。

- 以 OK 确认，生成 STL 源文件，关闭 LAD/STL/FBD 编辑器。
4. 现在在“ Sources” 文件夹中有两个源文件。
- STL 源文件(Protect_FB)，
 - S7-GRAPH 源文件(Source_FB)，
5. 生成加密的 FB
- 打开 STL 源文件 “ Protect_FB”，并在“ TITLE” 下的块的声明段插入关键字
KNOW_HOW_PROTECT
 - 通过“ File > Save” and “File > Compile” 保存并编译 STL 源文件。则 FB51 被加密。