

SIEMENS

西门子编程工具简介

Brief Introduction of the Programming Tools of Siemens

Getting Started

Edition (2007 年 8 月)

摘要

本文档主要用于讨论以下相关问题：

- ✧ 西门子基本编程工具的选择及比较
- ✧ 西门子工程工具的选择及比较
- ✧ 西门子编程工具的简单使用

关键词

编程工具； Step7； 梯形图； 语句表； 功能图； 工程工具； SCL； GRAPH； HiGraph； CFC； PLCSIM

Key Words

Programming tools； Step7； LAD； STL； FBD； Engineering tools； SCL； GRAPH； HiGraph； CFC； PLCSIM

目 录

西门子编程工具简介	1
1. 前言	7
2. 虚拟工程介绍	7
2.1. 重要提示:	8
2.2. 重要声明:	8
3. 标准编程工具	9
3.1. Step7 BASIC	9
3.2. Step7 Lite	9
3.3. Step7 professional (专业版)	10
3.4. Step7 中的基本编程语言	10
3.4.1. 概括	10
3.4.2. 基本编程语言与虚拟工程	10
3.4.3. 梯形图 (LAD)	10
3.4.4. 功能块图 (FBD)	12
3.4.5. 语句表 (STL)	14
3.4.6. STL / LAD / FBD 格式的转换	15
3.4.7. LAD/FBD/STL 之比较	15
4. 工程工具	15
4.1. 重要提示:	15
4.2. S7-SCL	15
4.2.1. S7-SCL 简介	15
4.2.2. SCL 与 STL:	16
4.2.3. SCL 的安装与使用:	16
4.2.4. SCL 应用于虚拟工程	16
4.2.5. SCL 简单示例	16
4.2.6. 重要提示:	26
4.3. S7-GRAPH	27
4.3.1. S7-GRAPH 简介	27
4.3.2. S7-GRAPH 与 STL:	27
4.3.3. S7-GRAPH 安装与使用:	27
4.3.4. S7-GRAPH 应用于虚拟工程	27
4.3.5. S7-GRAPH 简单示例	30

4.3.6. 重要提示:	38
4.4. S7-HiGraph	38
4.4.1. S7-HiGraph 简介	38
4.4.2. HiGraph 与 STL:	38
4.4.3. S7-HiGraph 的安装与使用:	39
4.4.4. S7-HiGraph 应用于虚拟工程	39
4.4.5. S7-HiGRAPH 简单示例	40
4.4.6. 重要提示:	51
4.5. S7-CFC	51
4.5.1. S7-CFC 简介	51
4.5.2. CFC 与 SCL 及 STL:	51
4.5.3. CFC 的安装与使用:	51
4.5.4. CFC 应用于虚拟工程	52
4.5.5. CFC 简单示例	52
4.5.6. 重要提示:	60
4.6. S7-PLCSIM	60
4.6.1. S7-PLCSIM 简介	60
4.6.2. PLCSIM 与真实 PLC:	61
4.6.3. PLCSIM 安装与使用:	61
4.6.4. S7-PLCSIM 用于虚拟工程	62
4.6.5. S7-PLCSIM 简单示例	62
4.6.6. PLCSIM 主要功能:	63
5. 总结	65

图片列表

图1: 虚拟工程工艺流程	8
图2: 典型电机启动控制电路	11
图3: 电机启动控制电路LAD 程序	12
图4: 电机启动控制电路FBD 程序	13
图5: 电机启动控制电路STL 程序	14
图6: 新建SCL Source	17
图7: 使用FB模板	17
图8: 更改FB编号	18
图9: 添加FB 输入输出参数	18
图10: 添加FB 输入输出参数	19
图11: 使用调用功能块向导	20
图12: 调用FC105	20
图13: SIMATIC Manager 中打开库文件	21
图14: 将FC105复制到当前项目中	21

图15: FB1参数定义.....	22
图16: FC105调用	23
图17: 使用IF 模板.....	23
图18: 编写滤波程序.....	24
图19: 生成调试信息.....	24
图20: 监控OB35	25
图21: 监控FB1	26
图22: 原料配比示意图	28
图23: 原料配比工艺流程图.....	29
图24: 添加功能块.....	30
图25: 选择功能块类型	31
图26: 编程元素	31
图27: 编程分支及流程	32
图28: 初始化.....	33
图29: 编程环境视图.....	33
图30: 流程分支	34
图31: 流程结束	34
图32: 参数设置	35
图33: 调用FB2	35
图34: 监控FB2	36
图35: 测试FB2.....	36
图36: 温度超高互锁.....	37
图37: 顺序控制功能.....	37
图38: 产品成型系统构成.....	39
图39: 成型机工艺流程	39
图40: 机械手工艺流程	40
图41: 成型机输入参数	41
图42: 成型机输出参数	41
图43: 编辑成型机消息	42
图44: 成型机静态变量参数	42
图45: 添加状态及条件	43
图46: 添加各种状态下的指令.....	43
图47: 成型机程序概览	44
图48: 机械手程序概览	45
图49: 分配成型机接口参数	46
图50: 分配机械手接口参数	47
图51: 设置GRAPH参数	48
图52: 监控graph group.....	48
图53: 监控state graph.....	49
图54: 选择监控变量	49
图55: 监控变量	50
图56: HiGraph应用例子.....	50
图57: 添加CFC程序.....	52
图58: 打开CFC	53

图59: CFC页面视图.....	53
图60: CFC当前所处位置.....	53
图61: CFC调用其它功能块.....	54
图62: 分配地址连接.....	54
图63: 内部连接.....	55
图64: 调用CFC程序库.....	55
图65: 调用STEP7标准程序库.....	56
图66: 模拟量输入滤波.....	56
图67: 手动/自动切换无扰动处理.....	57
图68: 输出处理.....	57
图69: 程序图概览.....	58
图70: 编译/下载设置.....	58
图71: 编译程序.....	59
图72: 监控程序.....	59
图73: 监控变量.....	60
图74: PLCSIM 调用.....	62
图75: PLCSIM 视图.....	63
图76: 录制事件.....	64
图77: 事件回放.....	64

表格列表

表 1: STEP7 Lite 与 STEP7 标准版的主要区别.....	9
表 2: LAD/FBD/STL之比较.....	15

重要提示: 本文为技术交流文档, 不能作为订货、选型等重要事宜的唯一依据, 建议您参考 Siemens 的标准产品样本和技术手册进行产品的选型和订货。

1. 前言

本文将对西门子编程工具作简单的介绍。由于任何一种编程工具或语言都不是一朝一夕就可以掌握的，所以本文的目的仅仅是让用户对西门子编程工具有一个总的基本概念，对于每种编程工具的详细使用及功能请参考相关的手册。

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>

S7-300 和 S7-400 的功能块图 (FBD) 编程

<http://support.automation.siemens.com/CN/view/zh/18652644>

使用 STEP 7 V5.3 编程

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

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

<http://support.automation.siemens.com/CN/view/zh/5581793>

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

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

S7-HiGRAPH V5.3

<http://support.automation.siemens.com/CN/view/zh/1137299>

S7-CFC for SIMATIC S7

<http://support.automation.siemens.com/CN/view/zh/24451339>

SFC for SIMATIC S7

<http://support.automation.siemens.com/CN/view/zh/24451748>

2. 虚拟工程介绍

为了在本文中的描述中更有利于用户理解，我们列举一个虚拟的工程项目，此项目包括简单的工艺要求及流程。针对各个工艺要求及流程，我们使用不同的编程工具来处理，并且通过简单的程序例子，使大家对各种编程工具有一个比较直观的认识。

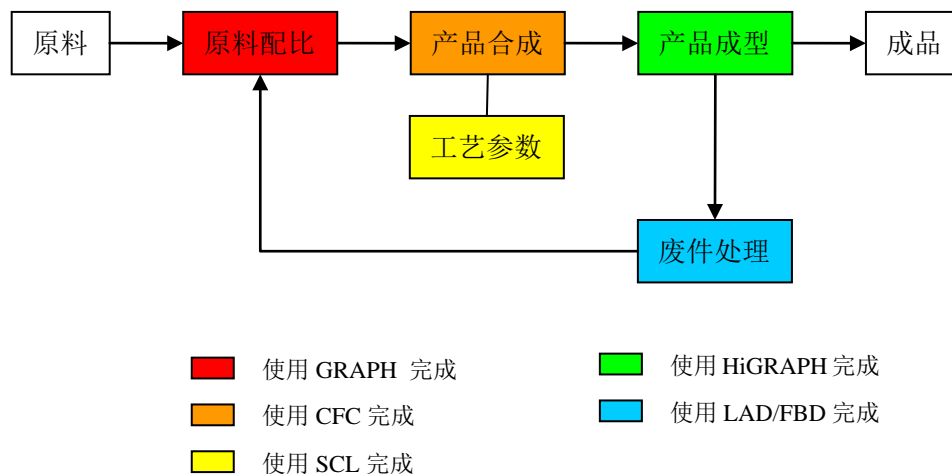


图 1: 虚拟工程工艺流程

2.1. 重要提示:

- ◇ 各种编程工具最终生成的执行代码均为STL，所以其并未在图 1 中出现。
- ◇ 图 1 中在各个工艺流程中使用了最适合的编程工具，并不意味着这种编程工具仅能用于此类控制任务。例如：SCL 可以用于工艺参数部分，但它也可以用于其它工艺段。
- ◇ CFC不仅仅可以用在产品合成阶段，也可用于将整个项目所有流程组织在一起。
- ◇ 假设虚拟项目使用的硬件系统为 CPU412-2
- ◇ 由于S7-PLCSIM为仿真工具，其可用于所有过程的仿真调试，所以其并未在图 1 中出现。

2.2. 重要声明:

- ◇ 本文的虚拟工程与真实工程实例有重大差别，请读者切勿将其与工程实例混淆。
- ◇ 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。

3. 标准编程工具

3.1. Step7 BASIC

STEP7 基本软件是用于 SIMATIC S7, SIMATIC C7, SIMATIC WinAC 自动化系统的标准工具。它的基本功能包括:

- ◇ 组态和参数化硬件
- ◇ 定义通信
- ◇ 编程
- ◇ 测试调试及服务
- ◇ 编制文件及归档
- ◇ 操作及诊断功能

STEP7 基本软件是大家最熟悉及使用最频繁的软件, 作为基本平台, 很多其它的软件包都是作为选件包集成安装在 STEP7 中使用的。关于 STEP7 的基本使用超出了本文范围, 故不再介绍。

3.2. Step7 Lite

STEP7 Lite 是 STEP7 的简化版, 界面简单直观, 适合于初学者使用, 它有很多使用功能限制, 如不能编程网络功能等等。它编辑的项目可以被 STEP7 及 STEP7 professional 打开。STEP7 Lite 与 STEP7 标准版的主要区别见下表:

工具 比较项目	STEP 7 Lite	STEP 7
组态的目标系统	S7-300/C7	S7-300/S7-400/C7/WinAC
支持的机架	数字量, 模拟量 I/O, IFM 仅支持中央机架	中央机架及分布式 I/O
网络/通信	不支持	支持
分布式 I/O	不支持	支持
组态消息 (用于 HMI 显示)	不支持	支持
读写 MMC 卡	只能在 CPU 中读写	在 CPU 中读写, 也支持 PG/PC 读写方式
导入/导出	程序, 符号表	程序, 符号表, 硬件组态
在线访问	MPI	MPI, PROFIBUS, Industrial Ethernet
操作系统	Windows XP Home Windows 2000 Professional Windows XP Professional	Windows 2000 Professional Windows XP Professional

表 1: STEP7 Lite 与 STEP7 标准版的主要区别

3.3. Step7 professional (专业版)

STEP7 professional (专业版) 适合于高级编程人员使用, 它在标准版的基础上增加了如下软件包及授权:

- ◇ S7-GRAPH
- ◇ S7-SCL
- ◇ S7-PLCSIM

3.4. Step7 中的基本编程语言

3.4.1. 概括

各个版本的 Step7 都支持的基本编程语言包括:

- ◇ 梯形图 (LAD)
- ◇ 功能块图 (FBD)
- ◇ 语句表 (STL)

3.4.2. 基本编程语言与虚拟工程

下面将以第 2 章中的虚拟工程中的废品处理环节来举例说明基本编程语言的使用。

废品处理工艺要求:

- ◇ 按下启动按钮后, 电机启动, 开始粉碎废品, 运行指示灯指示运行。
- ◇ 按下停止按钮后, 电机停止, 停止粉碎废品, 停止指示灯指示停止。
- ◇ 保护继电器保护电机运行。

3.4.3. 梯形图 (LAD)

梯形图 (LAD) 的英文全称是: Ladder Logic

由于 PLC 产生于替代继电器逻辑功能的需求, 因此其基本的编程语言梯形图也就非常接近于继电器电路, 它很容易被熟悉继电器控制的电气人员所掌握, 特别适合于数字量逻辑控制。梯形图由触点、线圈和用方框表示的指令构成。触点代表逻辑输入条件; 线圈代表逻辑运算结果, 控制的指示灯, 开关和内部的标志位等; 指令框用来表示定时器、计数器或数学运算等附加指令。在程序中, 最左边是主信号流, 信号流总是从左向右流动的。梯形图编程语言与原有的继电器控制的不同点是, 梯形图中的能流不是实际意义的电流, 内部的继电器也不是实际存在的继电器, 因此应用时, 需要与原有继电器控制的概念区别对待。

梯形图编程语言的特点是: 与电气操作原理图相对应, 具有直观性和对应性; 与原有继电器控制相一致, 电气设计人员易于掌握。下图为非常的典型电机启动控制电路的电气原理图:

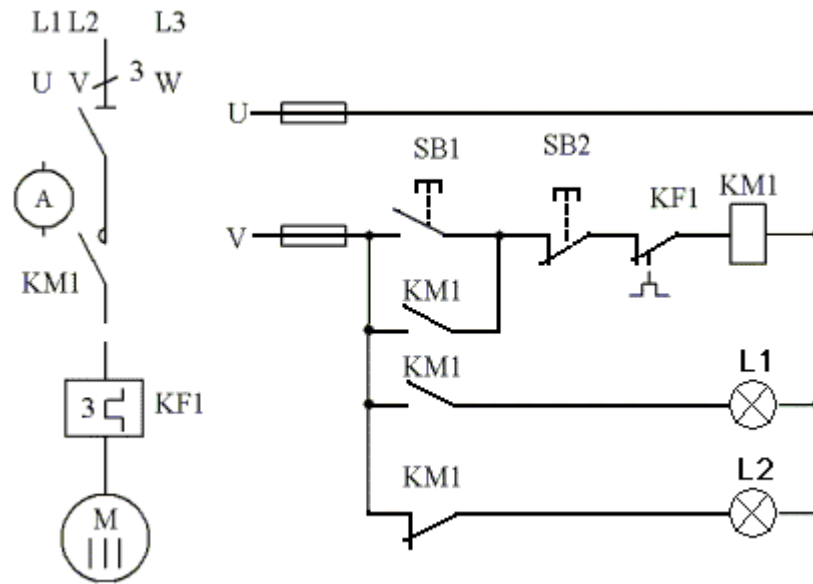


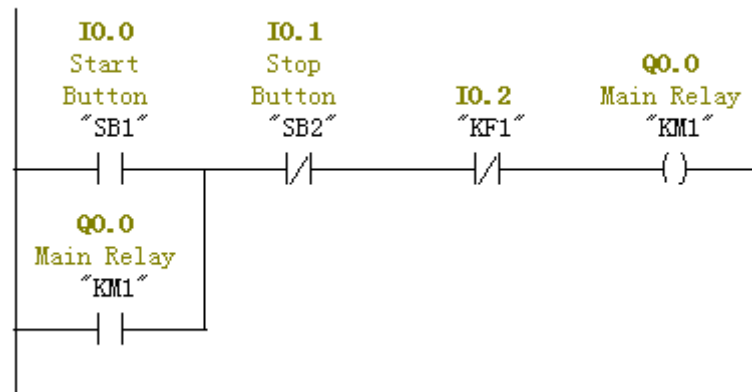
图 2：典型电机启动控制电路

图 2 符号说明：

- SB1: 启动按钮
- SB2: 停止按钮
- KF1: 保护继电器
- KM1: 主控继电器
- L1: 运行指示灯
- L2: 停止指示灯

下图为图 2 的 LAD 的编程格式，可以发现其编程类似于电气图纸，非常直观。易于调试。梯形图是一种图形化的编程界面，编程的同时进行语法检查，其转换为 STL 后的语法结构也是自动完成的。这样就减轻了编程人员的负担，方便了调试。

Network 1: Main Relay



Network 2: Light Run



Network 3: Light Stop



图 3: 电机启动控制电路 LAD 程序

梯形图编程语言的优点:

- ◇ 易于调试
- ◇ 程序易读性强
- ◇ 程序易于移植
- ◇ 易于分析逻辑关系

梯形图编程语言的缺点:

- ◇ 不适合人工编写复杂控制任务及大型程序（关于大型程序编写，请参考第4章）

3.4.4. 功能块图（FBD）

功能块图（FBD）的英文全称是：Function Block Diagram。

功能块图使用类似于布尔代数的图形逻辑符号来表示控制逻辑，一些复杂的功能用指令框表示，适合于有数字电路基础的编程人员使用。功能块图用类似于与门、或门的框图来表示逻辑运算关系，方框的左侧为逻辑运算的输入变量，右侧为输出变量，输入、输出端的小圆圈表示“非”

运算，方框用“导线”连在一起，信号自左向右。

下图为图 2 的 FBD 编程格式。可以发现其编程类似于电子电路逻辑图，非常直观，易于调试。功能块图是一种图形化的编程界面，编程的同时进行语法检查，其转换为 STL 后的语法结构也是自动完成的。这样就减轻了编程人员的负担，方便了调试。

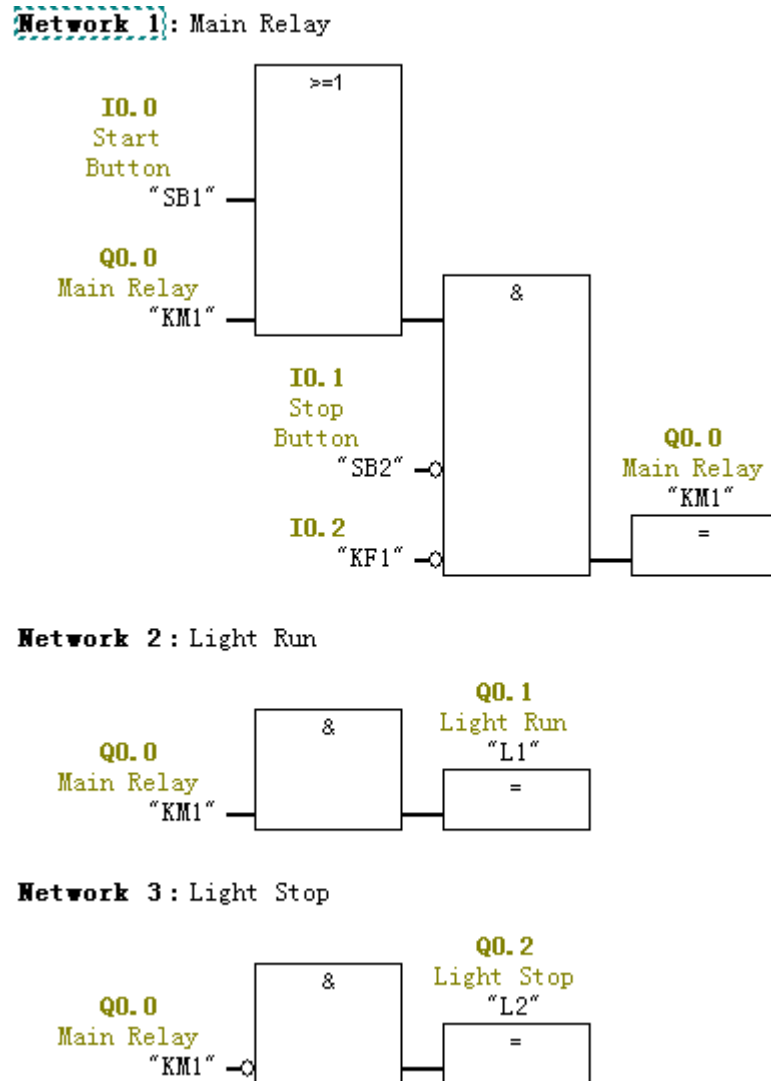


图 4：电机启动控制电路 FBD 程序

功能图编程语言的优点：

- ◇ 易于调试
- ◇ 程序易读性强
- ◇ 程序易于移植
- ◇ 易于分析逻辑关系

功能图编程语言的缺点：

- ◇ 不适合人工编写复杂控制任务及大型程序（关于大型程序编写，请参考第4章）

3.4.5. 语句表 (STL)

语句表 (STL) 的英文全称是 Statement List。

STL 一种文本编程语言，类似于微机的汇编语言，适合于经验丰富的程序员使用。由于其最接近于机器执行代码，可以直接访问 PLC 内部的各种资源，因此功能比较强大，可以实现一些其它编程工具不能实现的功能。STL 是其它各种编程工具生成代码的最终描述语言，也就是说其它各种编程工具编译后生成的代码最终都是以 STL 出现的。例如 GRAPH，虽然对于用户来讲，其编程调试界面都是图形化的，但真正在底层工作的是 STL，只是用户看不到 STL 如何工作。

下图为图 2 的 STL 编程格式。可以发现其编程风格非常简捷，完全是汇编语言的风格，但恰恰因为这一点，就要求编程人员必须对 PLC 的内部体系结构，语法规则，指令规则，数据类型，系统堆栈，状态字等等，有一个非常清晰的概念。否则，编程人员有可能在后期的调试中，面对各种困惑，程序也有可能存在隐患。

```

Network 1: Main Relay
  A(
  O   "SB1"           I0.0       -- Start Button
  O   "KM1"           Q0.0       -- Main Relay
  )
  AN  "SB2"           I0.1       -- Stop Button
  AN  "KF1"           I0.2
  =   "KM1"           Q0.0       -- Main Relay

Network 2: Light Run
  A   "KM1"           Q0.0       -- Main Relay
  =   "L1"            Q0.1       -- Light Run

Network 3: Light Stop
  AN  "KM1"           Q0.0       -- Main Relay
  =   "L2"            Q0.2       -- Light Stop
  
```

图 5: 电机启动控制电路 STL 程序

语句表编程语言的优点:

- ◇ 功能强大
- ◇ 程序代码效率高

语句表编程语言的缺点:

- ◇ 指令记忆困难
- ◇ 程序易读性差
- ◇ 不适合人工编写复杂控制任务及大型程序 (关于大型程序编写, 请参考第4章)

3.4.6. STL / LAD / FBD格式的转换

当使用 LAD/FBD 编程时，系统会自动添加严格的语法结构。而 STL 并不自动添加完整的语法结构（数据类型匹配，系统堆栈处理等）。因此 LAD/FBD 格式的程序可以转换为 STL 格式，而 STL 格式的程序并不一定可以转换为 LAD/FBD。

3.4.7. LAD/FBD/STL 之比较

语言 特性	LAD	FBD	STL
易读性	易	易	差
可移植性	易	易	差
逻辑分析	易	易	难
程序代码量	中	中	小
代码效率	中	中	高
指令记忆	易	易	难
支持功能	中	中	强大

表 2: LAD/FBD/STL 之比较

4. 工程工具

随着现代工控技术的不断发展，PLC 日益面对着越来越复杂的控制任务。如果工程技术人员在开发这些复杂的控制程序的时候，仍然使用传统的标准编程工具，将难于完成控制任务。为了解决这一问题，西门子推出了很多工程工具，使得工程师能够高质高量地完成控制任务程序编写。

4.1. 重要提示:

- ◇ 使用西门子工程工具的前提是对西门子标准工具Step7 的熟练掌握，如果对标准编程工具 STEP7 并不熟悉，**请忽略本章节。**
- ◇ 工程工具并不包括在STEP7标准版中，都需要单独安装
- ◇ 不同的工程工具对 PLC 硬件性能的要求有所不同，但都有不同程度的增加。当使用工程工具时，建议最低使用 CPU314 以上产品。

4.2. S7-SCL

4.2.1. S7-SCL简介

S7-SCL (Structured Control Language 结构化控制语言) 具有以下特点:

- ◇ 是一种类似于PASCAL的高级编程语言，
- ◇ 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-SCL 为 PLC 做了优化处理, 它不仅仅具有 PLC 典型的元素 (例如 输入 / 输出, 定时器, 计数器, 符号表), 而且具有高级语言的特性, 例如:

- ◇ 循环
- ◇ 选择
- ◇ 分支
- ◇ 数组
- ◇ 高级函数

S7-SCL 其非常适合于如下任务:

- ◇ 复杂运算功能
- ◇ 复杂数学函数
- ◇ 数据管理
- ◇ 过程优化

4.2.2. SCL与STL:

SCL 可以编译成 STL, 虽然其代码量相对于 STL 编程有所增加, 但我们更关心的是程序结构和程序的总体效率。类似于计算机行业的发展, 汇编语言已经被舍弃, 取而代之的是 C/C++ 等高级语言。SCL 对工程设计人员要求较高, 需要其具有一定的计算机高级语言的知识 and 编程技巧。

4.2.3. SCL的安装与使用:

STEP7 标准版并不包括 SCL 软件包及授权, 需单独购买, STEP7 Professional 版包括了 SCL 的软件包及授权, 安装即可。在 S7 程序中, S7-SCL 块可以与其它 STEP7 编程语言生成的块互相调用。SCL 生成的块也可以作为库文件被其它语言引用。由于 S7-SCL 程序由 ASCII 文本构成, 所以它非常容易被导入或导出。

4.2.4. SCL应用于虚拟工程

下面将以第 2 章中的虚拟工程中的工艺参数环节来举例说明 SCL 的使用。

工艺参数环节要求:

- ◇ 采集某个过程量, 进行工程量转换, 对其进行软件滤波, 计算10个采样值去除最大值及最小值之后的平均值。(见下面举例)
- ◇ 将过程参数存储, 并进行分析, 优化控制策略 (限于篇幅, 不做介绍)

4.2.5. SCL简单示例

在下面的例子中, 编写一个完成软件滤波程序的 FB1, 程序每调用一次 FB1, 其采集一个新的

过程变量，存储在 FIFO 堆栈中，共 10 个周期的采样值，超过 10 个周期的采样值将被舍弃。程序将此 10 个采样值中的最大最小值找出，并计算出除去最大值及最小值之后的平均值。

平均值（滤波输出）=（10 个采样值之和-最大值-最小值）/8

平均值（非滤波输出）=转换后的实时采样值

1) 新建项目，插入 SCL Source

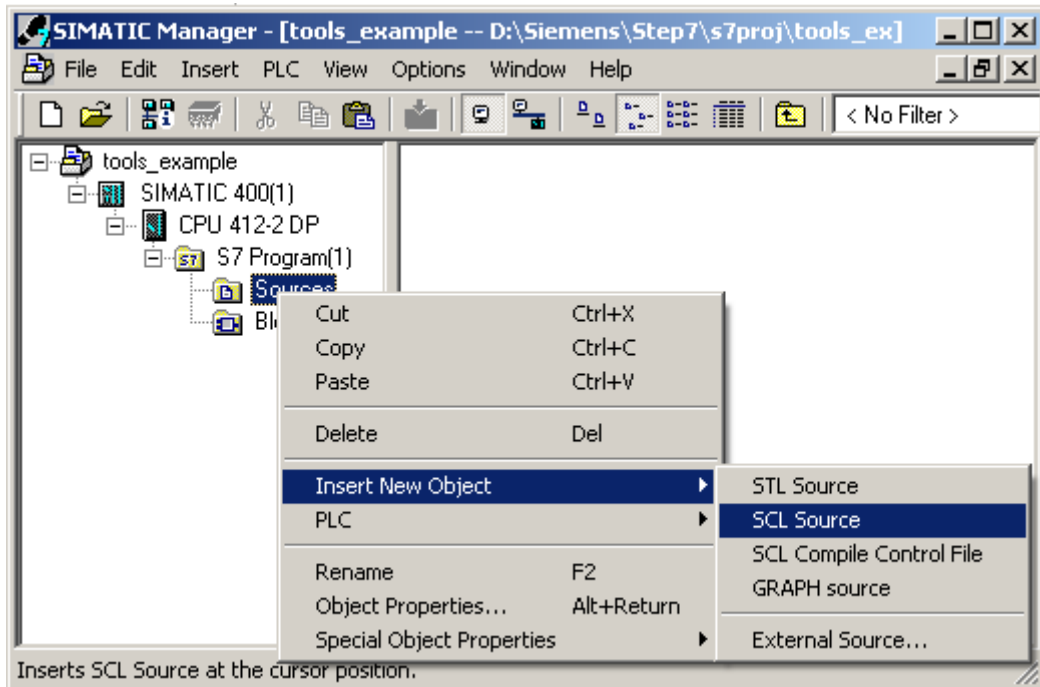


图 6: 新建 SCL Source

2) 双击，SCL Source 打开 SCL 环境，并使用 FB 模板

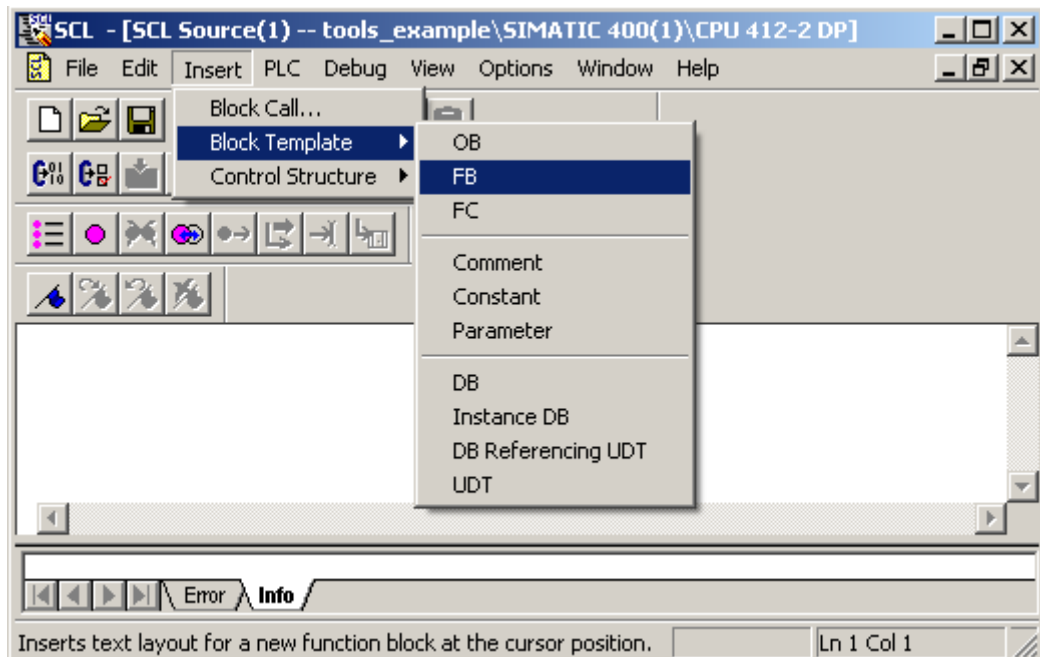


图 7: 使用 FB 模板

3) 更改 FB 编号

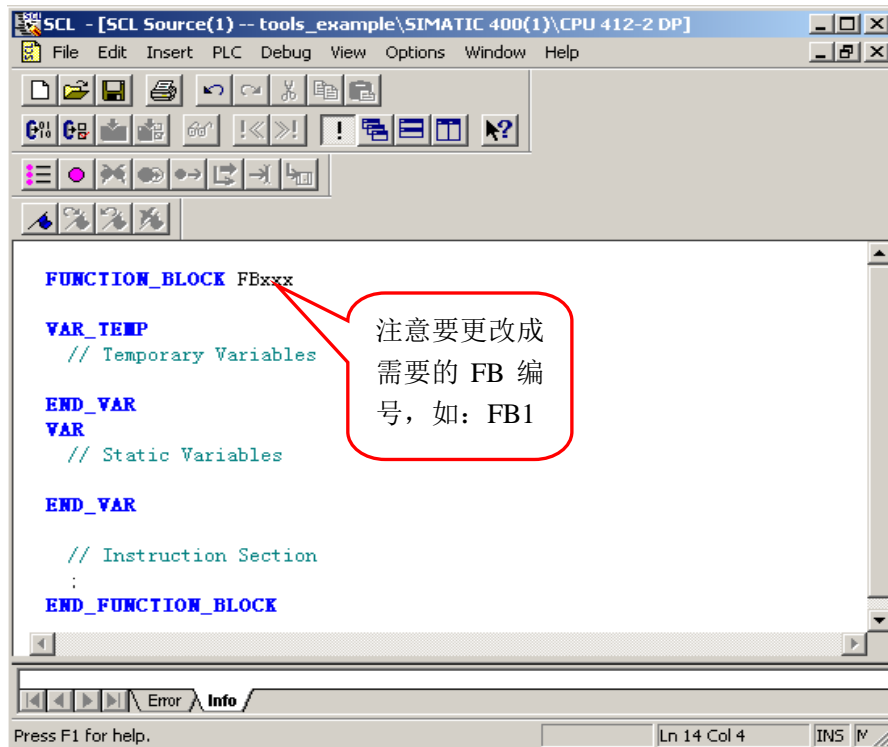


图 8: 更改 FB 编号

4) 使用参数模板

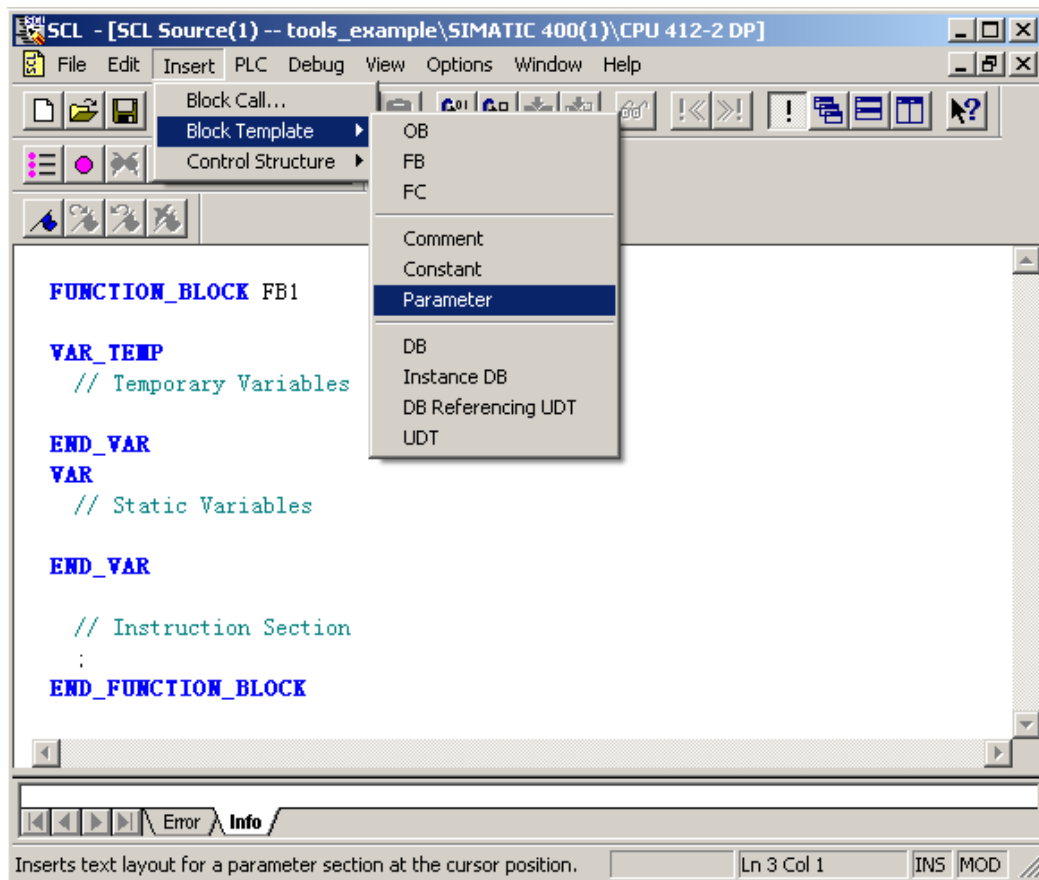


图 9: 添加 FB 输入输出参数

5) 编辑 FB 参数

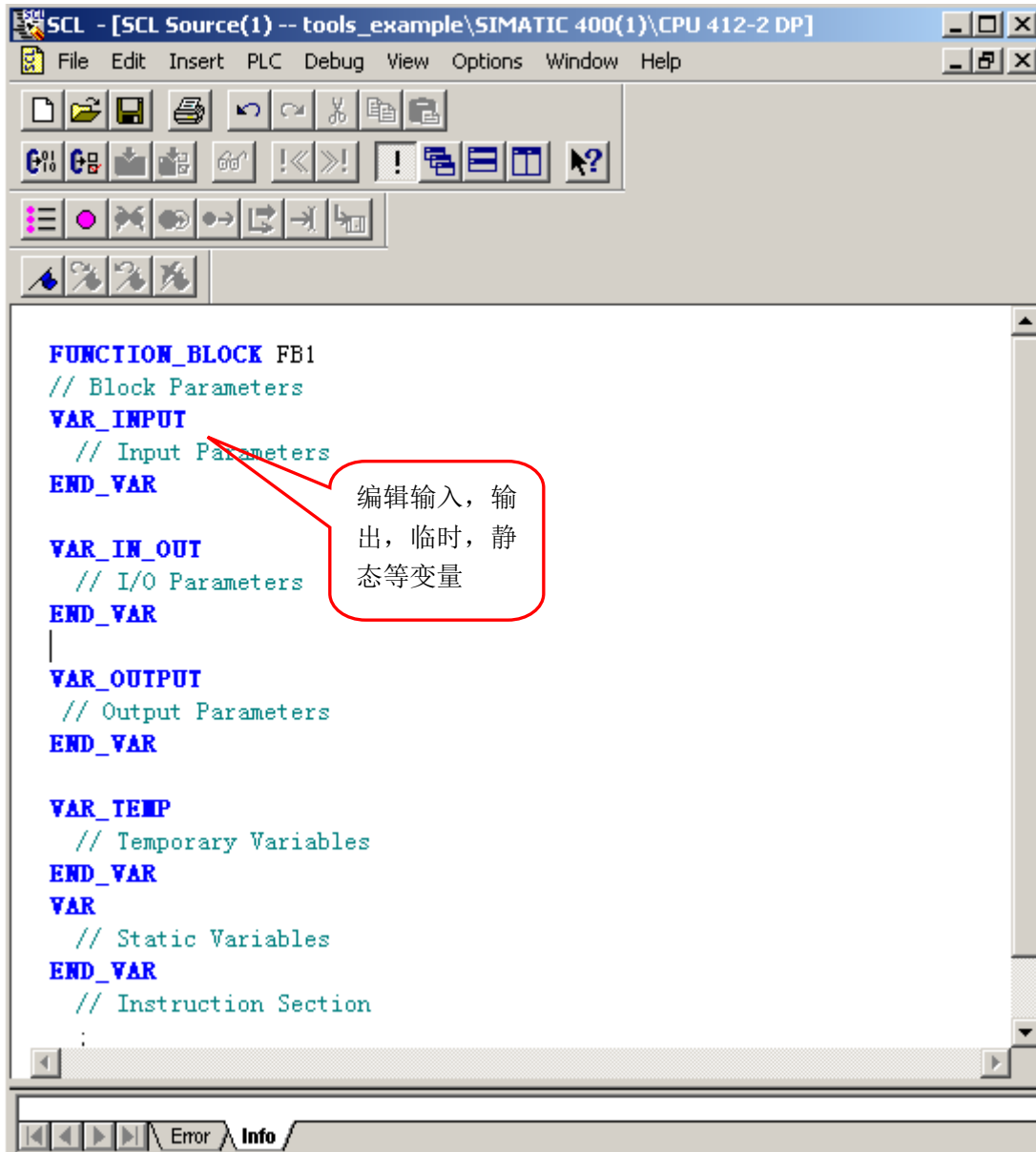


图 10: 添加 FB 输入输出参数

6) 使用调用功能块向导

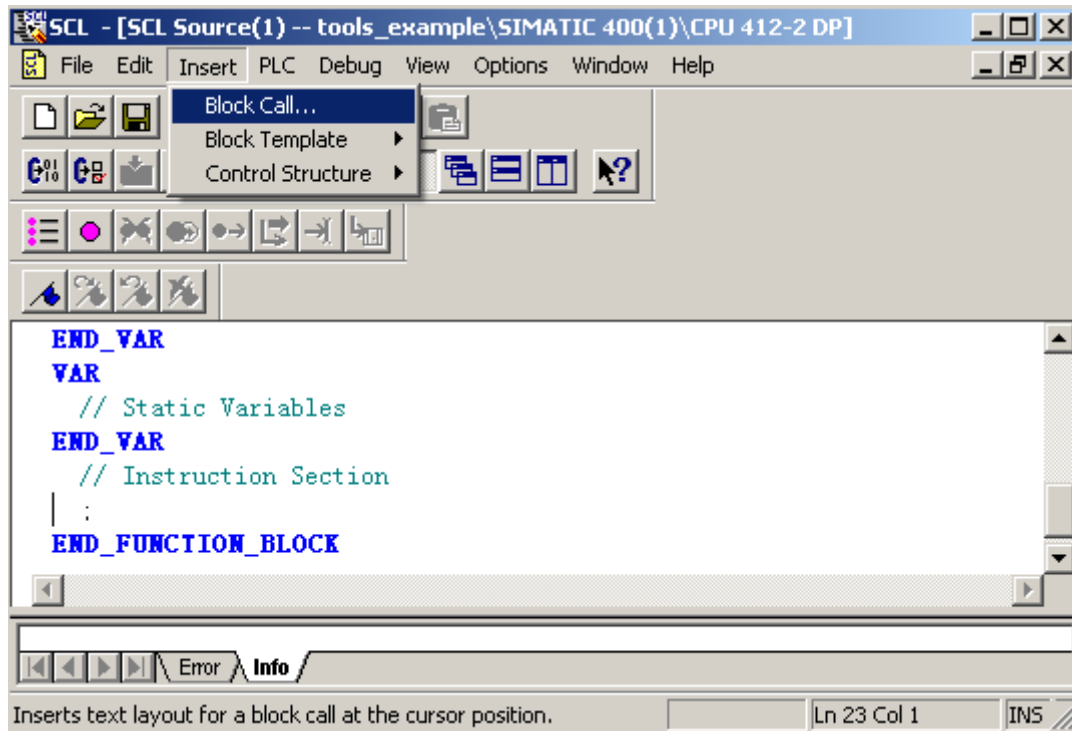


图 11: 使用调用功能块向导

7) 调用 FC105

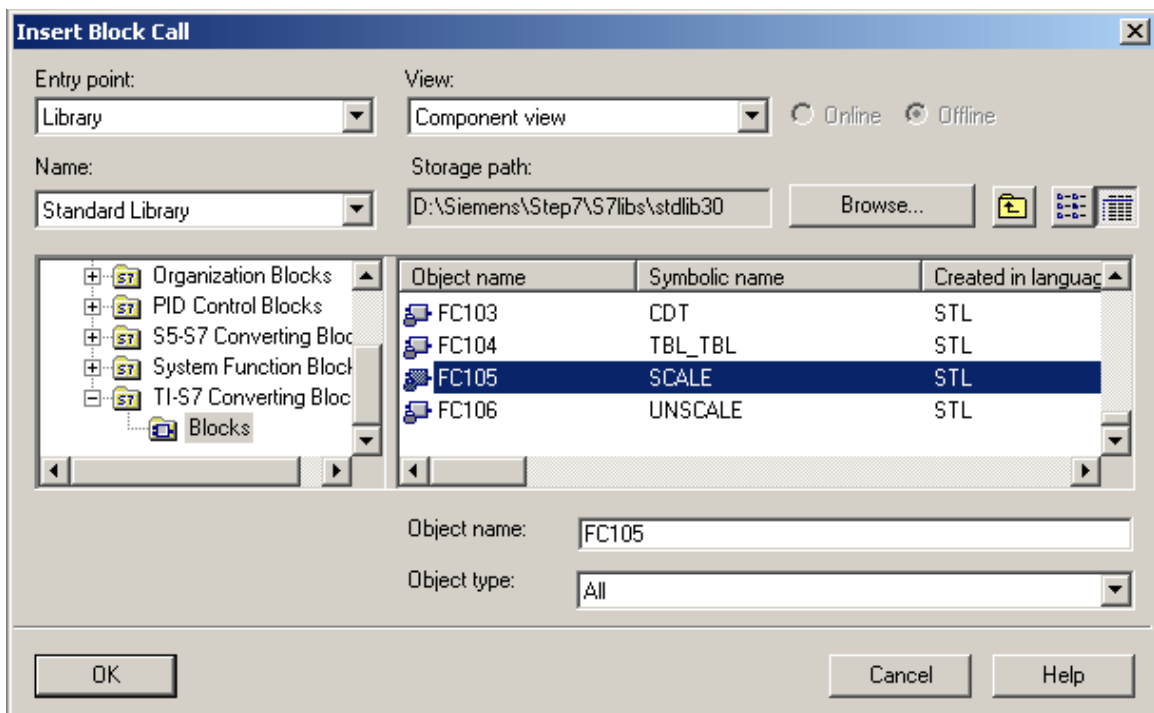


图 12: 调用 FC105

- 8) 由于在 SCL 中调用了 FC105，还需在 SIMATIC Manager 将 FC105 添加到项目中

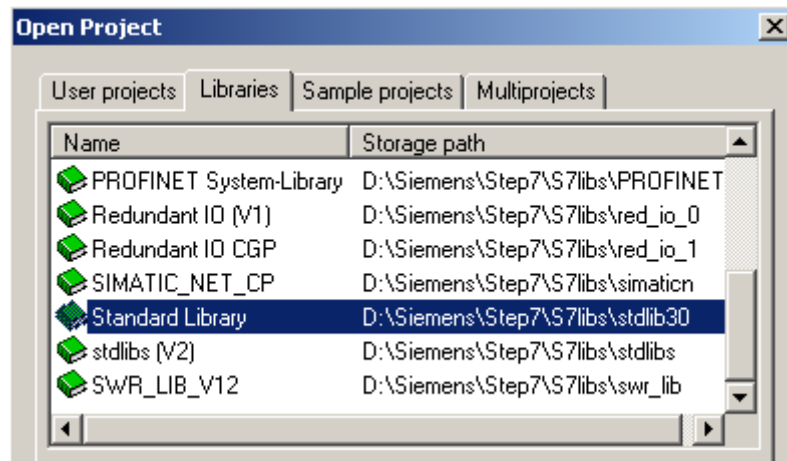


图 13: SIMATIC Manager 中打开库文件

- 9) FC105 复制到当前项目中

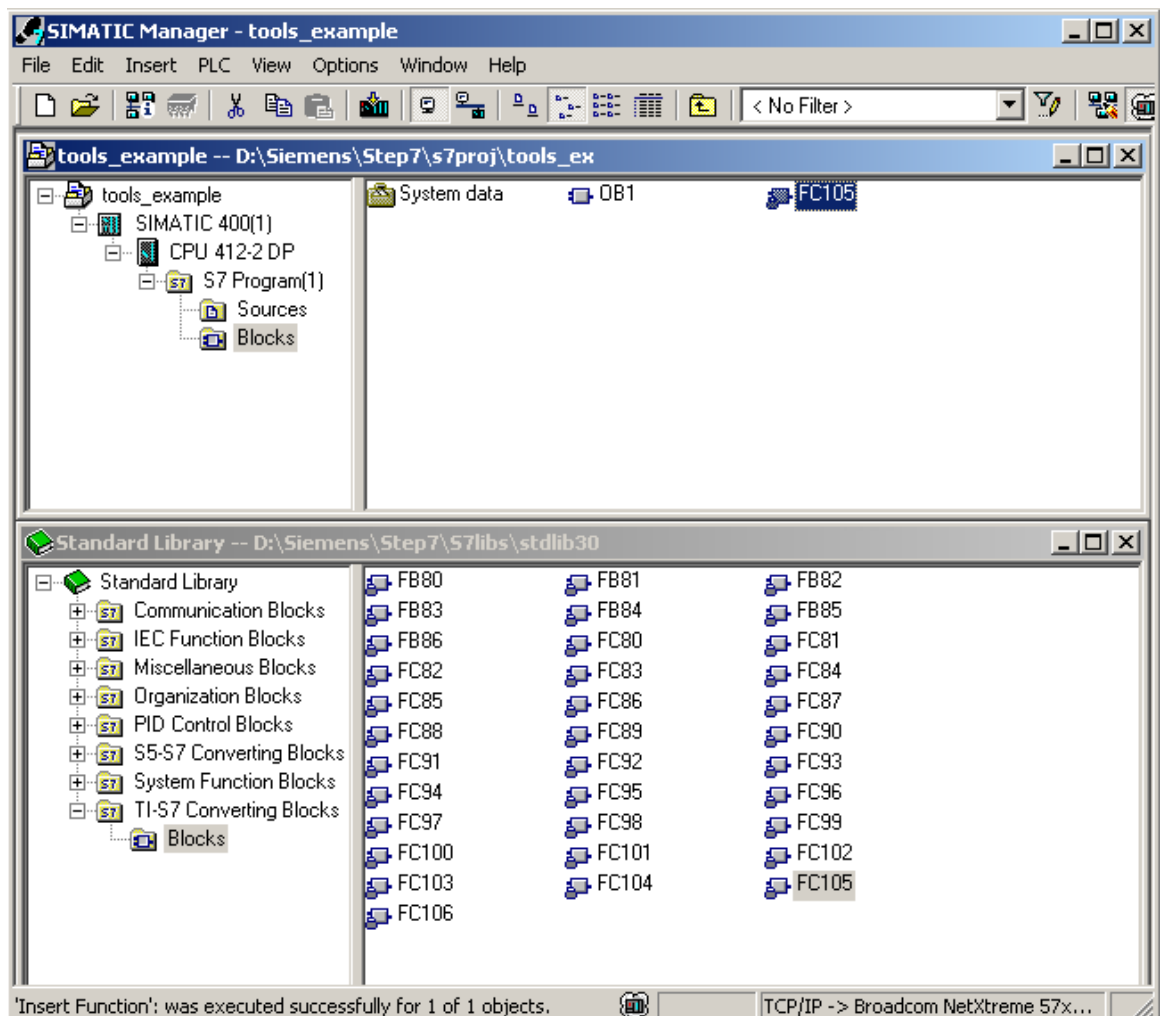


图 14: 将 FC105 复制到当前项目中

10) FB1 参数定义

```

FUNCTION_BLOCK FB1
VAR_INPUT                                     //定义输入参数
    PIW_IN:      INT;
    HI_LIM_IN:   REAL :=100.0;
    LO_LIM_IN:   REAL :=0.0;
    BIPOLAR_IN:  BOOL :=FALSE;
END_VAR

VAR_IN_OUT
END_VAR

VAR_OUTPUT                                     //定义输出参数
    SCALED_VAL:  REAL;
    SCALED_FILTERD: REAL;
    ERR:  BOOL;
END_VAR

VAR_TEMP                                       //定义临时变量
    RET_VAL_105: WORD;
    LOOP_COUNT:  INT;
    MAX_DATA:    REAL;
    MIN_DATA:    REAL;
    TOTAL:       REAL;
    TOTAL_FILTERD: REAL;
END_VAR

VAR                                             //定义静态变量
    DATA_STORE: ARRAY[0..9] OF REAL;
END_VAR
    
```

图 15: FB1 参数定义

11) FC105 调用

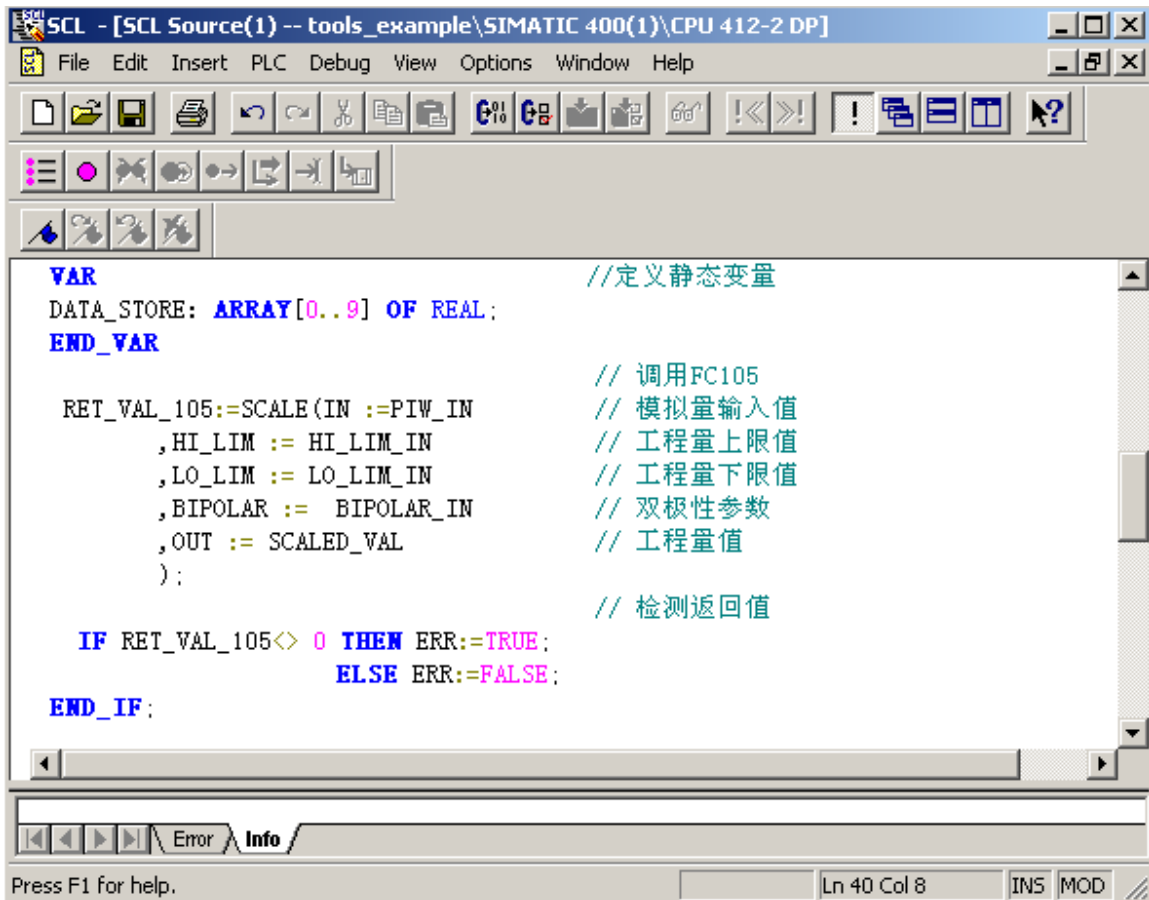


图 16: FC105 调用

12) 下面的程序中使用了判断条件，在此使用 IF 模板

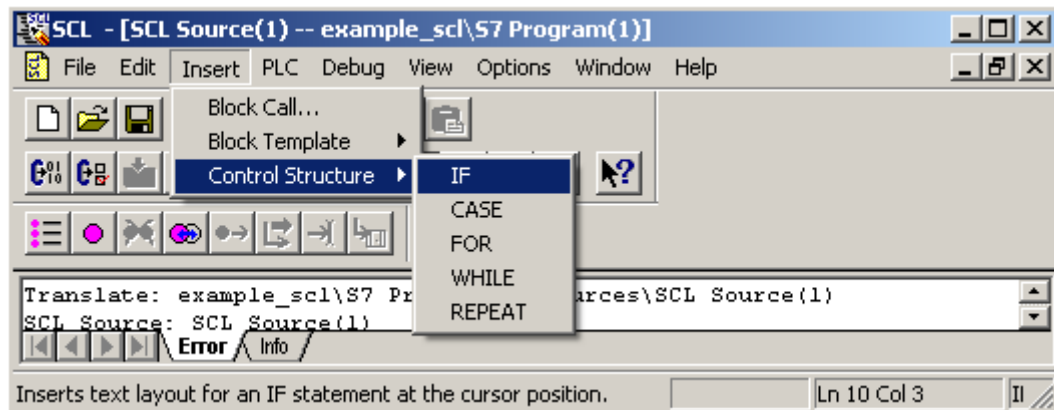


图 17: 使用 IF 模板

13) 使用 IF, FOR-NEXT 等高级语言的方式，很容易就可编写出滤波程序

```

MAX_DATA:=DATA_STORE[0];      // 初始化
MIN_DATA:=DATA_STORE[0];
TOTAL:=0.0;
TOTAL_FILTERD:=0.0;

FOR LOOP_COUNT:= 0 TO 8 BY 1 DO      // FIFO 堆栈
    DATA_STORE[LOOP_COUNT]:=DATA_STORE[LOOP_COUNT+1];
END_FOR;
DATA_STORE[9]:=SCALED_VAL;          // 加入新检测值

FOR LOOP_COUNT:= 0 TO 9 BY 1 DO
    //查找最大值
    IF MAX_DATA < DATA_STORE[LOOP_COUNT] THEN MAX_DATA:=DATA_STORE[LOOP_COUNT];
END_IF;
    //查找最小值
    IF MIN_DATA > DATA_STORE[LOOP_COUNT] THEN MIN_DATA:=DATA_STORE[LOOP_COUNT];
END_IF;
    //计算累加和
    TOTAL:=TOTAL+DATA_STORE[LOOP_COUNT];
END_FOR;
    //计算滤波后累加和
    TOTAL_FILTERD:=TOTAL- MIN_DATA- MAX_DATA;
    //计算滤波后平均值
    SCALED_FILTERD:=TOTAL_FILTERD/8.0;

END_FUNCTION_BLOCK
    
```

图 18: 编写滤波程序

14) 在菜单 Option-Customize 中选择生成调试信息

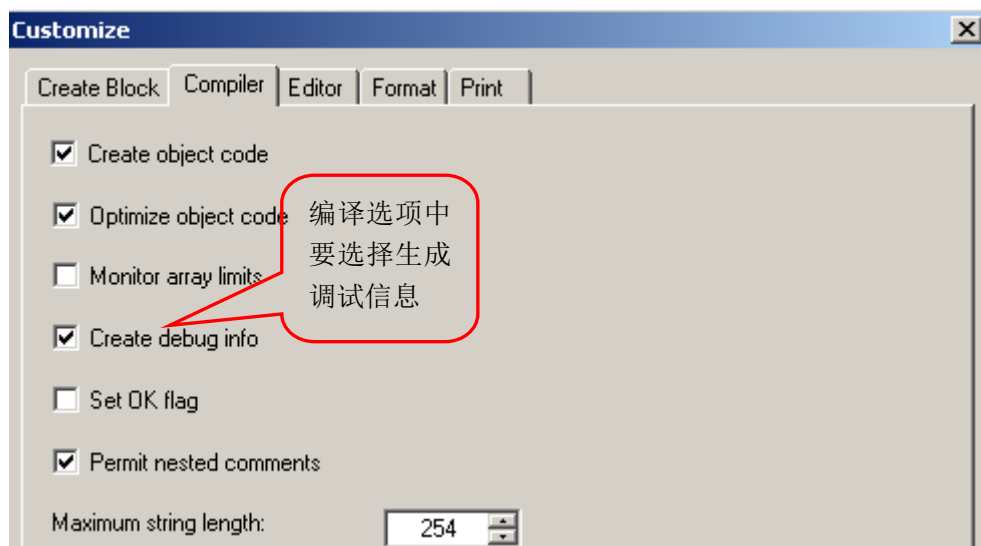


图 19: 生成调试信息

16) 可以在高级语言界面下监控 FB1

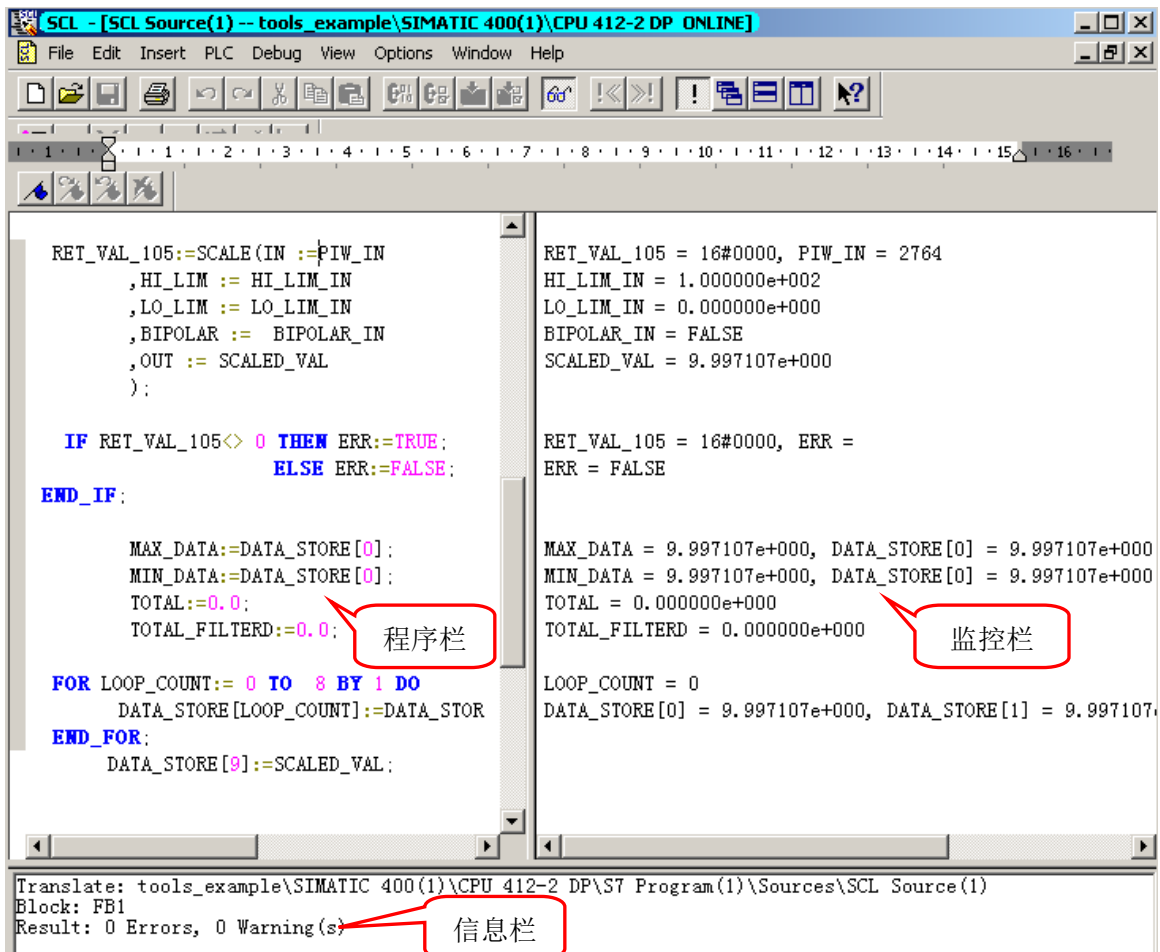


图 21: 监控 FB1

至此，一个简单的 SCL 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 SCL 的具体使用，请按照本文提供的地址连接下载 SCL 手册。

4.2.6. 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。

4.3. S7-GRAPH

4.3.1. 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）
- ◇ 转换条件（每个顺控器最多250个）
- ◇ 分支条件（每个顺控器最多250个）
- ◇ 逻辑互锁（最多32个条件）
- ◇ 监控条件（最多32个条件）
- ◇ 事件触发功能
- ◇ 切换运行模式：手动、自动及点动模式

4.3.2. S7-GRAPH与STL:

S7-GRAPH 可以编译成 STL，虽然其代码量相对于 STL 编程有所增加，但开发人员可以将更多的精力集中到控制任务上，提高了工程效率。

4.3.3. S7-GRAPH安装与使用:

STEP7 标准版并不包括 S7-GRAPH 软件包及授权，需单独购买，STEP7 Professional 版包括了 S7-GRAPH 的软件包及授权，安装即可。在 S7 程序中，S7-GRAPH 块可以与其它 STEP7 编程语言生成的块组合互相调用，S7-GRAPH 生成的块也可以作为库文件被其它语言引用。

4.3.4. S7-GRAPH应用于虚拟工程

下面将以第 2 章中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:

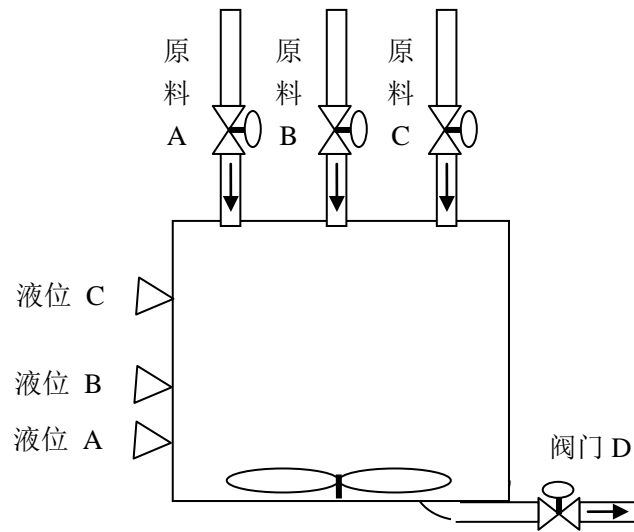


图 22: 原料配比示意图

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：

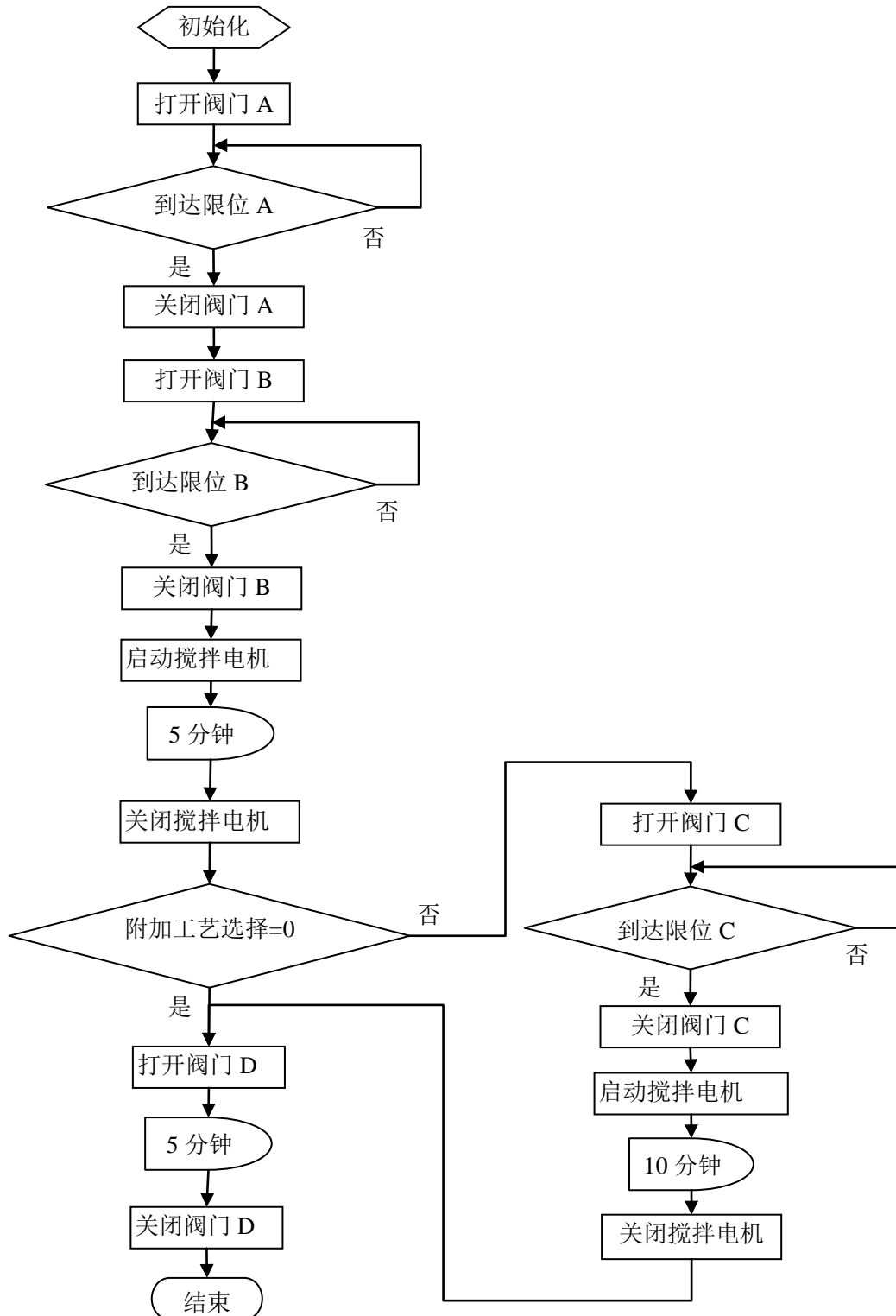


图 23：原料配比工艺流程图

4.3.5. 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

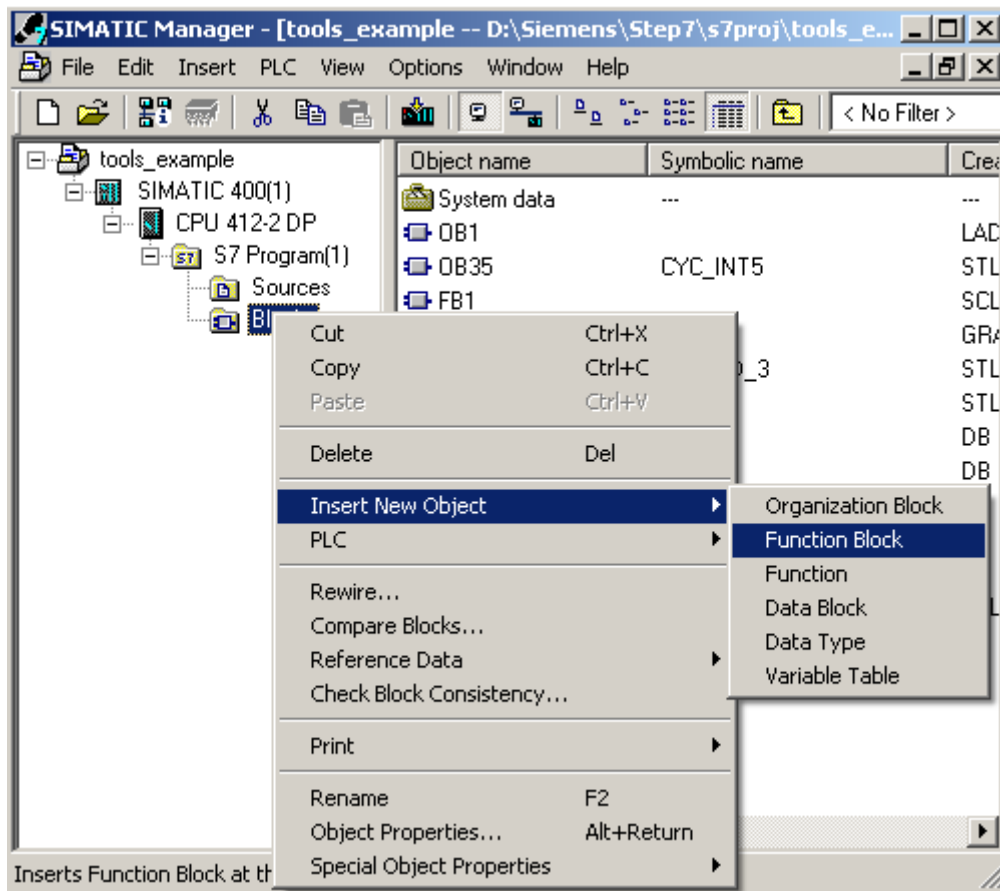


图 24: 添加功能块

2) 选择功能块类型

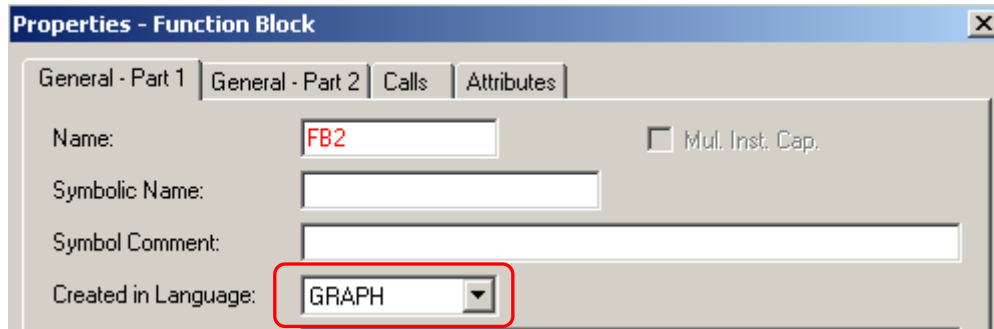


图 25: 选择功能块类型

3) 编程元素

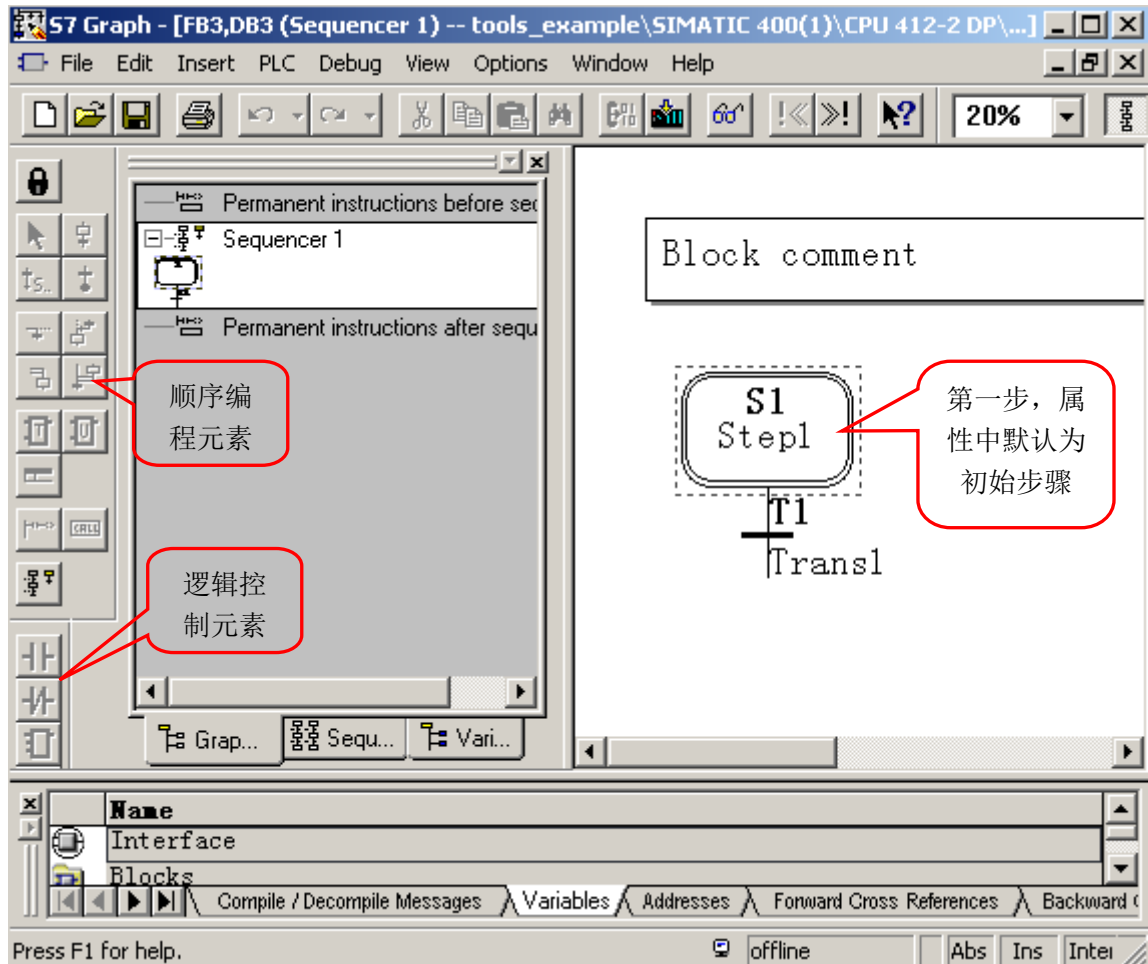


图 26: 编程元素

4) 编程分支及流程

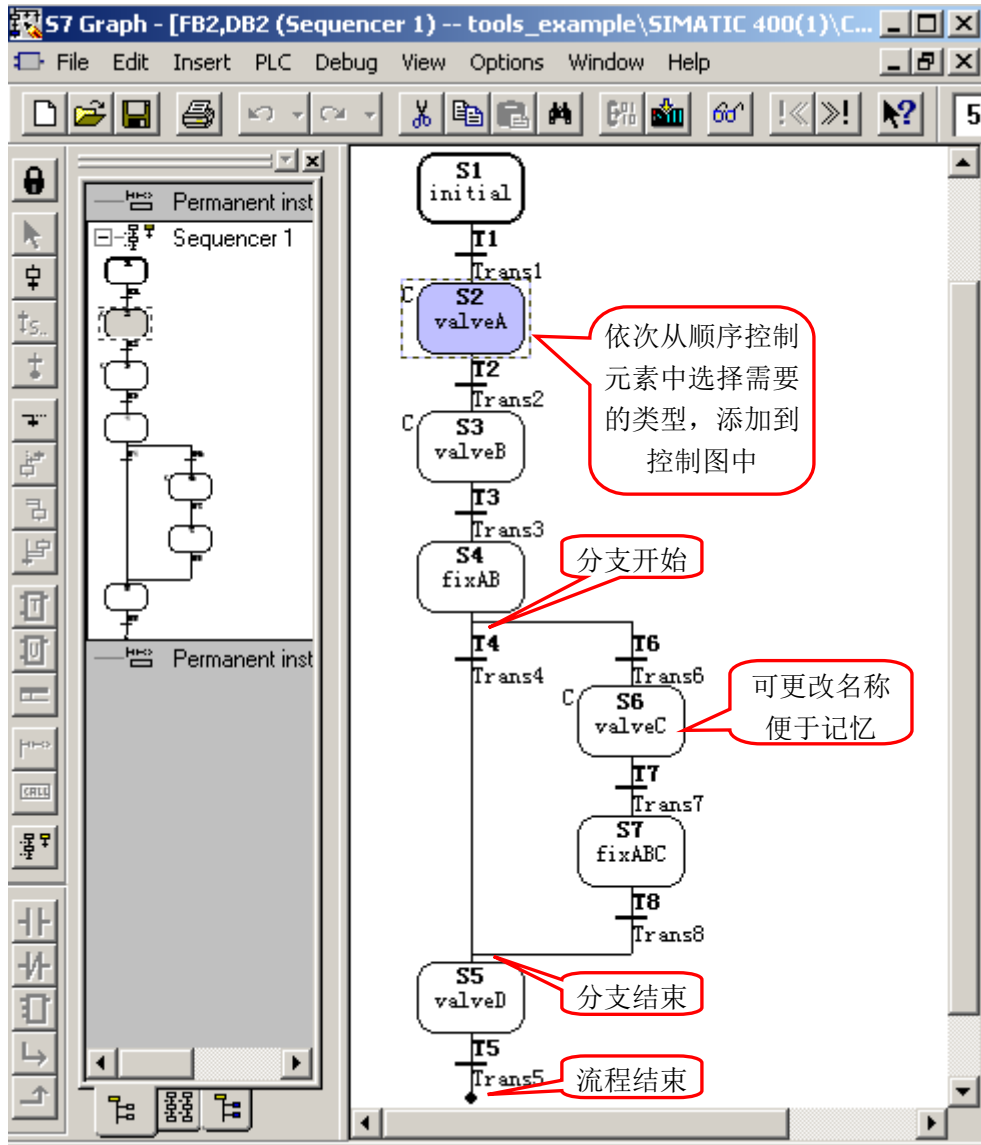


图 27: 编程分支及流程

5) 初始化处理

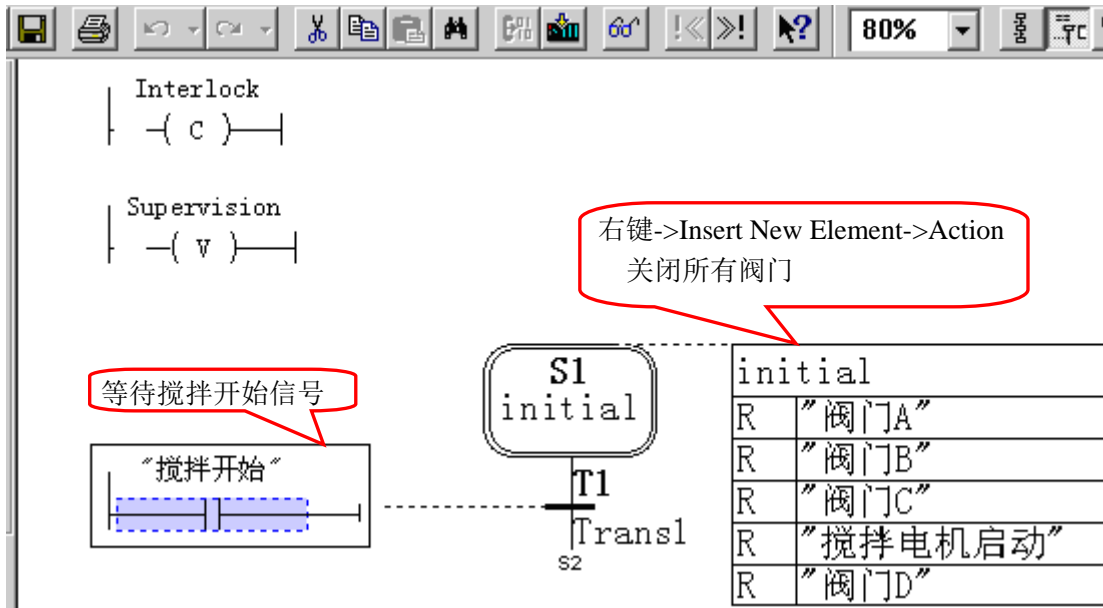


图 28: 初始化

6) 互锁条件及指令

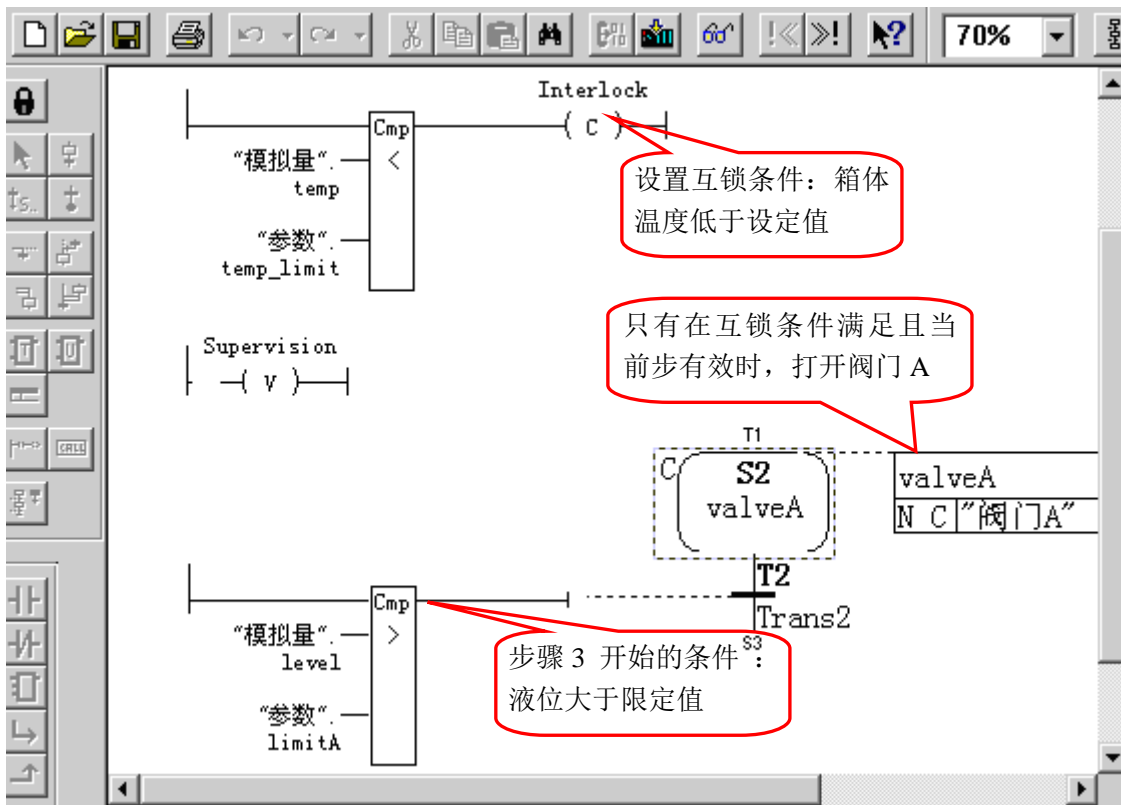


图 29: 编程环境视图

7) 流程分支

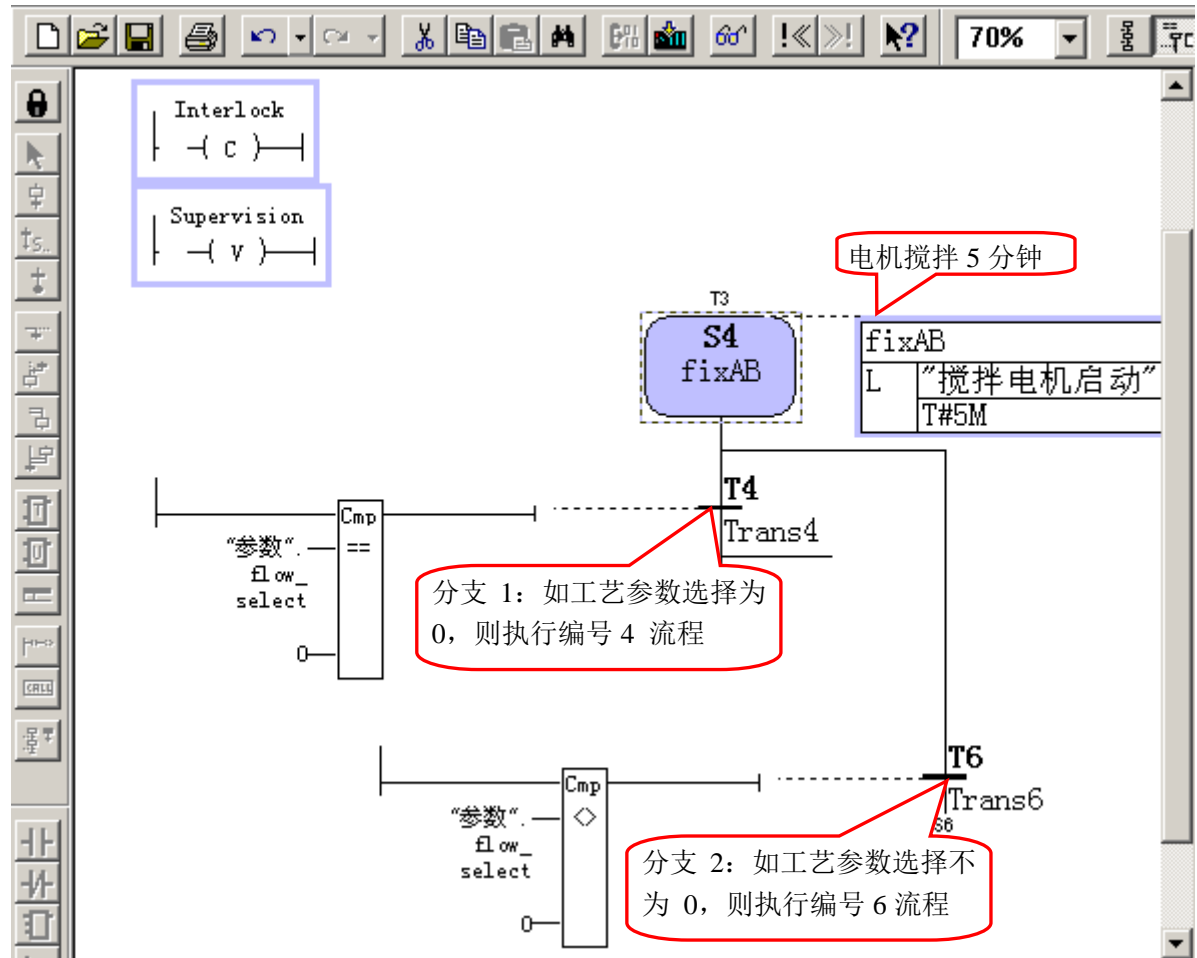


图 30: 流程分支

8) 省略步骤 6-7 具体程序

9) 步骤 5 结束

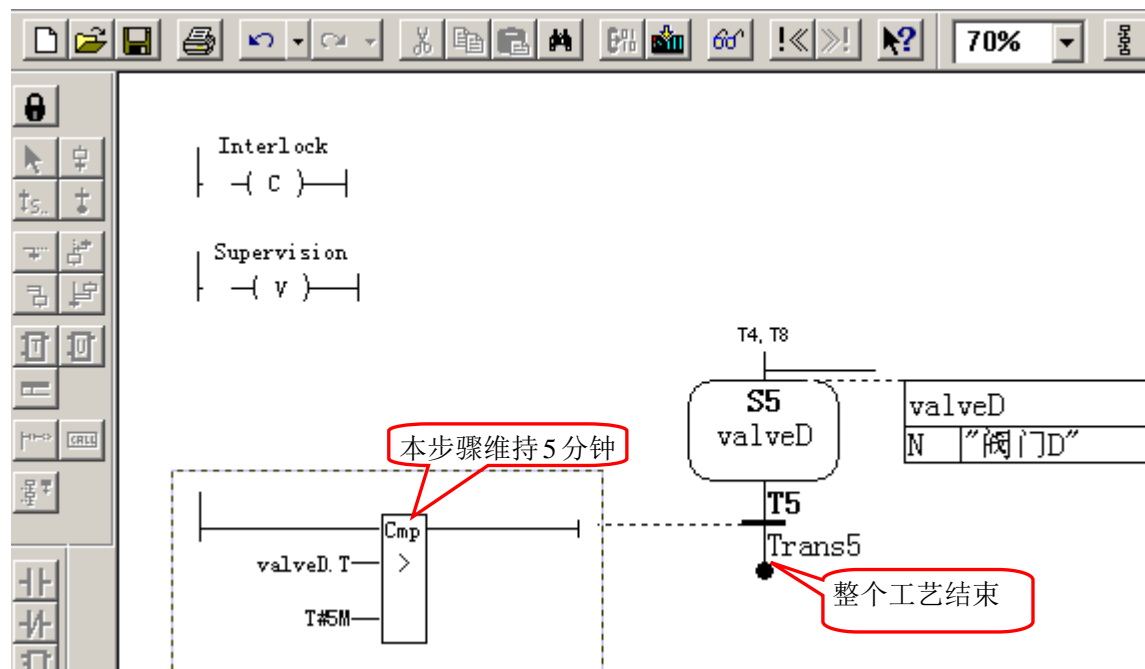


图 31: 流程结束

- 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

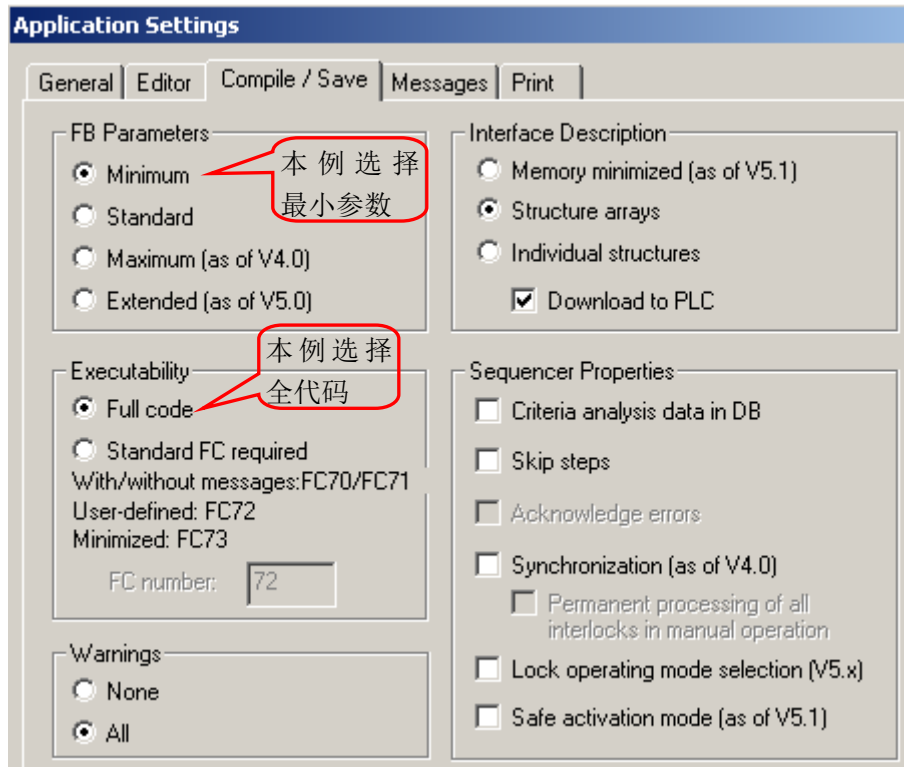


图 32: 参数设置

- 11) 存盘时程序会自动编译，在 OB1 中调用 FB2

OB1 : "Main Program Sweep (Cycle)"

Network 1: Title:

```
CALL FB 2, DB2
INIT_SQ:=M20.0
```

初始化输入控制

图 33: 调用 FB2

12) 下载后，监控 FB2

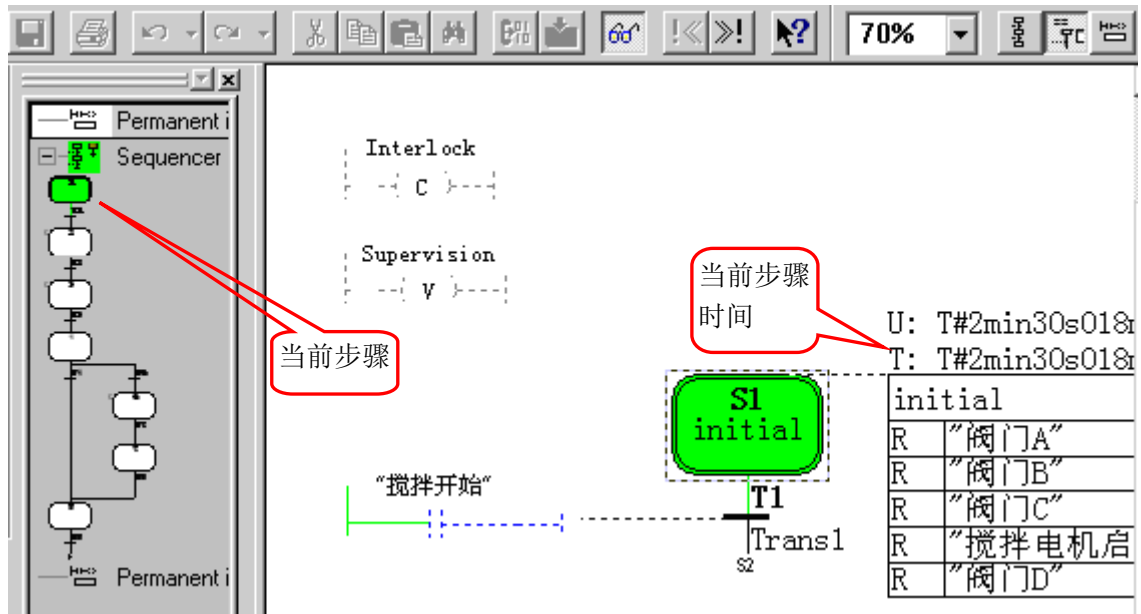


图 34: 监控 FB2

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

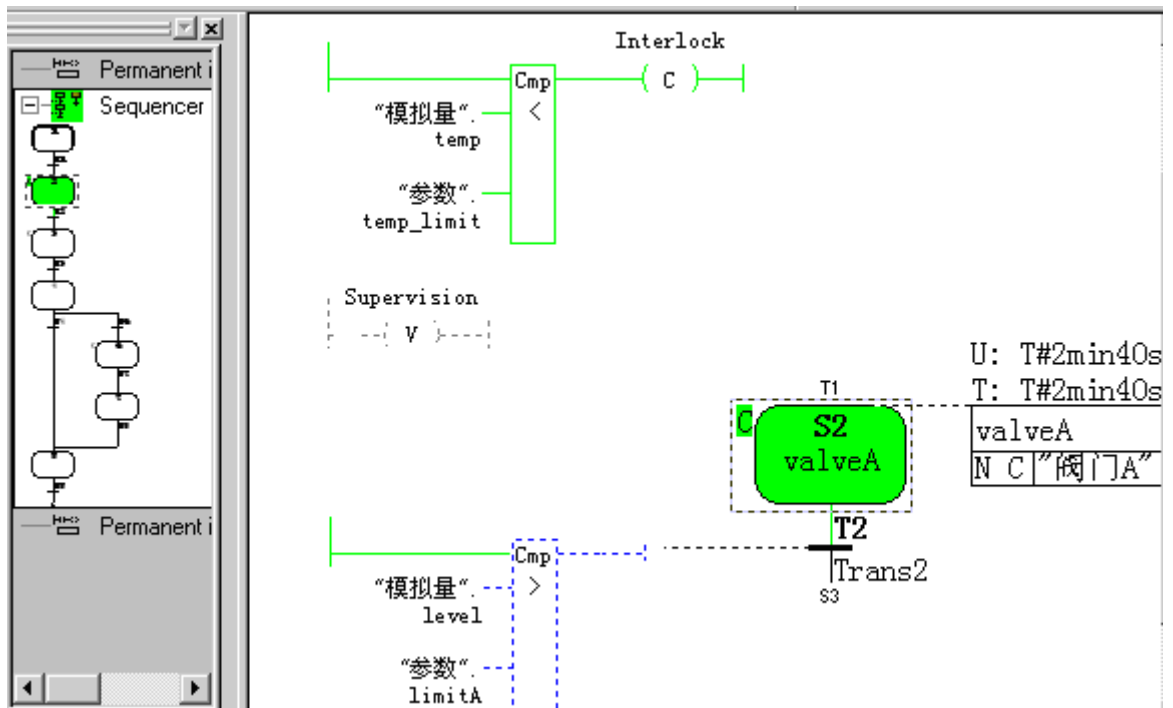


图 35: 测试 FB2

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

Address / ...	Symbol	Type	Comment	Status value
Q1.1	"阀门B"	BOOL		0
DB20.DB4	"参数",limitB	REAL		4.0
DB20.DB14	"参数",temp_limit	REAL		90.0
DB10.DB4	"模拟量",temp	REAL		99.0
DB10.DB0	"模拟量",level	REAL		3.5

图 36: 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 37: 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 GRAPH 不熟悉情况下, 请勿使用此功能。

至此, 一个简单的 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 GRAPH 的具体使用, 请按照本文提供的地址连接下载 GRAPH 手册。

4.3.6. 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。

4.4. S7-HiGraph

4.4.1. S7-HiGraph简介

S7-HiGraph 具有以下特点:

- ◇ 通过绘制功能图表来实现异步控制
- ◇ 非常适合于机械设计工程师, 调试及维护工程师
- ◇ 利于自动化工程师与机械工程师相互沟通
- ◇ 集成了信号监控及触发功能
- ◇ 适用于 SIMATIC S7-300 (推荐用于CPU314以上CPU), S7-400, C7 and WinAC

S7-HiGRAPH 不仅仅具有 PLC 典型的元素(例如 输入/输出, 定时器, 计数器, 符号表), 而且具有图形化编程语言语言的特性, 其非常适合于如下任务:

- ◇ 异步控制
- ◇ 自动机械设计

4.4.2. HiGraph 与STL:

S7-HiGraph 程序最终编译成 STL。其代码量相对于 STL 编程有所增加。

4.4.3. S7-HiGraph的安装与使用:

STEP7 各版本均并不包括 S7-HiGraph 软件包及授权，需单独购买。在 S7 程序中，S7-HiGraph 块可以与其它 STEP7 编程语言生成的块组合互相调用。S7-HiGraph 生成的块也可以作为库文件被其它语言引用。

4.4.4. S7-HiGraph应用于虚拟工程

下面将以第 2 章中的虚拟工程中的产品成型环节来举例说明 S7-HiGraph 的使用，此工艺环节要求：

1-3 号成型机各自生产产品，当某台成型机产品就绪时，机械手将产品取走，并放置在传送带上。

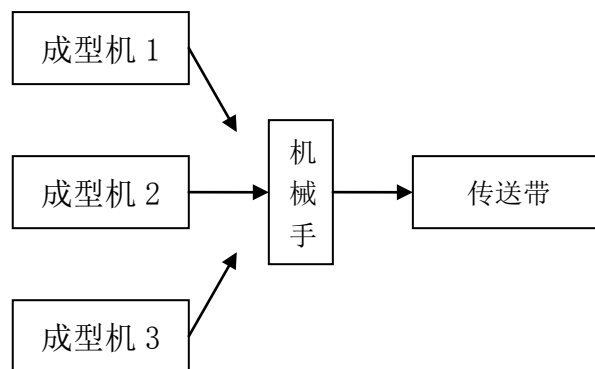


图 38: 产品成型系统构成

成型机工艺流程:

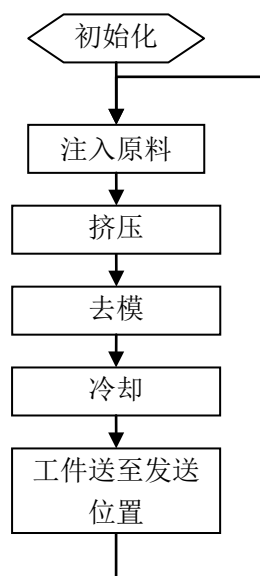


图 39: 成型机工艺流程

机械手工艺流程:

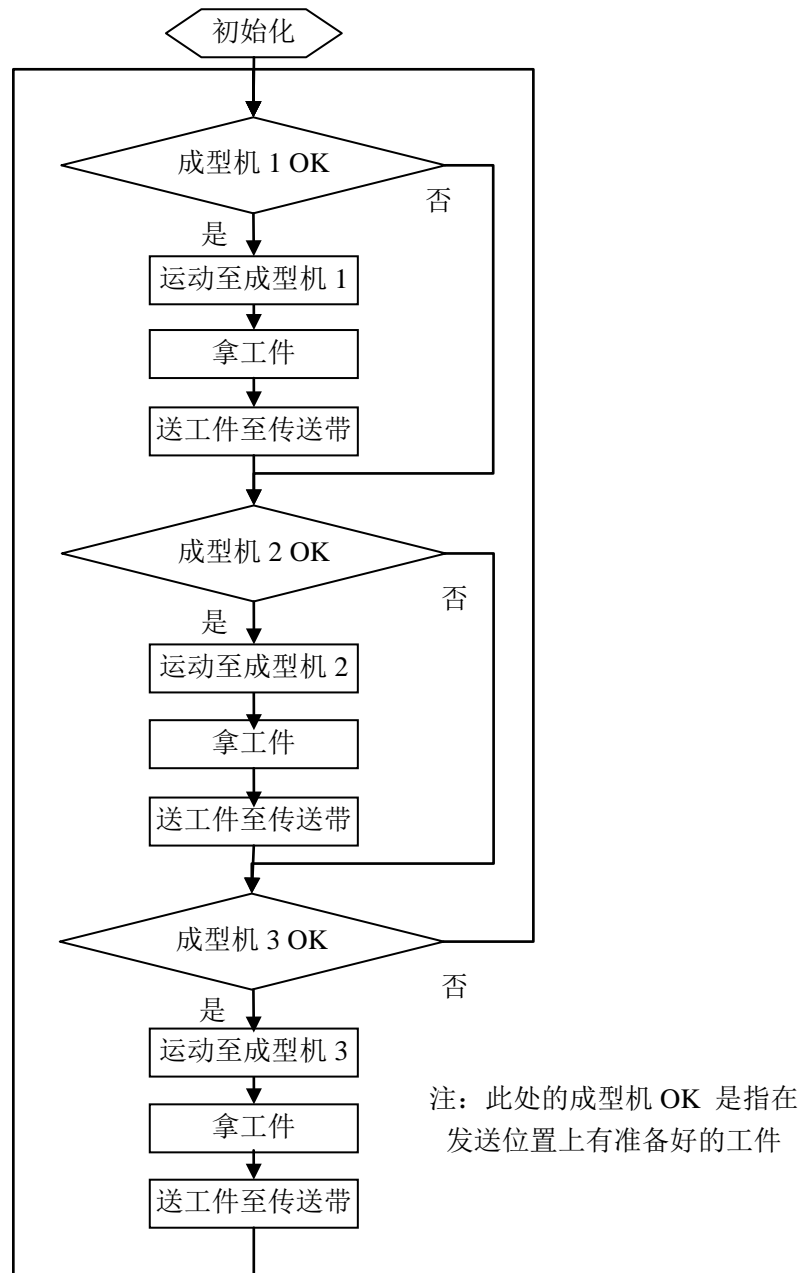


图 40: 机械手工艺流程

4.4.5. S7-HiGRAPH简单示例

在下面的例子中，将编写一个 S7-HiGraph 用于成型工艺控制的程序：

- ◇ 控制每个成型机
- ◇ 控制机械手
- ◇ 协调成型机及机械手工作

- 1) 右键点击 Sources -> Insert New Object-> State graph, 名字为 molding
- 2) 右键点击 Sources -> Insert New Object-> State graph, 名字为 robot
- 3) 右键点击 Sources -> Insert New Object-> Graph group
- 4) 双击 molding 打开, 编辑成型机程序, 定义输入参数

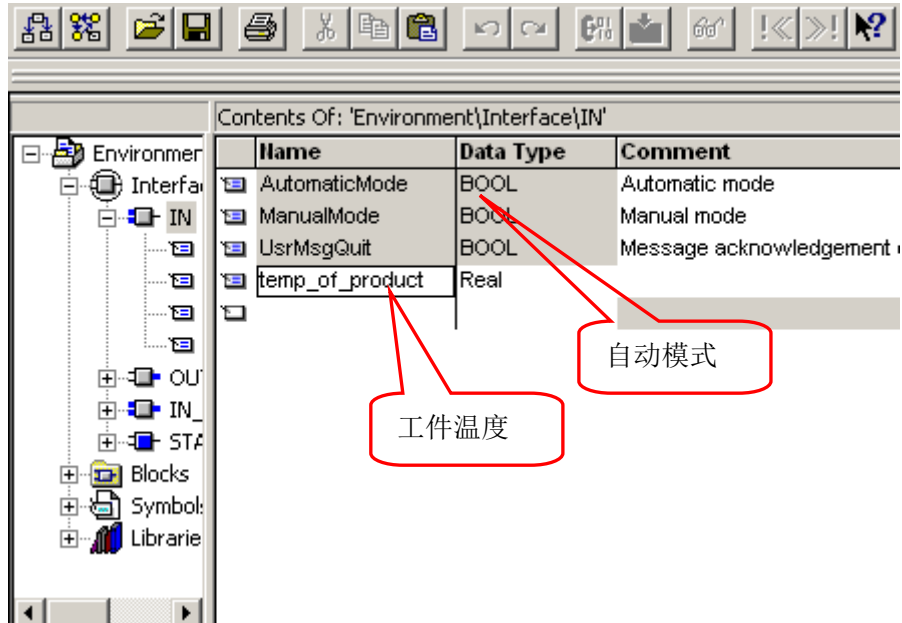


图 41: 成型机输入参数

- 5) 编辑成型机输出参数

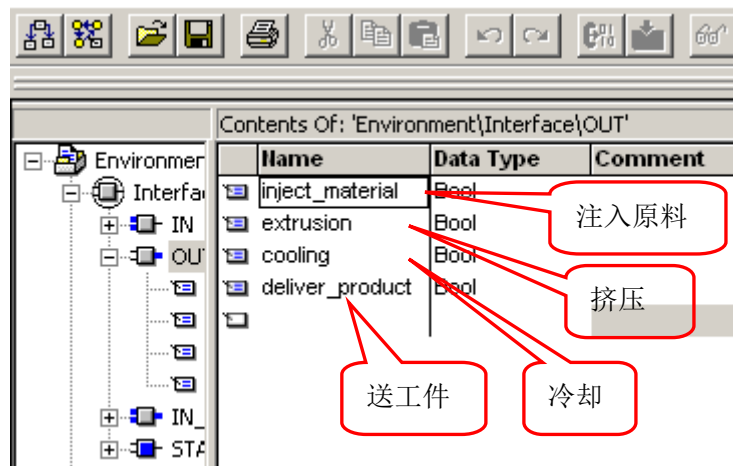


图 42: 成型机输出参数

6) 编辑成型机消息

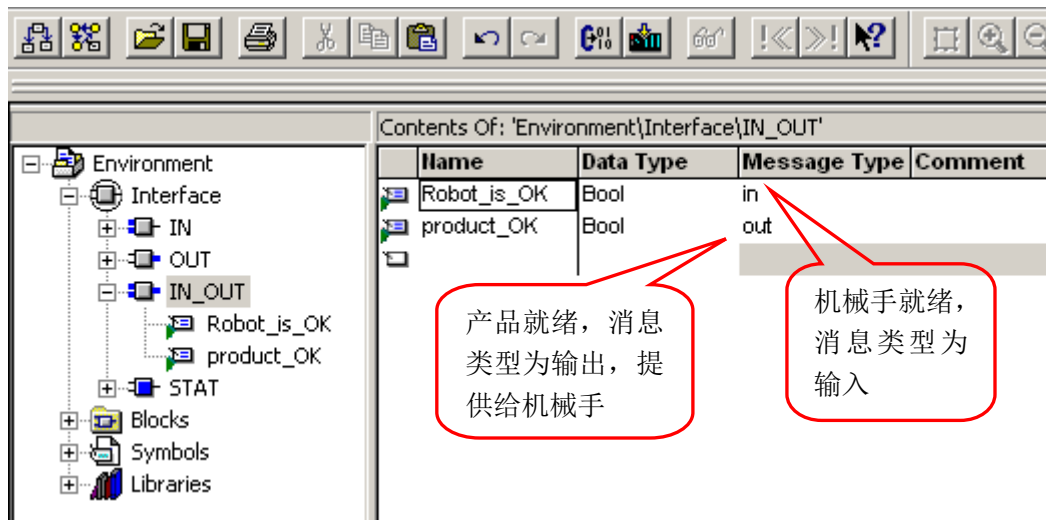


图 43: 编辑成型机消息

7) 编辑成型机静态变量参数

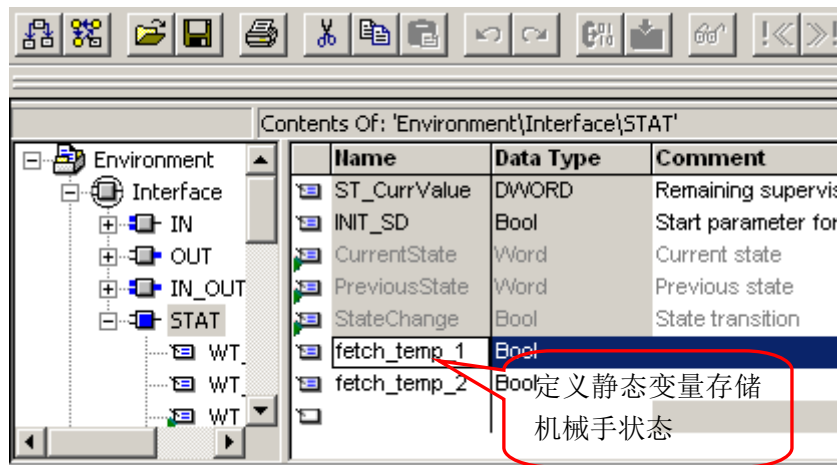


图 44: 成型机静态变量参数

8) 添加状态及条件

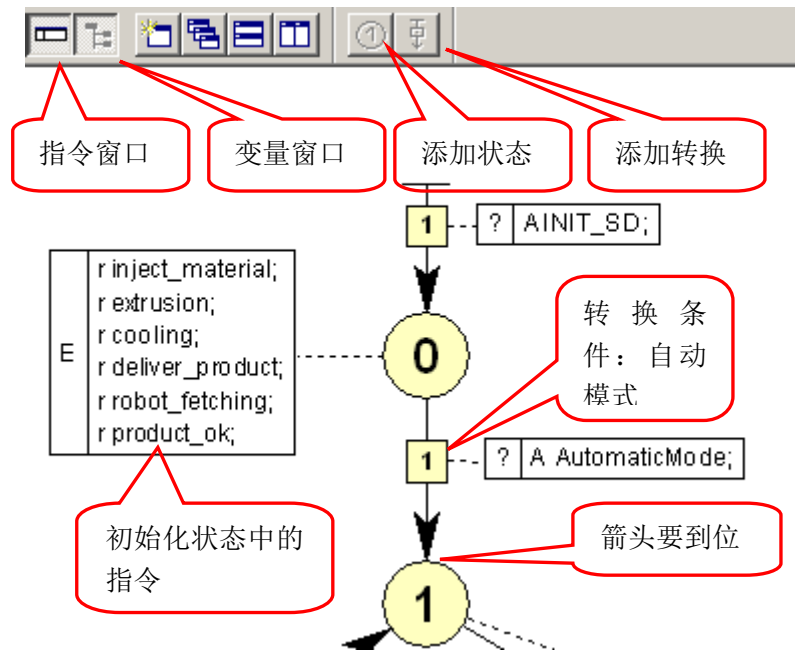


图 45: 添加状态及条件

9) 双击状态, 添加各种指令

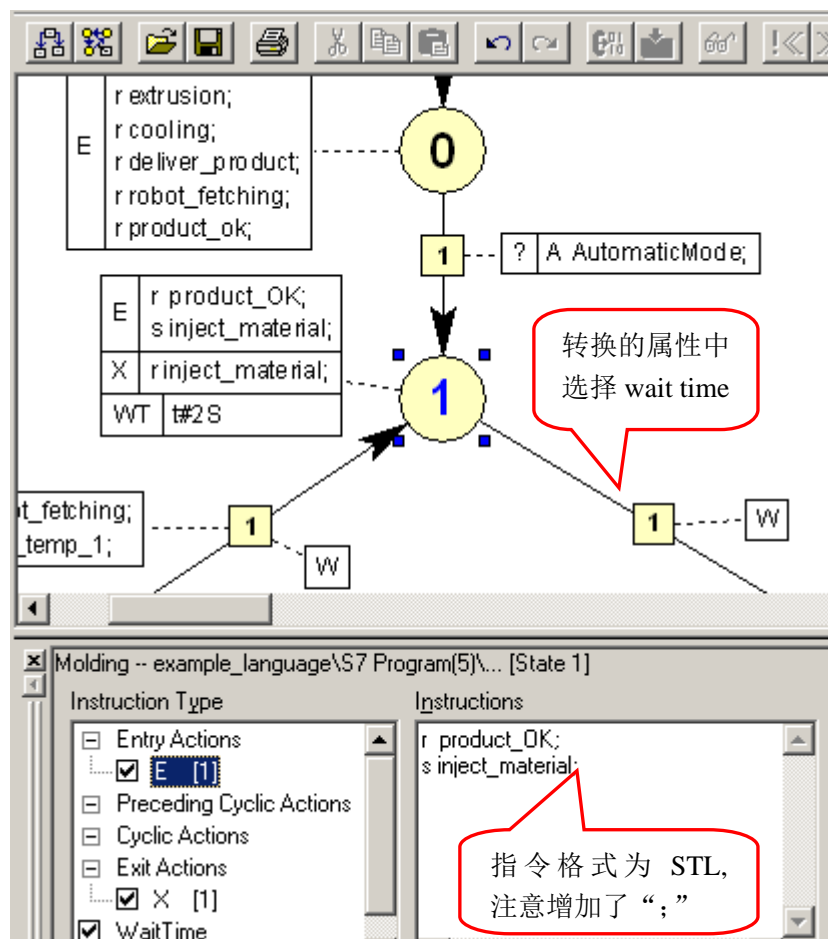


图 46: 添加各种状态下的指令

10) 成型机程序概览

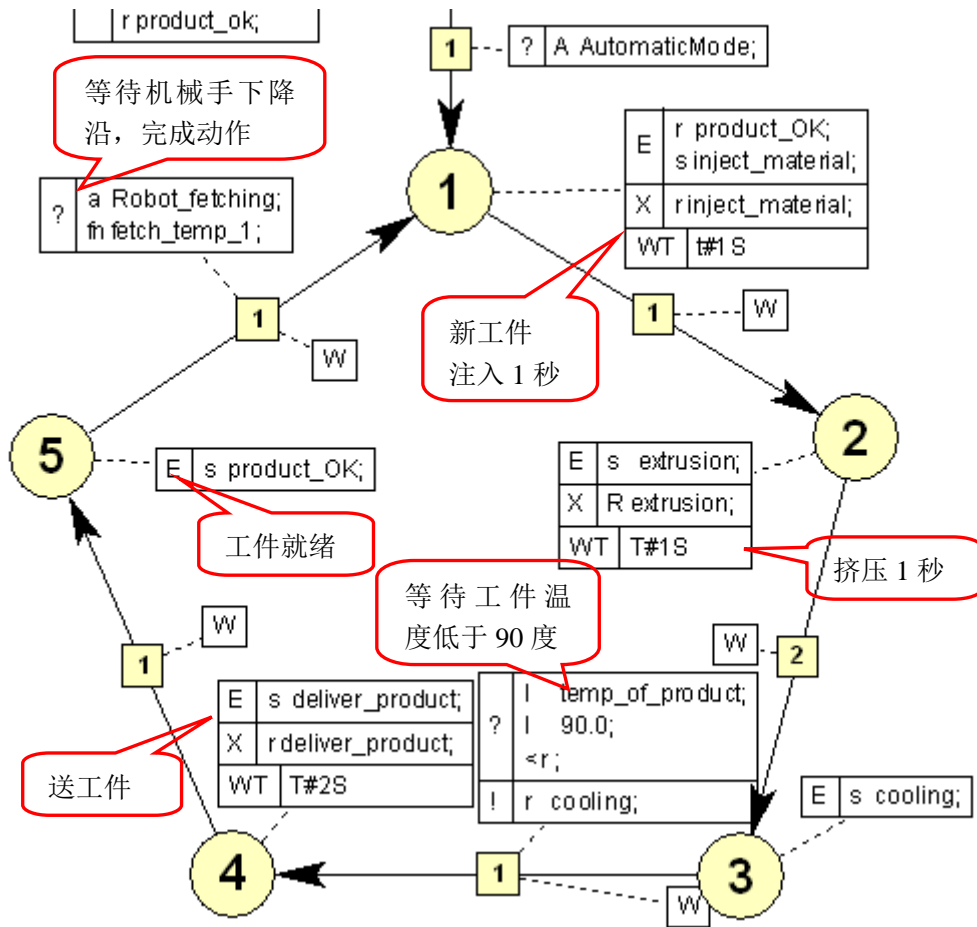


图 47: 成型机程序概览

11) 机械手程序概览

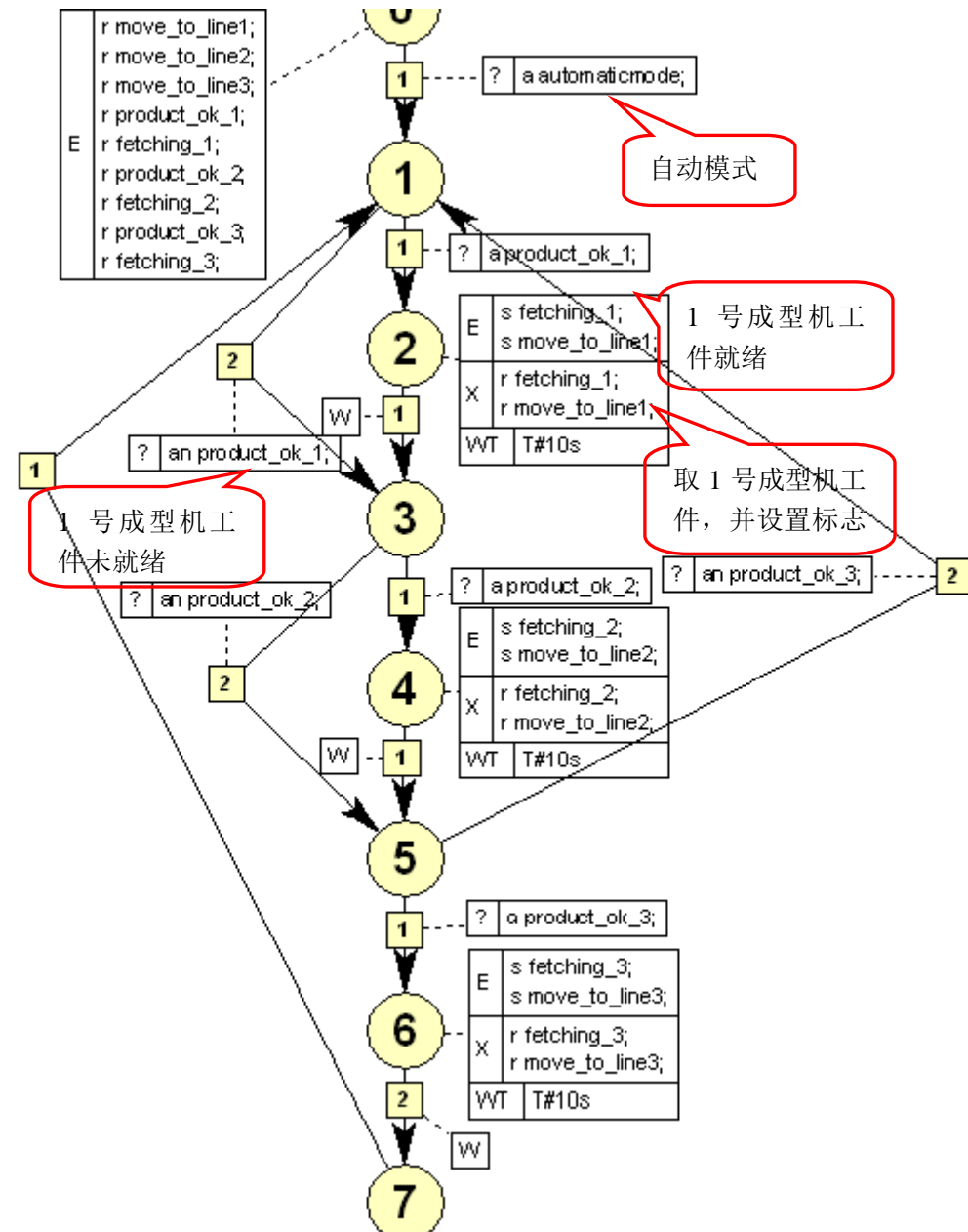


图 48: 机械手程序概览

- 12) 双击 Graph group (1) 打开, 右键 Insert instance, 添加项目中的 molding 和 robot
- 13) 也可以右键设置 Run Sequence
- 14) 将 molding 复制并粘贴 2 次
- 15) 选择 molding, 分配参数

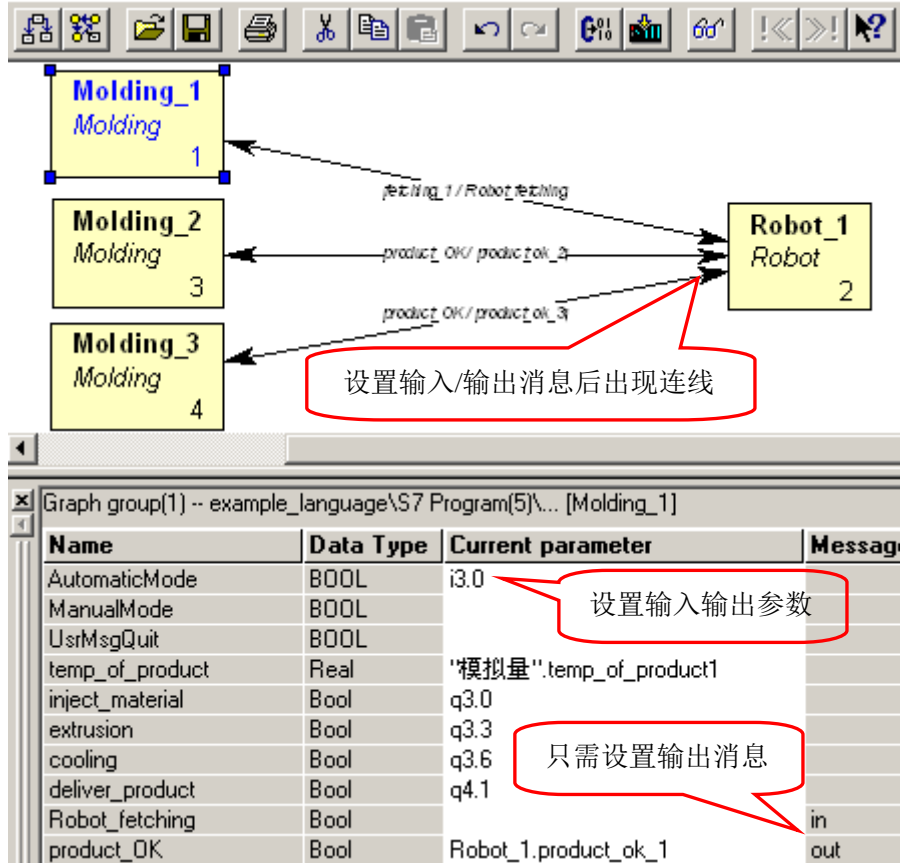


图 49: 分配成型机接口参数

- 16) 依次分配成型机 2, 3 号的接口参数, 在输入/输出消息配置完成后, 模块之间出现连接线

17) 分配机械手接口参数

- 自身的 fetching_1, fetching_2, fetching_3, 分别送给 1-3 号成型机, 来传送是否正在拿某台成型机的工件
- 自身的 Product_ok_1, Product_ok_2, Product_ok_3, 分别接受 1-3 号成型机工件就绪的信号

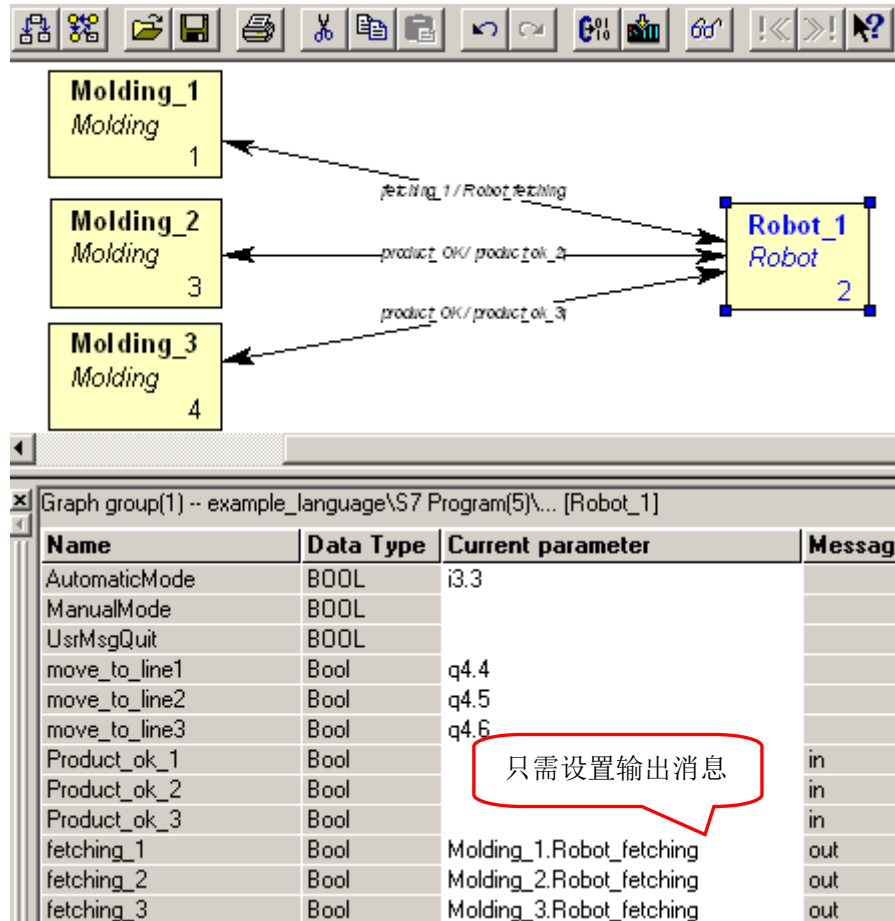


图 50: 分配机械手接口参数

- 18) 在菜单中 Options->Settings for graph groups/state graphs->Compile 设置编译生成的 FC 和 DB 序号。



图 51: 设置 GRAPH 参数

- 19) 编译, 在 OB1 中调用 FC30, 并下载

- 20) 监控 graph group

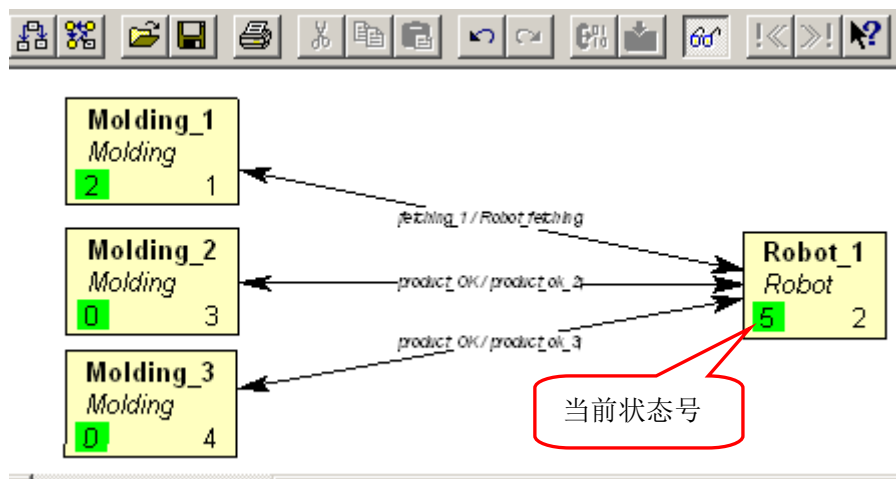


图 52: 监控 graph group

21) 双击 molding_1, 可以监控每个状态图

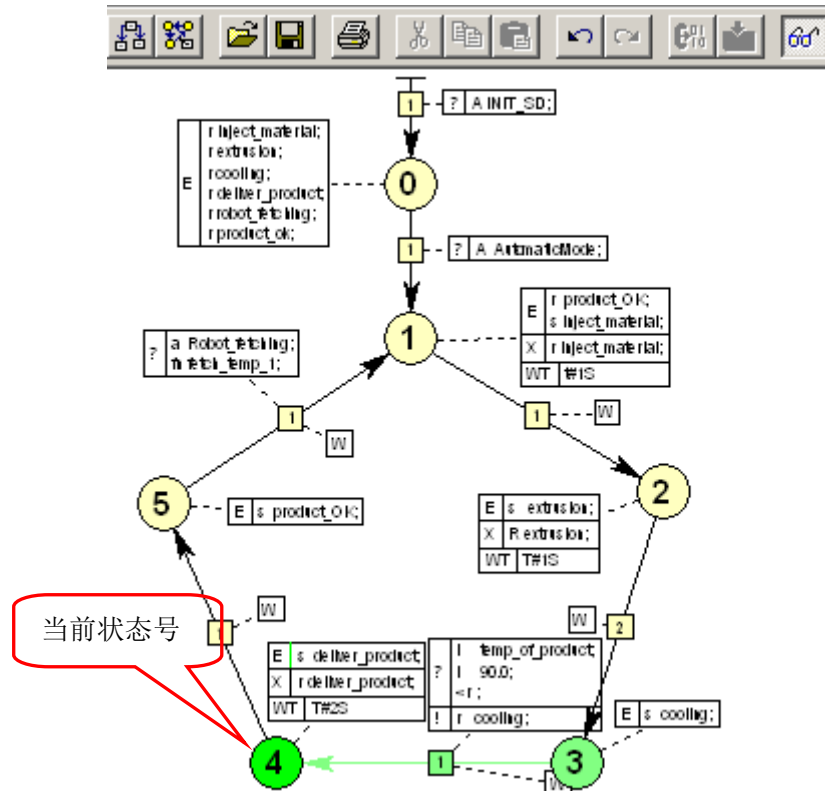


图 53: 监控 state graph

22) 在菜单中的 Debug -> Select variable, 选择关心的变量

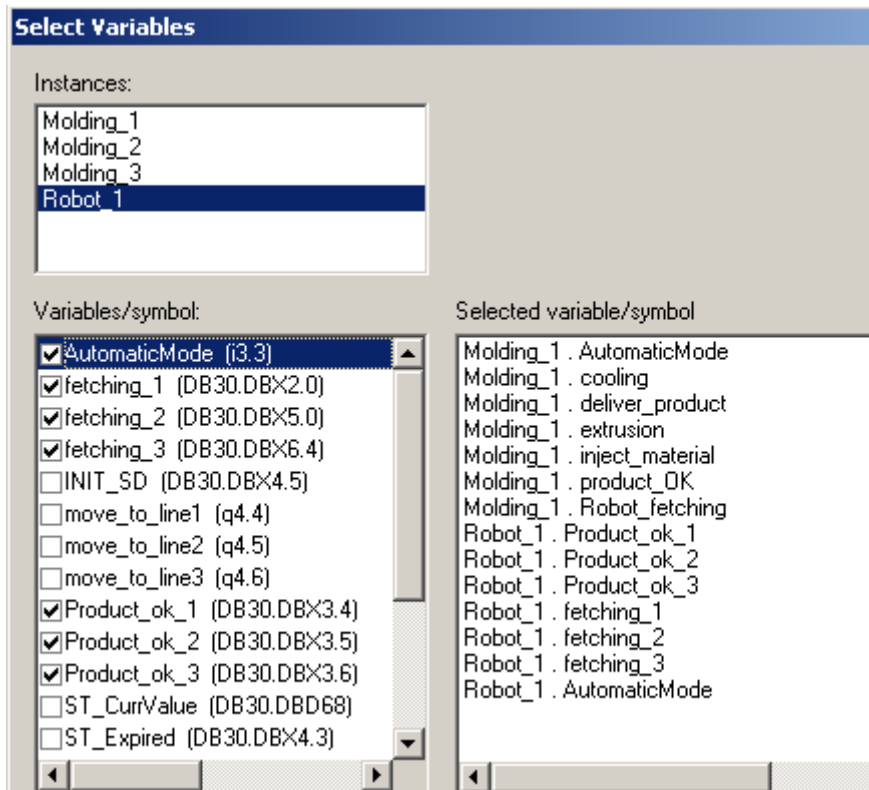


图 54: 选择监控变量

23) 确认后，系统使用变量表进行监控

	Address	Name	Symbol	Display format	Status value
1	I 3.0	AutomaticMode	"molding1_auto"	BOOL	true
2	Q 3.6	cooling	"cooling1"	BOOL	false
3	Q 4.1	deliver_product	"deliver_product1"	BOOL	false
4	Q 3.3	extrusion	"extrusion1"	BOOL	false
5	Q 3.0	inject_material	"inject_material1"	BOOL	true
5	DB30.DBX 3.4	Product_ok_1		BOOL	false
7	DB30.DBX 2.0	fetching_1		BOOL	false
8	DB30.DBX 3.4	Product_ok_1		BOOL	false
9	DB30.DBX 3.5	Product_ok_2		BOOL	false
10	DB30.DBX 3.6	Product_ok_3		BOOL	false
11	DB30.DBX 2.0	fetching_1		BOOL	false
12	DB30.DBX 5.0	fetching_2		BOOL	false
13	DB30.DBX 6.4	fetching_3		BOOL	false
14	I 3.3	AutomaticMode	"robot_auto"	BOOL	true
15	DB10.DBD 8		"模拟量".temp_of_pr	FLO...	0.0

图 55: 监控变量

至此，一个简单的 HiGraph 程序示例就结束了，限于篇幅，本文对例子程序作了简化处理，并没有真正体现 HiGraph 优势所在。下图为 HiGraph 手册中的一个状态图，是一个典型应用例子：

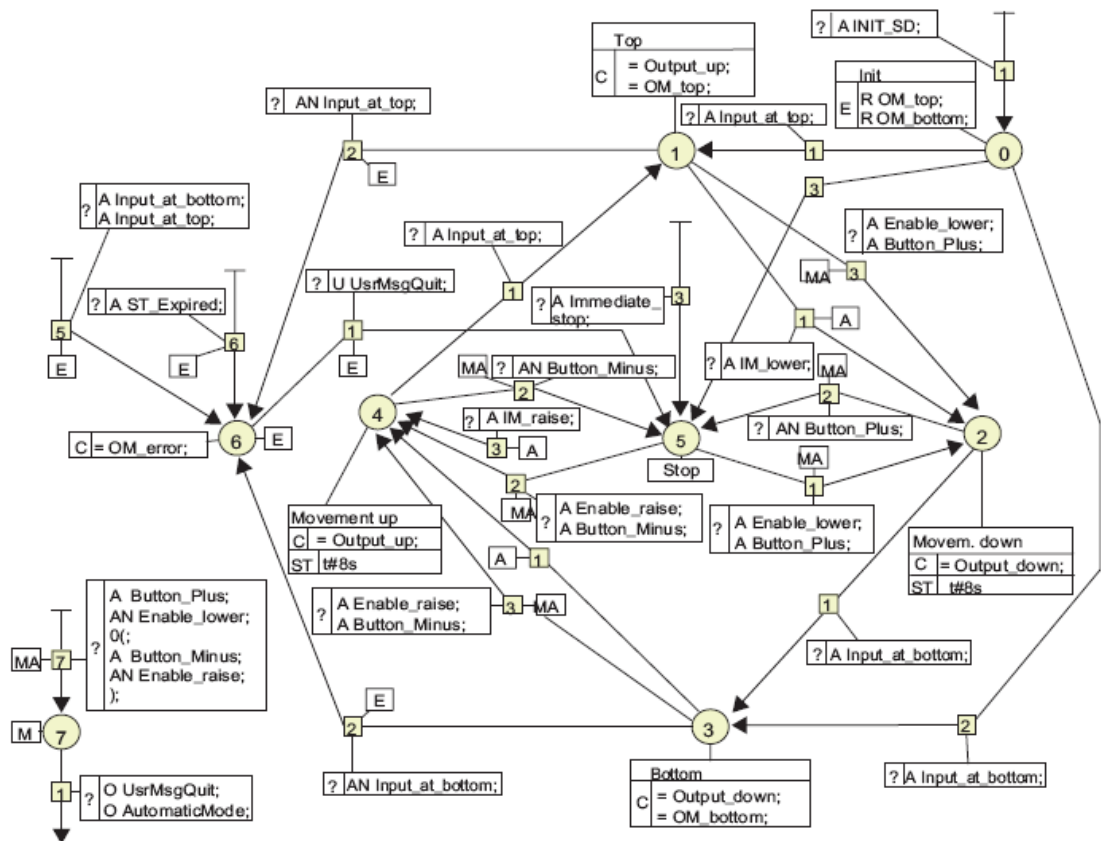


图 56: HiGraph 应用例子

任何编程语言都有其复杂性，并非一朝一夕就可掌握，关于 HiGraph 的具体使用，请按照本文提供的地址连接下载 HiGraph 手册。

4.4.6. 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。

4.5. S7-CFC

4.5.1. S7-CFC简介

S7-CFC (Continuous Function Chart 连续功能图表) 具有以下特点:

- ◇ 通过绘制功能图表来自动生成程序
- ◇ 拥有强大的预制程序库，同时也可使用STEP7中的标准块
- ◇ 通过简单的连线来降低开发成本并减少错误
- ◇ 优化集成在STEP7中，与STEP7 兼容
- ◇ 适用于 SIMATIC S7-300 (推荐用于CPU314以上CPU), S7-400, C7 and WinAC

S7-CFC 不仅仅具有 PLC 典型的元素 (例如 输入 / 输出, 定时器, 计数器, 符号表), 而且具有图形化编程语言语言的特性, 其非常适合于如下任务:

- ◇ 过程控制
- ◇ 系统工程

4.5.2. CFC 与SCL及STL:

CFC 会生成 SCL 代码, 最终编译成 STL。其代码量相对于 STL 编程有所增加。

4.5.3. CFC的安装与使用:

STEP7 各个版本均并不包括 CFC 软件包及授权, 需单独购买。一般情况下, 在 S7 程序中, S7-CFC 会组织整个项目, 调用其它编程语言生成的块。

4.5.4. CFC应用于虚拟工程

下面将以第 2 章中的虚拟工程中的产品合成环节来举例说明 CFC 的使用，此工艺环节要求：

- ✧ PID 控制

4.5.5. CFC简单示例

在下面的例子中，将编写一个 CFC 用于 PID 控制的程序：

- ✧ PID 程序编写在 OB35 中，并调用工艺参数环节的SCL程序及STEP7 标准功能块
- ✧ 完整虚拟工程的程序可编写在 OB1 中（限于篇幅，省略）

程序流程：

- ✧ 将模拟量输入进行 WORD 至 INT 的转换，送至软件滤波功能块的输入
- ✧ 将模拟量进行工程量转换，选择是否进行软件滤波，否则将实时值送至PID模块
- ✧ 滤波值/实时值的工程量作为过程变量送至PID功能块
- ✧ PID 功能块的输出送至 FC106 进行输出值转换
- ✧ FC106 的输出值进行INT至WORD转换，送至模拟量输出模板

1) 添加 CFC 程序

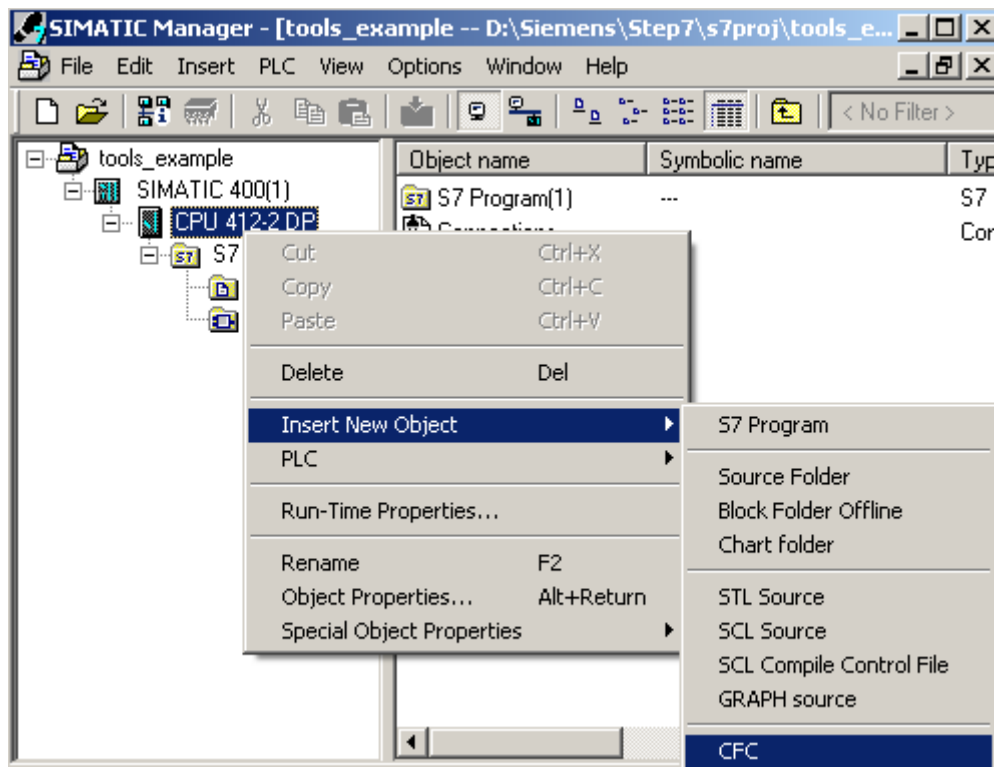


图 57：添加 CFC 程序

- 2) 程序文件夹下增加了 Charts 文件夹及 CFC 文件，双击打开 CFC

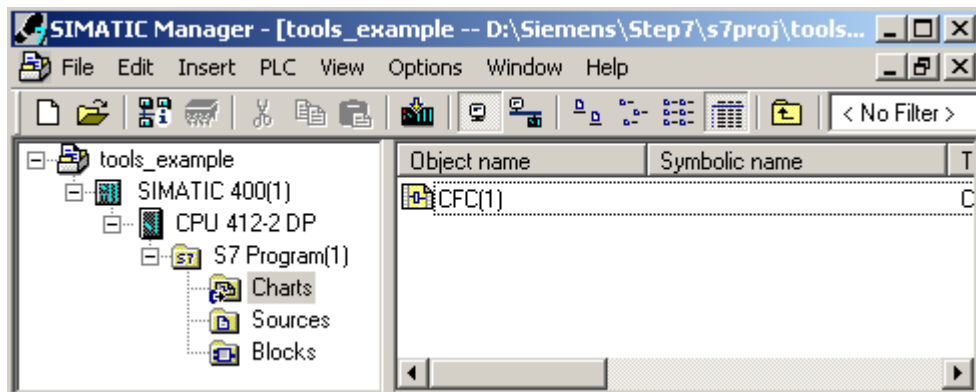


图 58: 打开 CFC

- 3) CFC 页面视图

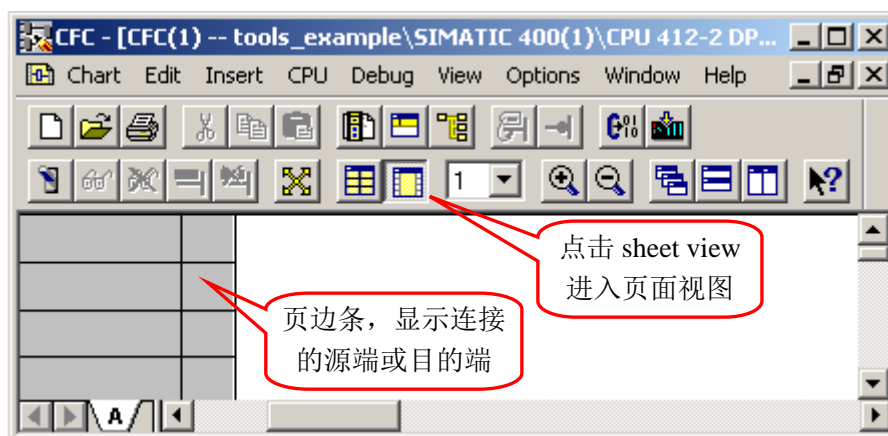


图 59: CFC 页面视图

- 4) CFC 当前所处位置

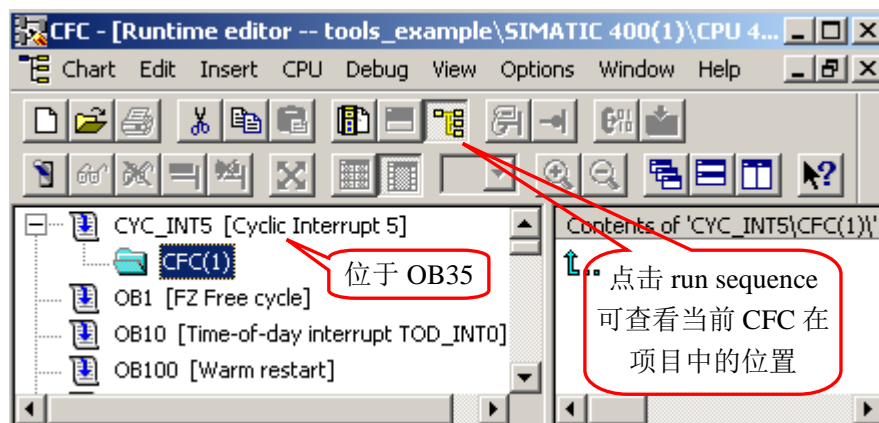


图 60: CFC 当前所处位置

5) 调用其它功能块

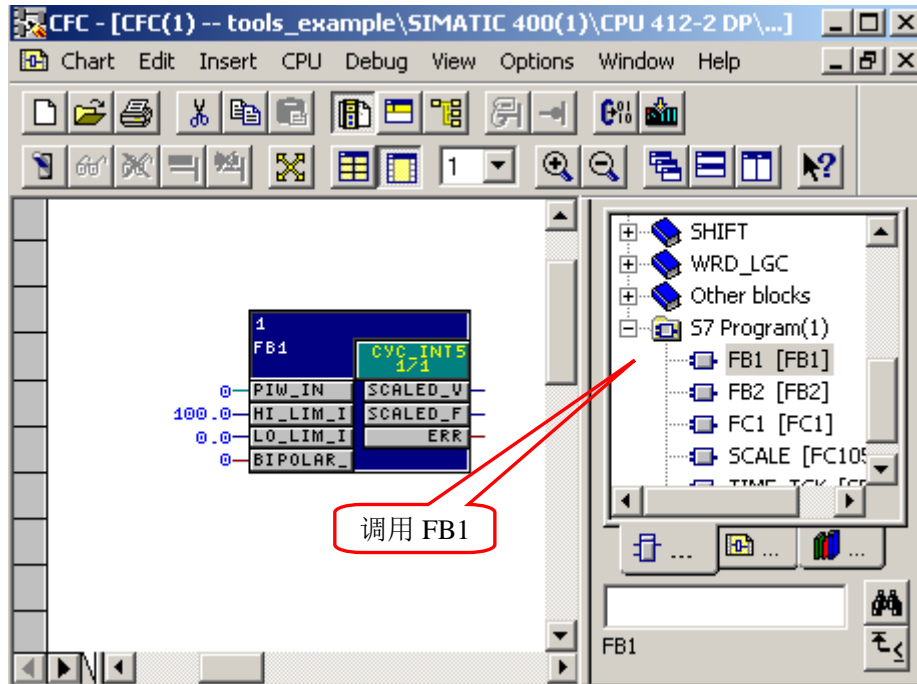


图 61: CFC 调用其它功能块

6) 调用分配地址连接

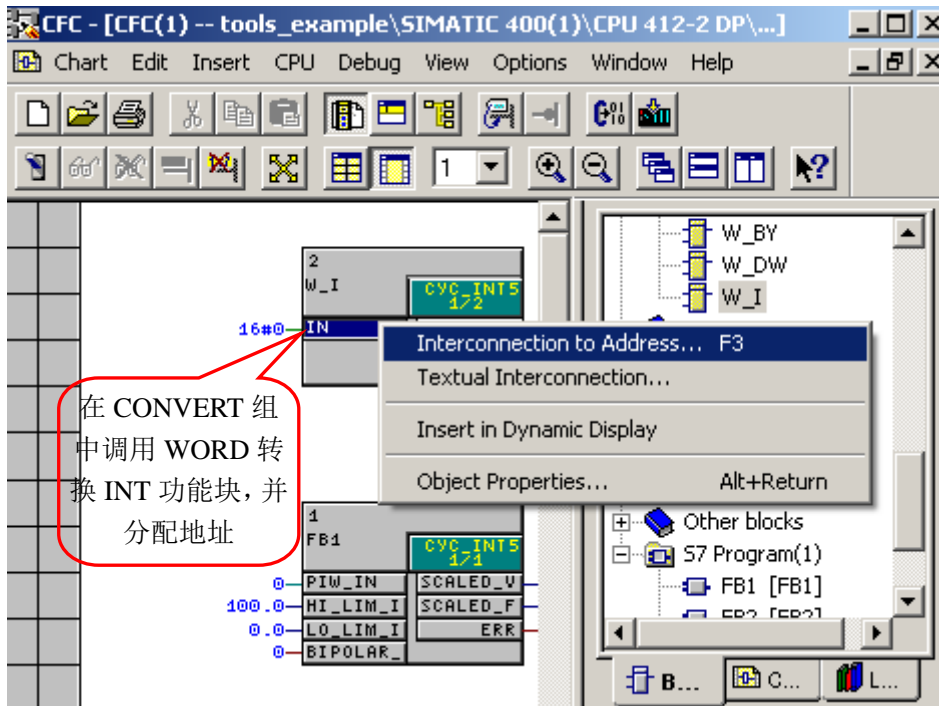


图 62: 分配地址连接

7) 内部连接只要数据类型一致，直接拖拽就可建立连接

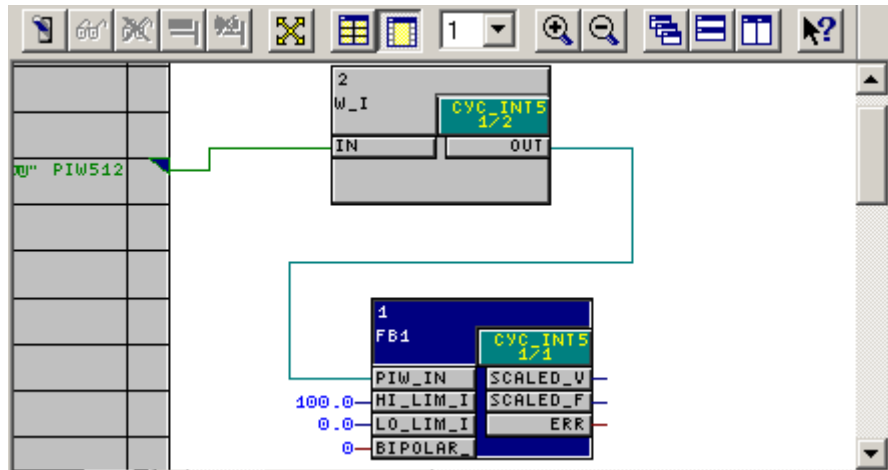


图 63: 内部连接

8) 调用 CFC Library > ELEM 400 CPU 程序库中的 PID 模块，此处由于和前面的 FB1 重名，更改名称为 FB10

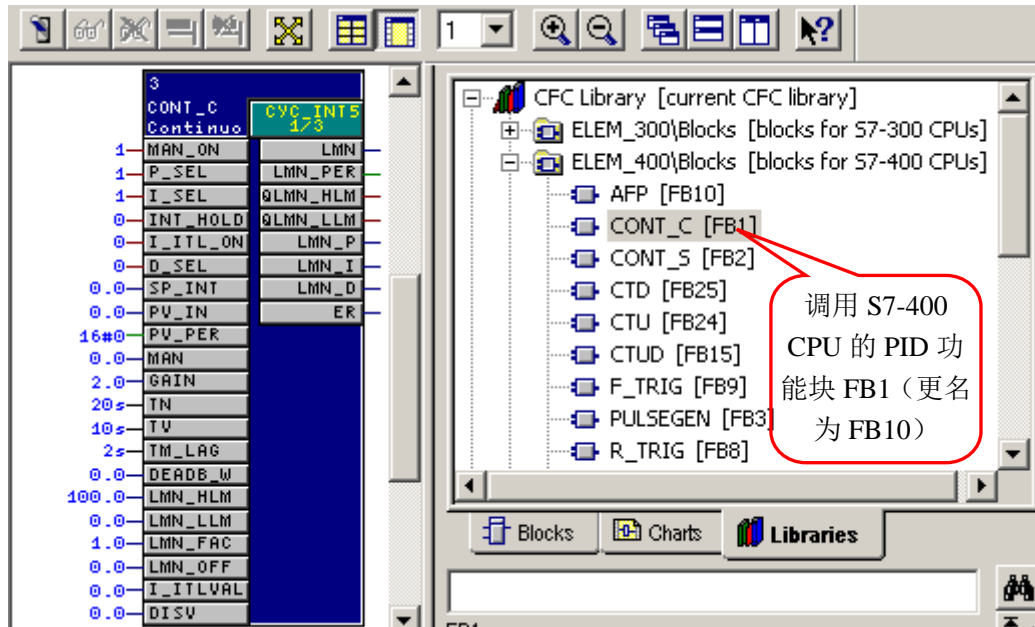


图 64: 调用 CFC 程序库

- 9) 依次调用 CFC 块中的 MULTIPLX>SEL_R 进行实数选择，标准程序库中的 Standard library>TI-S7 Converting Blocks 中的 FC106 进行模拟量输出，Conversion INT -> WORD 进行数据类型转换。

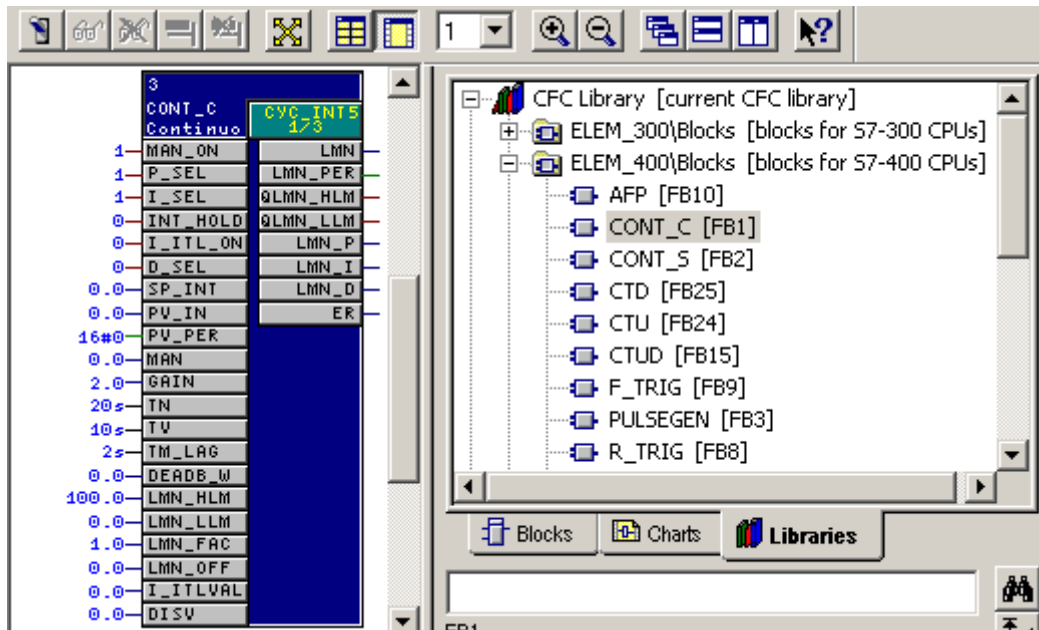


图 65: 调用 STEP7 标准程序库

- 10) 模拟量输入滤波

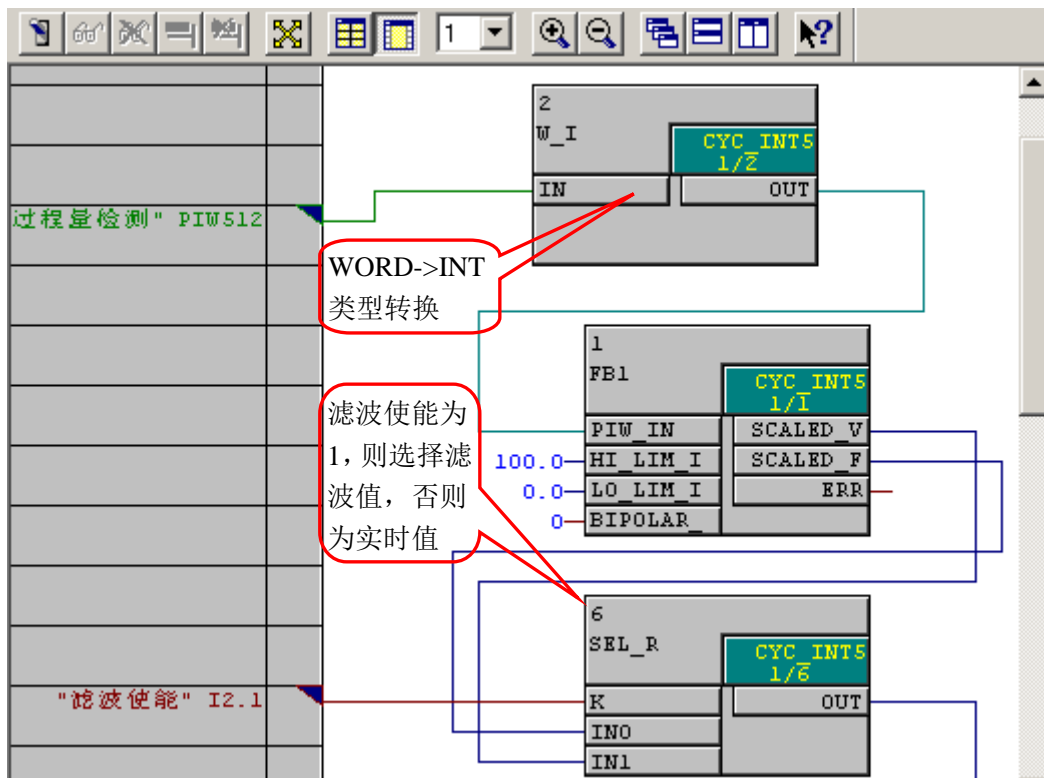


图 66: 模拟量输入滤波

11) 手动/自动切换无扰动处理，将 LMN 值送至 MAN

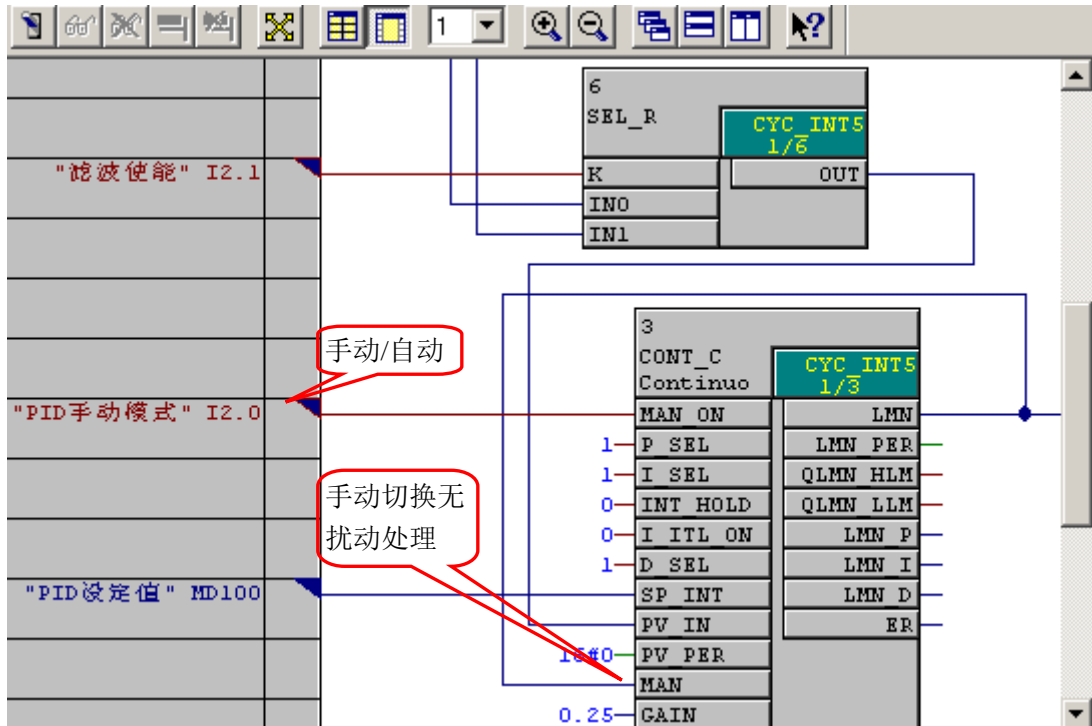


图 67: 手动/自动切换无扰动处理

12) 输出处理

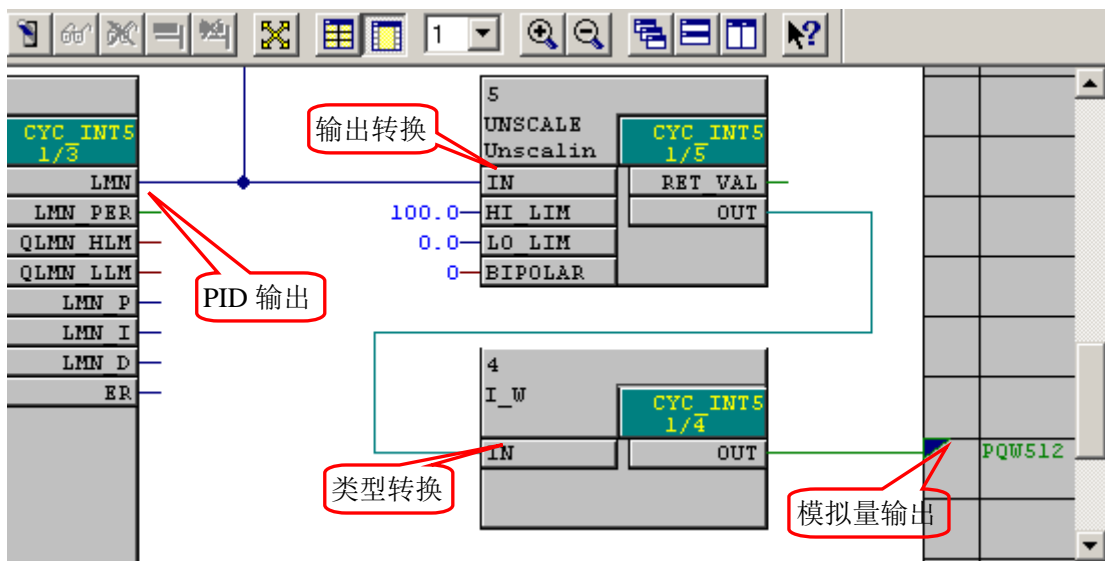


图 68: 输出处理

13) 程序图，连接关系标号可以减少交叉连线，连接关系标号一致则相通

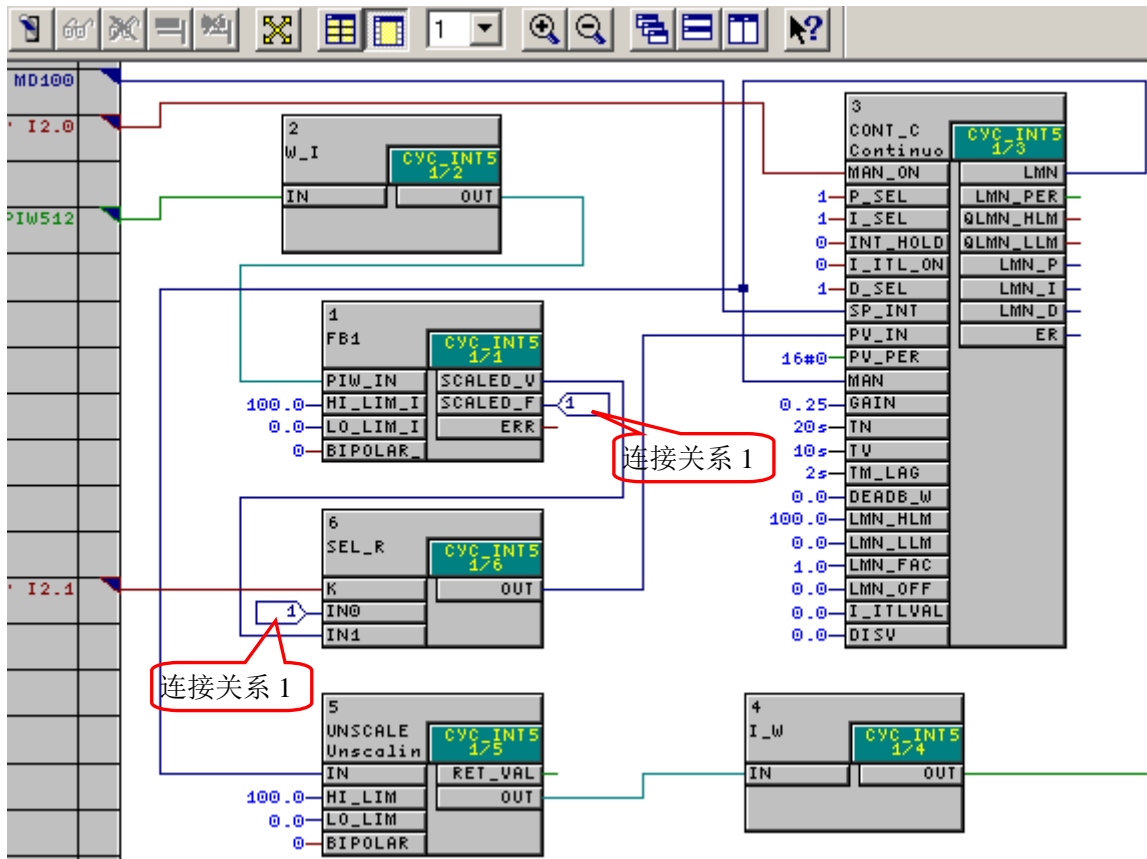


图 69: 程序图概览

14) 将原来的 OB35, OB100 中的程序删除，准备编译，打开菜单 Options -> Customize -> Compile/Download 选项，这里将 FC number from 更改为 1-150。

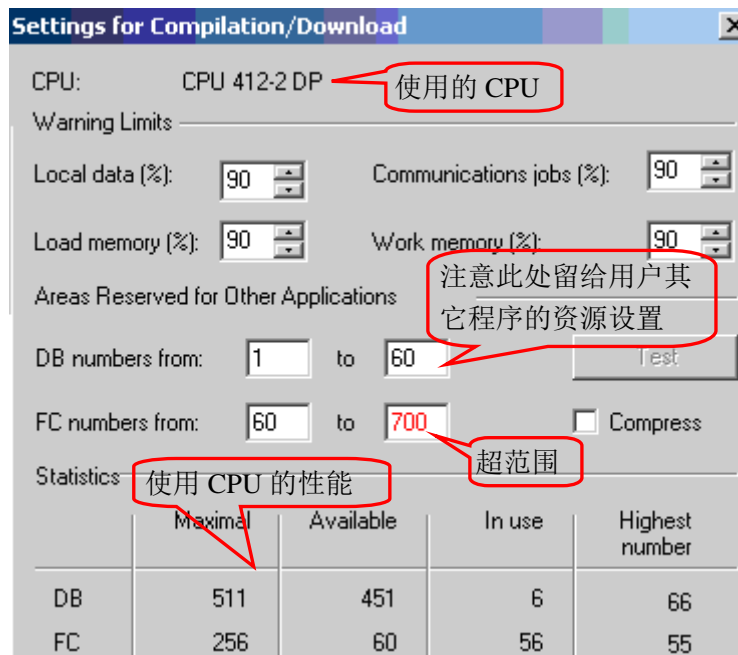


图 70: 编译/下载设置

- 15) 选择 File->Compile, 经过编译没有错误后, 项目 Blocks 目录下会自动生成很多 SCL 编写的程序块及数据块, OB35, OB100 也变更为 SCL 编写。

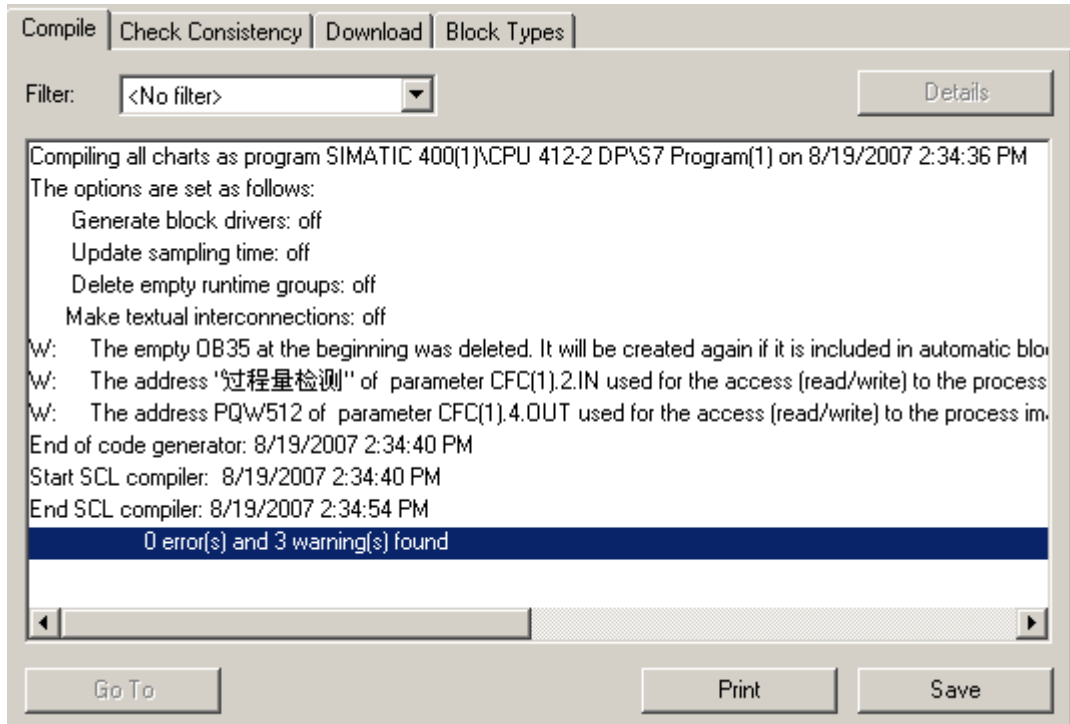


图 71: 编译程序

- 16) 下载程序后, 在菜单 Debug-> Test Mode 更改为测试模式, 监控程序

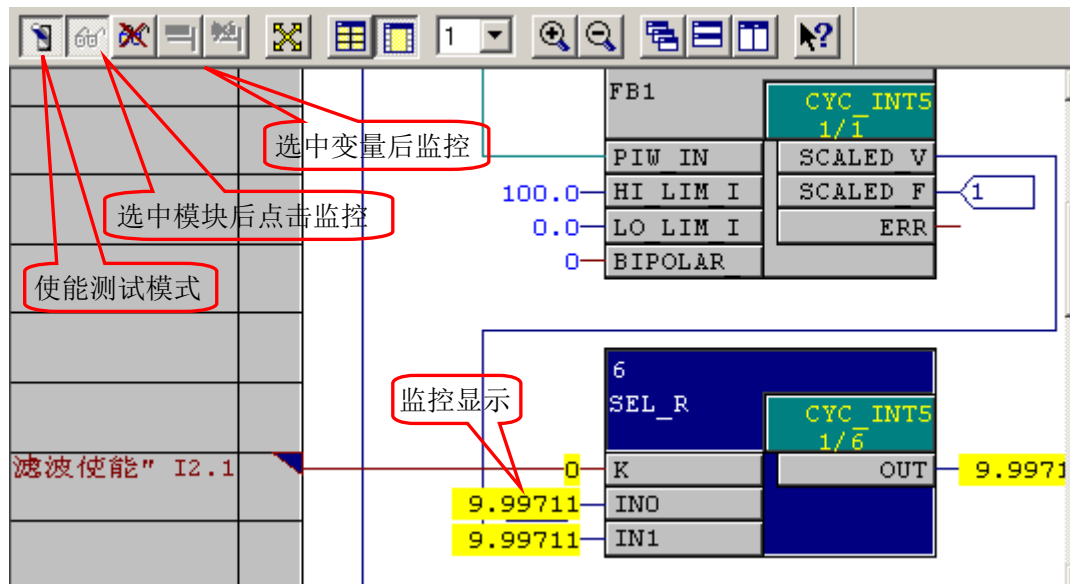


图 72: 监控程序

17) 在 View -> Dynamic Display 中可以监控变量

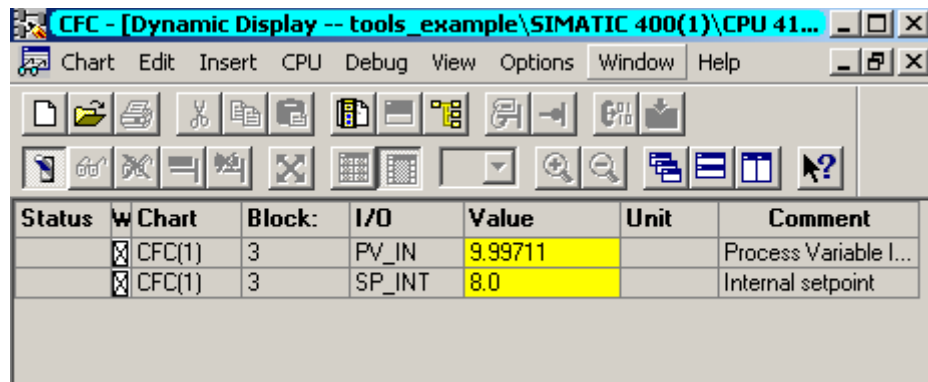


图 73: 监控变量

至此，一个简单的 CFC 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 CFC 的具体使用，请按照本文提供的地址连接下载 CFC 手册。

4.5.6. 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。

4.6. S7-PLCSIM

4.6.1. S7-PLCSIM简介

使用 S7-PLCSIM 具有以下优点:

- ◇ 在PG/PC上进行不依赖于硬件的S7程序测试
- ◇ 在程序开发早期消除错误
- ◇ 降低开发成本，加速开发进程，提高程序质量
- ◇ 适用于 LAD, FBD, STL, S7-GRAPH, S7-HiGraph, S7-SCL, CFC, S7-PDIAG, WinCC (本地安装)

4.6.2. PLCSIM与真实PLC:

PLCSIM 并不能完全代替真实的 PLC, 它与真实的硬件 PLC 有着如下的差别:

- ◇ 当对PLCSIM进行“STOP”操作后, 程序再开始时, 从中断处开始执行
- ◇ 当对PLCSIM进行“STOP”操作时, 不影响输出状态
- ◇ 当在子窗口修改变量时, 其修改立刻有效, 而不会等到下个周期
- ◇ 你可以手动修改或复位定时器的值
- ◇ 可以实现单周期操作模式
- ◇ 你可以触发中断OB块
- ◇ 对过程映像区的修改立刻生效
- ◇ 不支持所有的诊断信息, 例如EEPROM错误
- ◇ PLCSIM 提供高档 CPU 才拥有的系统资源 (例如定时器范围为 T0-T2047, M 范围为 16KB), 所以当使用PLCSIM 模拟通过的程序 (假设使用了定时器 T2000), 可能会无法下载到低档CPU上运行 (例如CPU315-2AG10-0AB0定时器范围为T0-T255,)。
- ◇ **不支持FM功能模块**
- ◇ **不支持通信功能**
- ◇ PLCSIM类似于400有4个累加器, 所以不同于仅有2个累加器的300CPU
- ◇ 对于调用以下块, PLCSIM 执行空操作:
 - SFB12, SFB13, SFB14, SFB15, SFB16, SFB19, SFB20, SFB21, SFB22 and SFB23,
 - SFC7, SFC11, SFC25, SFC35, SFC36, SFC37, SFC38, SFC48, SFC60, SFC61,
 - SFC62, SFC65, SFC66, SFC67, SFC68, SFC69 and SFC81
 - OB81, OB84, OB87 and OB90

关于 PLCSIM 与真实 PLC 的详细差异, 请参考以下地址链接:

<http://support.automation.siemens.com/CN/view/zh/11610958>

4.6.3. PLCSIM安装与使用:

STEP7 标准版并不包括 S7-PLCSIM 软件包及授权, 需单独购买, STEP7 Professional 版包括了 S7-PLCSIM 的软件包及授权, 安装即可。在菜单 Options 中, 可以激活 PLCSIM, 此时再进行上传/下载/监控等操作就是针对与 PLCSIM 了, 而不会对真实 PLC 进行操作 (不论 PLC 是否联机)。

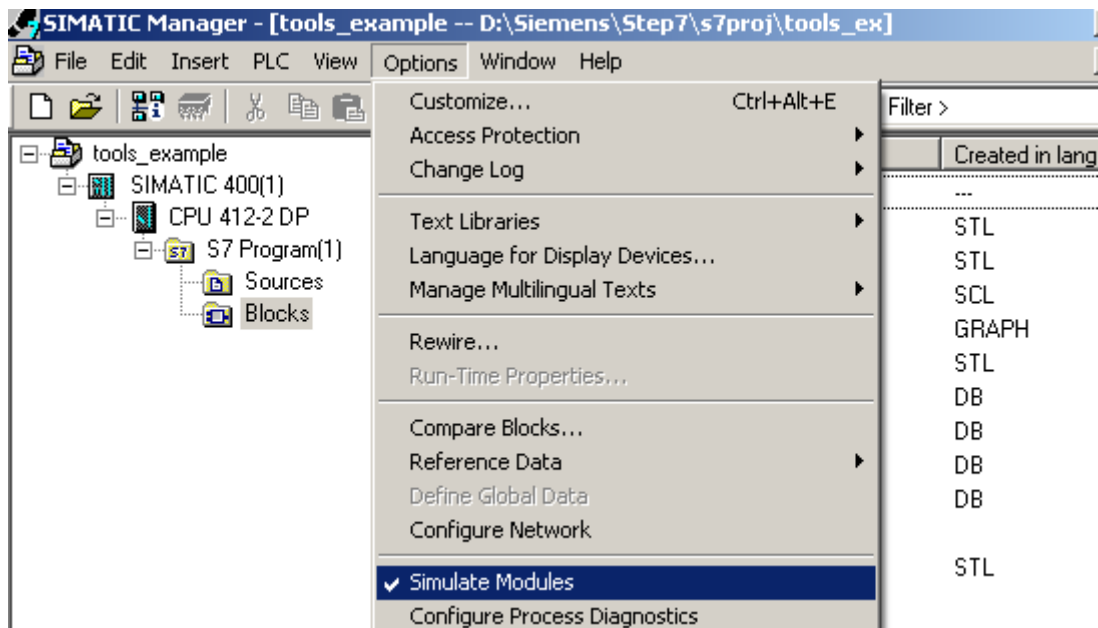


图 74: PLCSIM 调用

4.6.4. S7-PLCSIM用于虚拟工程

PLCSIM 可以用在第 2 章中的虚拟工程的整个调试环节。可以通过修改各种输入条件来模拟各种工况。

4.6.5. S7-PLCSIM简单示例

下图为 PLCSIM 工作界面，用户可以通过 Insert >Input Variable 插入变量（输入/ 输出/ 中间寄存器/定时器/计数器/数据块）方式来模拟各种工况。

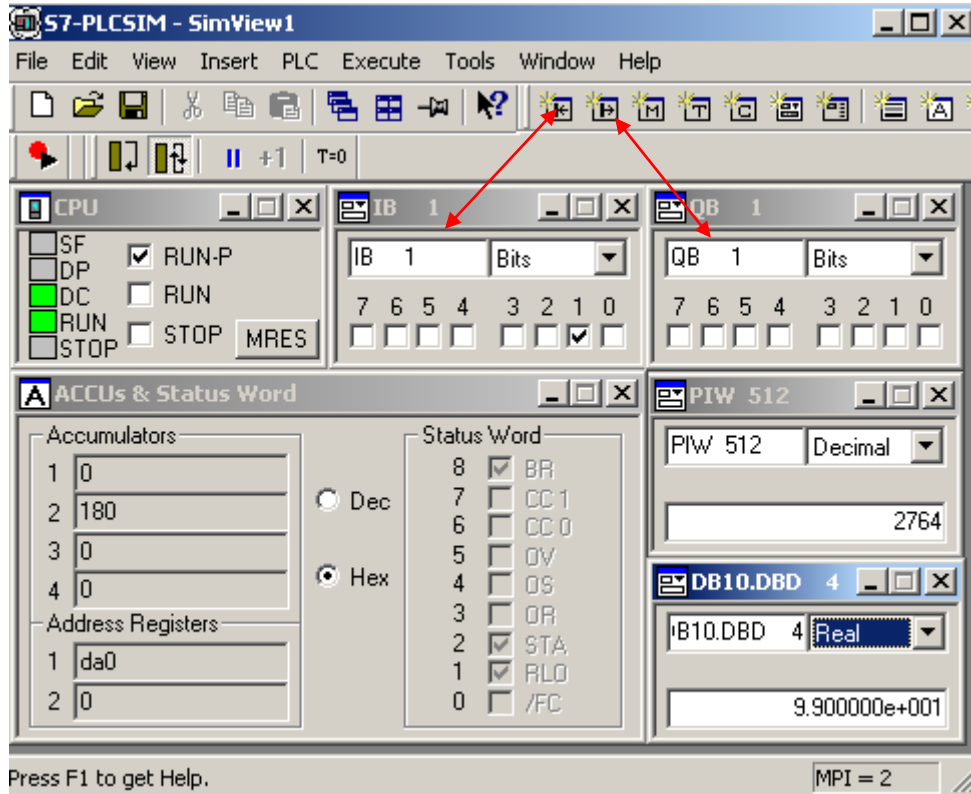


图 75: PLCSIM 视图

4.6.6. PLCSIM主要功能:

- 1) File> Save PLC As 可将当前模拟的 PLC 存储为一个文件,下次使用时可以直接打开此文件,而不需要下载过程,方便调试
- 2) PLC>电源打开/关闭,清除内存/复位功能, MPI 地址设置
- 3) Execute>Scan Mode 可更改扫描模式,此功能为 PLCSIM 特有功能,可单步扫描
- 4) Execute>Trigger Error OB 触发错误组织块
- 5) Tools>Record/Playback 录制事件文件

PLCSIM 主要用于模拟工况,而即使一个简单的工况也可能是由一定时间段内的各种触发事件组成的。如果重复调试某个工况,而完全依赖于手工操作模拟,是比较困难的。PLCSIM 可以解决这个难题:编程人员可以将手工模拟过程录制成一个事件文件,针对不同的工况,可以录制不同的事件文件。选择不同的事件文件,即可模拟不同的工况,而不必一次又一次地去手动输入。

- 1) 录制事件：此时操作者的每一步操作都会被记录下来

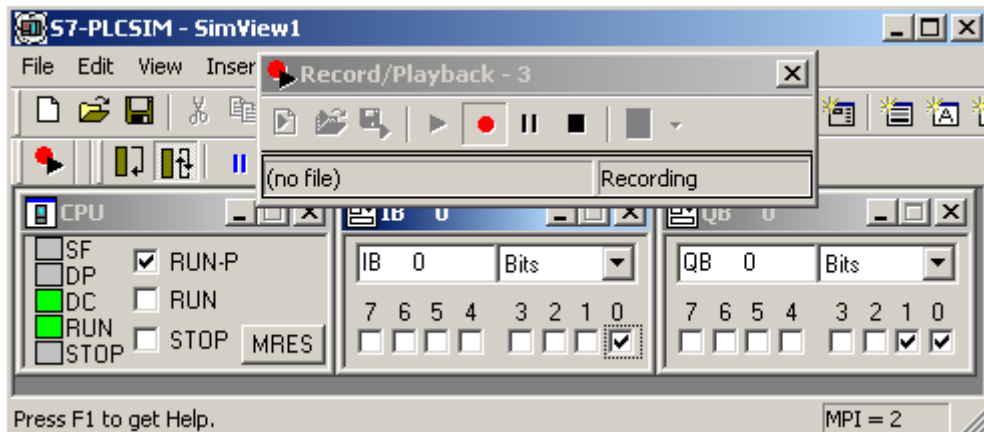


图 76：录制事件

- 2) 回放事件：此时操作者的每一步操作会依次被重现（现在为第 2 个操作）



图 77：事件回放

至此，关于 PLCSIM 的介绍就结束了。PLCSIM 非常有利于工程调试，但务必要注意其与真实 PLC 的区别，切记!!!

5. 总结

西门子推出了很多工程工具，用来解决 PLC 日益面对着越来越复杂的控制任务。除了本文所介绍的工具体之外，还包括：

- ◇ S7-PDIAG
- ◇ TeleService
- ◇ DOCPRO
- ◇ S7-200 PC Access
- ◇ S7 Technology
- ◇ D7-SYS
- ◇ Engineering package Drive ES
- ◇ Distributed Safety Software

由于这些工具在西门子其它文档中已经有所介绍或者超出了本文讨论的范围，这里不再做介绍，关于它们的使用，请参考相关文档。