

1. 前言

本文可以作为 S7-CFC 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-CFC 编程语言。西门子提供了 S7-CFC 编程语言的详尽手册，在安装 S7-CFC 软件包后，通过点击 Windows 菜单 开始->Simatic->Documentation->English 可以阅读到名称为“ CFC - Manual for S7” 的 PDF 手册。此手册共分为 5 个章节，其详细地讲解了 S7-CFC 编程语言。一切关于 S7-CFC 使用的问题请以此手册为准。

相对于其它编程语言，S7-CFC 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-CFC 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-CFC 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。

S7-CFC 可以单独作为编程语言使用，本文所讲述的内容仅限于其普通编程方式。对于 S7-CFC 在西门子的 PCS7 及运动控制系统中的应用。由于 PCS7 及运动控制的一些特性，本文不再讲述。相关资料读者可以参考西门子 PCS7 及运动控制方面的资料。

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-CFC for SIMATIC S7

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

2. 软件的基本信息

2.1. 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
- ✧ 拥有强大的预执行程序库，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

2.2. 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。

2.3. 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。用户也可以在项目中编写独立于 S7-CFC 的程序，单独下载调试这部分程序，但用户应当尽量避免这种用法。

2.4. S7-CFC 软件兼容性

不同 S7-CFC 软件版本与 STEP7 及操作系统之间的兼容性：图中的 X 表示兼容，- 表示不兼容

CFC			STEP 7 V5.3			STEP 7 V5.4				
Product Name	Version	Order Number	Win 2000 SP4	Win XP SP1	Win XP SP2	Win 2000 SP4	Win XP SP1	Win XP SP2	Win 2003	Win 2003 SP1
CFC	V7.0	6ES7 658-1EX07-2YA5	X	X	X	X	X	X	X	X
CFC	V6.1	6ES7 658-1EX16-2YA5	X	X	X ¹⁾	X	X	X ¹⁾	X ¹⁾	-
CFC	V6.0	6ES7 658-1EX06-2YA0	X	X	X	X	X	X	-	-
CFC	V5.2	6ES7 813-0CC05-0YX0	X	-	-	(-)	-	-	-	-

表 2-1 S7-CFC 软件兼容性

注：1) 仅仅 S7-CFC V6.1+SP1 或以后版本支持

2.5. S7-CFC 中英文词汇对照关系

由于很多英文的科技专用词汇没有明确统一的中文词汇，所以在本文的讲解当中，尽量保持手册中的英文信息。本文中将尽量减少使用中文词汇代替英文专用词汇，需要代替的中英文词汇对照关系如下：

- Chart 图表
- Chart Partition 图表分区
- Nested Charts 嵌套图表
- Sheet Bar 页边条
- OS 操作员站
- AS 自动化站（控制器）

3. S7-CFC 基本概念与基本使用

3.1. S7-CFC 程序构成元素

作为 STEP7 的选项包，S7-CFC 软件在安装后，将被集成在 STEP7 中使用。S7-CFC 编程界面为图形界面，包含若干个 Chart。用户在 Chart 中以图形的方式插入已经编制好的块，指定这些块之间的连接关系，也可以为这些块分配输入/输出地址。在这种通过鼠标点击托拽的方式完成项目编写后，即可编译项目。S7-CFC 程序将被整体编译，其生成的块可以以 SCL 代码的形式出现（最终将生成 STL 代码）。用户将这些代码下载到 PLC 中即可执行。用户可以指定这些程序的运行顺序及运行特性。

3.1.1. Charts（图表）

作为 STEP7 的选项包，S7-CFC 软件在安装后，将被集成在 STEP7 中使用。S7-CFC 编程界面为图形界面，其基本元素为 Chart，每个 Chart 在 CPU 中的名字都是唯一的。可以在 SIMATIC Manager 或 CFC 编辑器中生成 Chart。

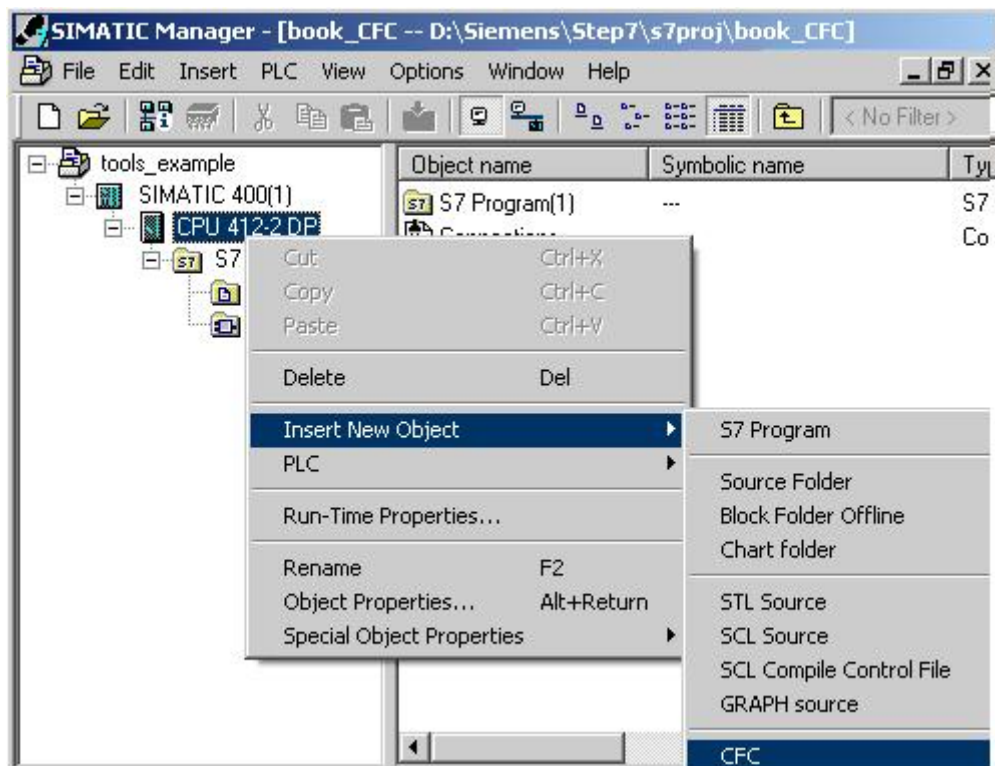


图 3-1: 生成新 S7-CFC 程序

通过插入 CFC 程序，程序文件夹下增加了 CFC 的基本元素 Charts 文件夹及 CFC Chart。可以通过重复插入 CFC 的操作，即可生成多个 Chart，它们的名称在项目中是唯一的。双击其中的

Chart，可以打开 S7-CFC 程序 (例如 CFC (1))。

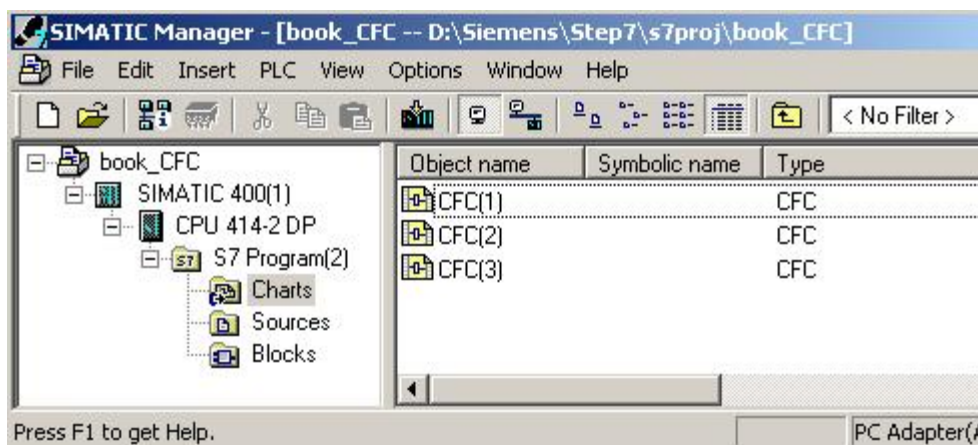


图 3-2: 多个 Chart 的视图

在 CFC 编辑器中，也可以在 Chart 中点击右键，添加 Chart，注意：这种方法生成的 Chart 为嵌套 Chart，不同于前面生成的普通 Chart。

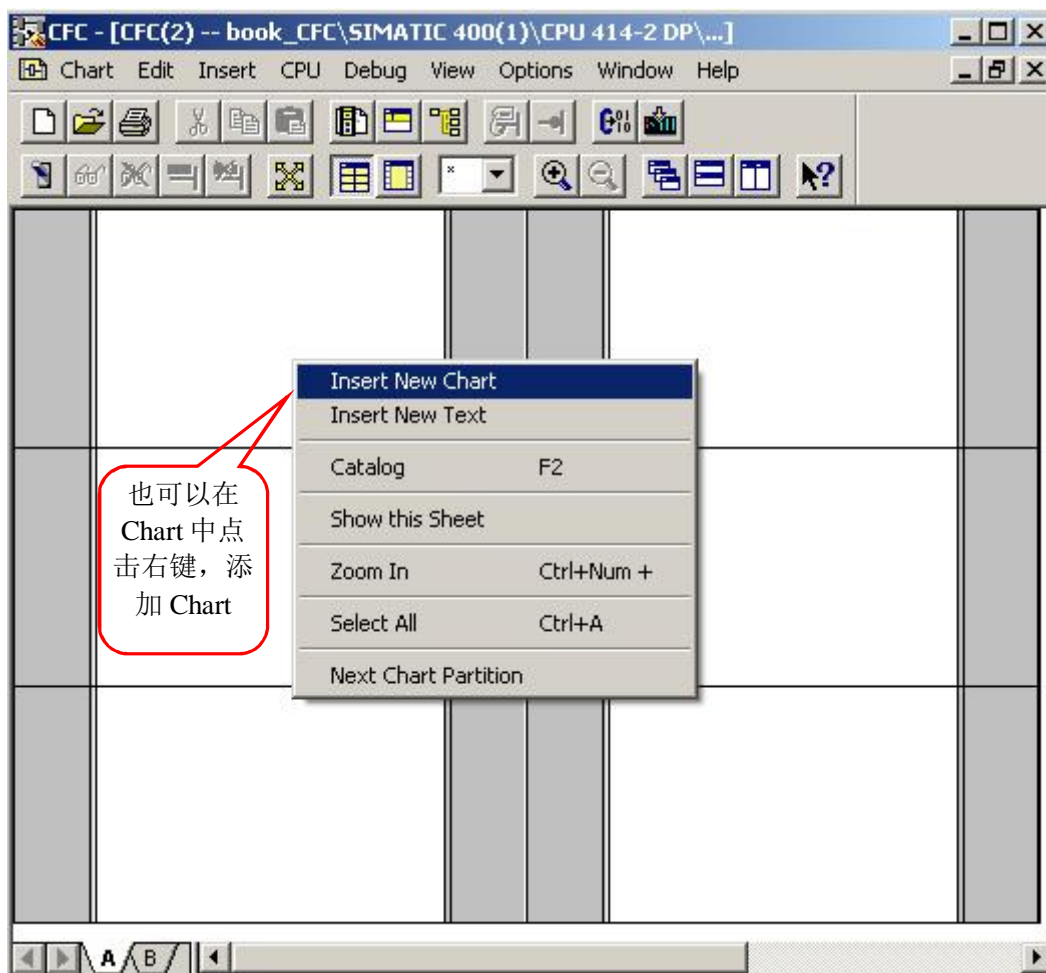


图 3-3: 在 CFC 编辑器中添加 Chart

3.1.2. Chart Partitions (图表分区)

每个 Chart 最多可以包括 26 个 Chart Partition，每个 Chart Partition 都有自己的名字，用字母表

示。可以在 CFC 编辑器中，右键点击 Chart Partition 的名称，然后添加新的 Chart Partition。

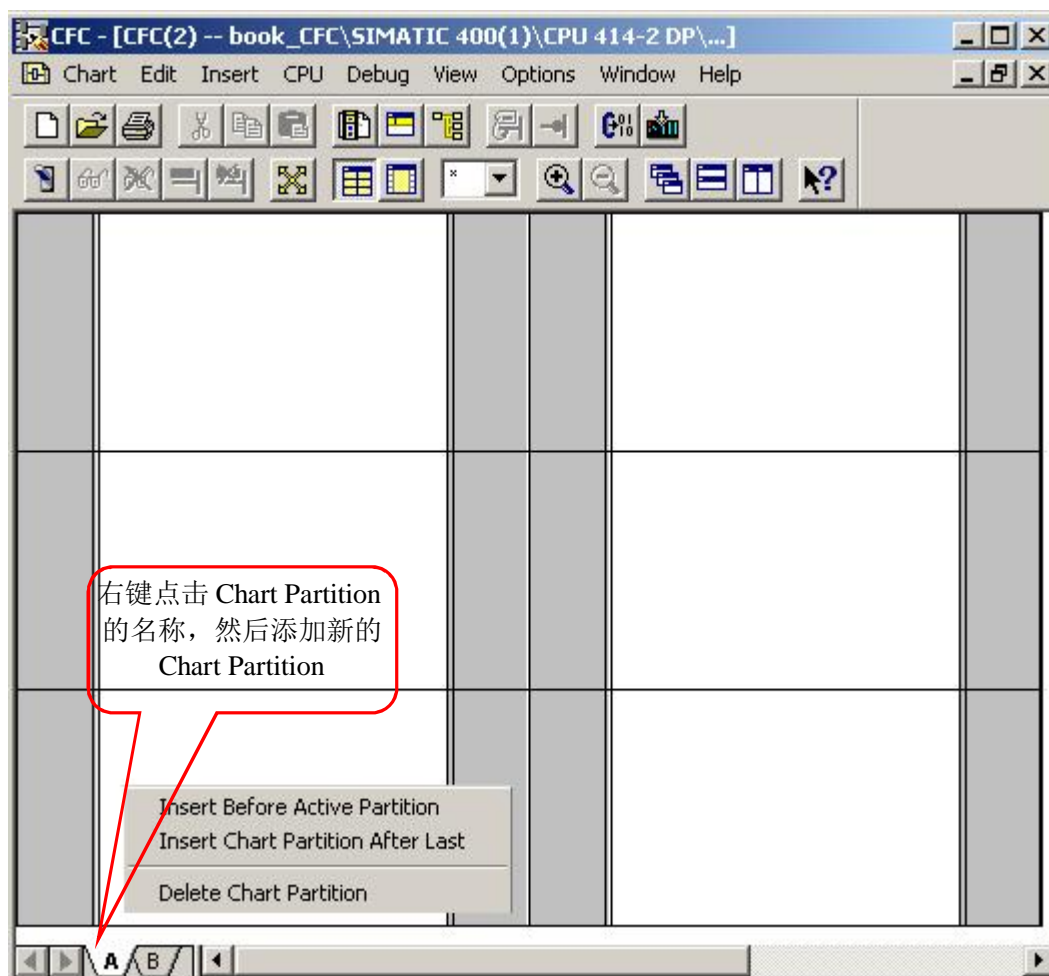


图 3-4: 在 Chart 中添加 Chart Partition

3.1.3. Sheet（页）及 Sheet Bars（页边条）

每个 Chart Partition 包括 6 个 Sheet，它们按照 3 行 2 列的形式排列。在 CFC 编辑器中，可以通过点击缩小按钮，将视图缩小到最小，此时可以清楚地看出 Sheet 及 Sheet Bars 的排列位置。每个 Sheet 包括一个中央空白工作区及两侧的 Sheet Bars。用户可以空白工作区放置需要的功能块，在两侧的 Sheet Bars 中可以指定这些功能块的连接关系及 Chart 与 Chart 之间的连接关系。

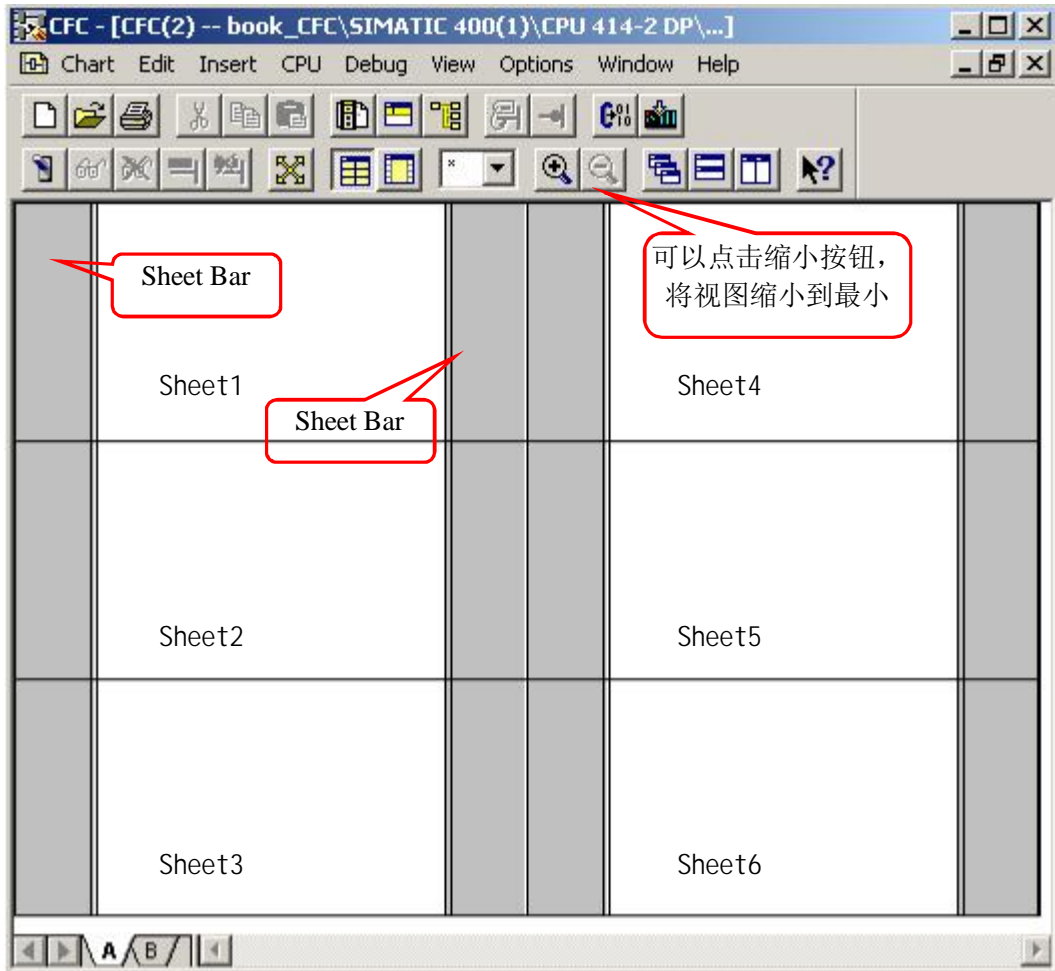


图 3-5: Sheet 及 Sheet Bar

可以通过点击放大按钮, 将视图放大到最大, 此时可以清楚地看出 Sheet 的结构。

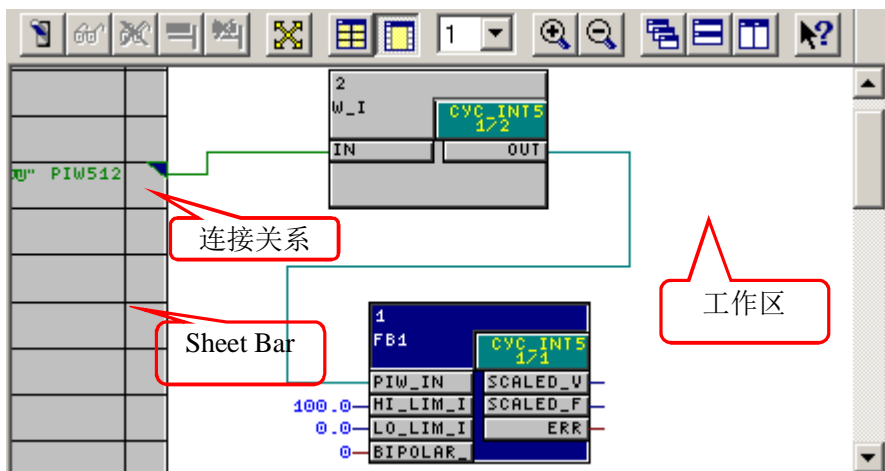


图 3-6: Sheet 构成

3.1.4. Overflow Page (溢出页)

当一个 Sheet 中包含太多的与其它 Sheet 的连接时, 会出现 Sheet Bar 被完全占用的情况, 这时系统会自动生成 overflow page, 这部分仅作为 Sheet Bar 的扩展出现。

3.1.5. Nested Charts (嵌套图表)

在一个 Chart 中可以嵌入另外一个 Chart, 用户可以指定此 Chart 的输入/输出及连接关系, 操作方法可以参考后续章节中的详细介绍。

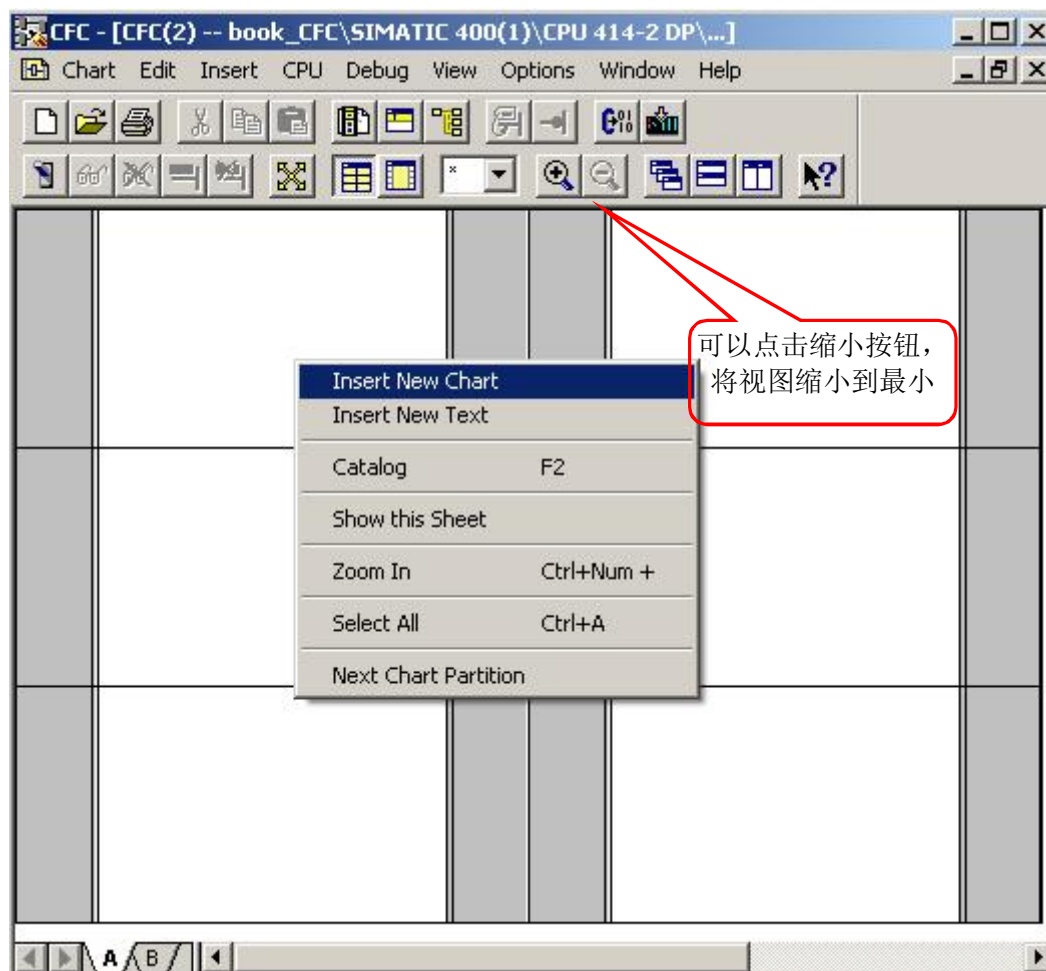


图 3-7: 嵌套图表

3.2. 操作 S7-CFC 程序块

3.2.1. 程序块的导入

在 CFC 编程界面的 Blocks 的视图中, 包含了 CFC 集成的块, 类似于 LAD 编程环境, 用户可以将这些块拖拽到工作区来使用。如果用户希望使用项目中其它的块, 例如本项目 S7 Program(2)\Blocks 目录中的 FB1, FC105 等等, 则需要通过导入的方式加入到 CFC 编程界面中。本例 CFC 编程界面中的 Program (2) 目录, 在导入其它块之前, 是没有功能块的。如下图:

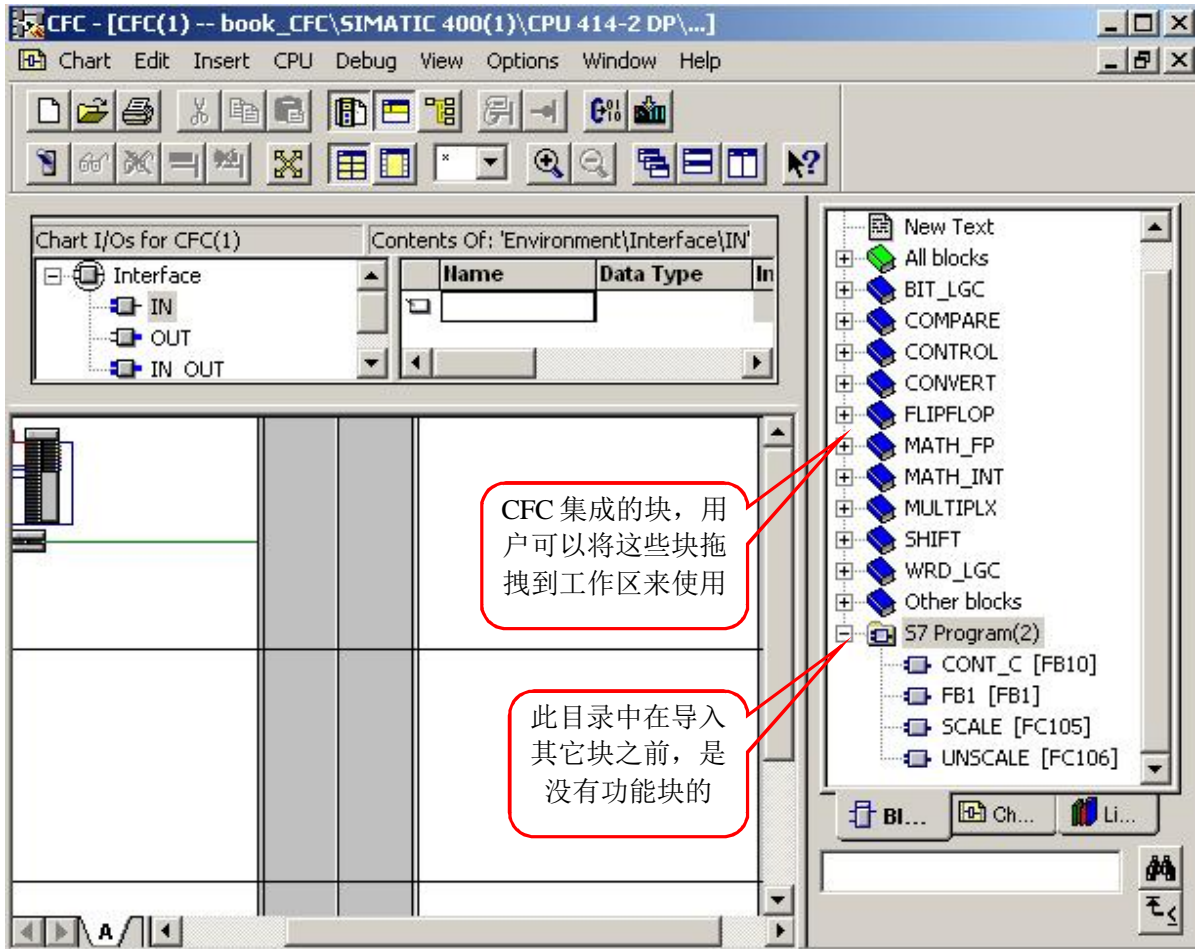


图 3-8: CFC 的 Blocks 视图

在菜单 Options 中，选择 Blocks Types...，此时本项目 S7 Program(2)\Blocks 目录中的 FB1, FC105 等等便被显示在对话框的左侧，选择需要的块，点击中间的箭头，则依次可以将需要的块导入到 CFC 编程界面中

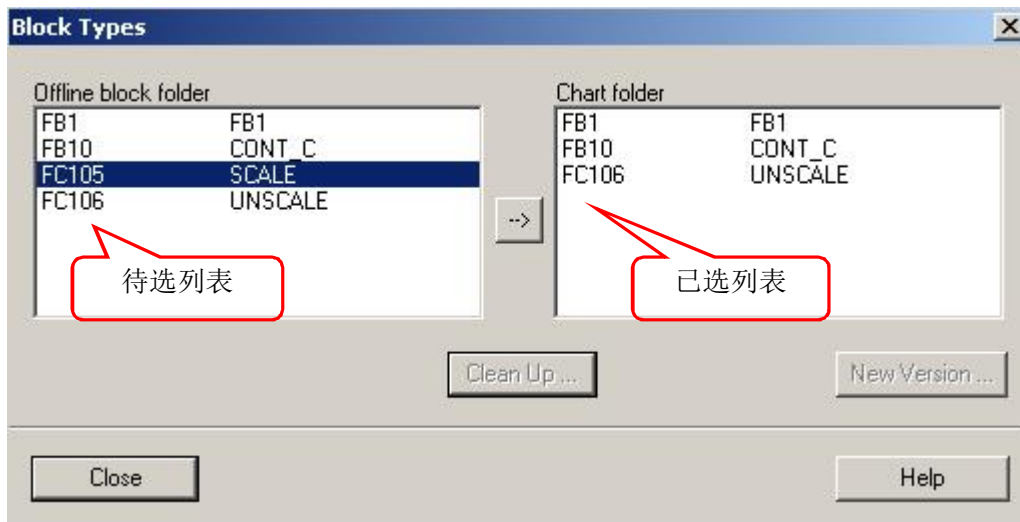


图 3-9: CFC 的 Blocks 视图

3.2.2. 程序块的清除与更新

如果用户认为在 CFC 编程界面下存在不必要的块，可以通过 Clean Up...按钮来清除在 CFC 编

程环境下未使用的块（CFC 程序中已引用的块无法清除）。



图 3-10: CFC 的 Blocks 清除与更新

如果用户在本项目 S7 Program(2)\Blocks 目录中的 FC105 有更改，则可以通过界面中的 New Version...，来重新导入 FC105。

当用户重新导入块的新版本时，可以分为 3 种情况：

1. 不需要编译及下载完整程序的修改，此修改与 OS（操作员站）无关，仅与 ES（工程师站）有关。此时用户只需要在 RUN 模式下载变动部分即可（例如某个 Block I/O 属性修改为隐藏）
2. 修改与 OS（操作员站）相关，此时需要对 OS（操作员站）项目进行编译（例如修改一个消息文本）
3. 结构被改变，例如：添加了输入/输出，或者消息文本。这种修改会导致如下结果：
 - 连接和参数设置可能会丢失
 - 必须编译并下载完整程序，并且 CPU 必须转换为 STOP 模式
 - 如果此功能块需要被 OS（操作员站）监视及控制，则需要对 OS（操作员站）项目进行编译
 - 如果用户希望保留在控制器中的参数设置，则用户在下载之前图表之前，首先应当从控制器当中回读图表。

3.2.3. 引用程序块改变对系统的影响

如果在 CFC 图表中所引用的程序块的接口描述（程序块的输入/输出）或者系统属性发生了改变（例如用户在本项目 S7 Program(2)\Blocks 目录中的 FB1 有更改），在 CFC 中更新 FB1 的新版本之后，则 CFC 图表中所有被使用到的 FB1（假设 FB1 在多个图表中被多次引用），都将自动被更新。

改变程序块的输入/输出对引用的程序块的影响

- 添加一个输入/输出

此更改影响被引用的块，增加的输入/输出被赋予默认的系统属性，如果在块增大后，没有更

多的空间来容纳此功能块，则在视图中此功能块将与其它功能块的显示重叠在一起。

- 删除一个输入/输出

此更改影响被引用的块，被删除输入/输出将从被引用的块上删除。如果此输入/输出被赋予了连接关系，则连接关系（或者 SFC 的访问）也将被删除。此连接关系（或者 SFC 的访问）被删除的事件将被记录在系统日志当中。

- 改变输入/输出的顺序

当块的输入/输出的顺序被修改，这些输入/输出的连接关系，参数设置及属性将被保留。

- 改变输入/输出的数据类型

所有这些输入/输出的连接关系，参数设置都将丢失。这种情况也在用户删除某个输入/输出后，再重新生成此输入/输出。

- 改变输入/输出的名称

所有这些输入/输出的连接关系，参数设置都将丢失。系统无法自动生成旧名称的参照，这种情况也在用户删除某个输入/输出后，再重新生成此输入/输出。

修改程序块的系统属性对引用的程序块的影响

被引用块的属性将被自动更改。

修改程序块对其它功能块的影响

无法再执行仅仅下载修改的操作（必须下载完整程序）。

由于在控制器中的块与 CFC 图表中的程序不一致，所以 "Read Back Chart" 回读图表功能将无法使用。

修改程序块对 WinCC 的影响

- 如果块的类型被改变，则当 CFC 编译时，可能会生成新的 DB 块。为了保证正常在线访问，这些数据必须重新传送给 WinCC（操作员站）。
- 如果块的输入/输出被修改，并且用户希望此点被监视/控制（属性 S7_m_c=true），则需要遵循如下规则：
 - 如果添加一个输入/输出，则当进行 OS 编译时，此输入/输出可以被 WinCC 识别，并被使用
 - 如果删除一个输入/输出，WinCC 中的标签将不存在，与此标签的关联关系也应当被删除
 - 如果更改输入/输出的名称，则在 WinCC 中的标签名称也将改变。与此标签的关联的图片、界面等也需要调整。
- 任何影响 WinCC 的改变，都要求用户重新编译 OS（操作员站）

3.3. 编辑 S7-CFC 程序块

用户可以在图表中编辑块的属性，这些属性可以是整个块的属性，也可以是单独输入/输出的属性。

3.3.1. 设置对象属性

用户可以通过右键点击功能块，选择 Object Properties-General，查看块的属性。这些属性中比较重要的是

- **Name** 块在整个图表中的唯一的名称，此名称（最大 16 个字符）将被显示在块的头部
- **Operator C and M possible** (OCM Possible) 如果用户希望此功能块能够被 OS 监视及控制，可以选择此项，这样就可激活“ Operator Control and Monitoring”、“ Messages”按钮及“ block icon”输入框。
- **Operator Control and Monitoring** 此按钮将打开一个显示哪些输入/输出将被监视/控制的对话框，用户可以在此修改其在 WinCC 中的属性。
- **block icon** 被监视/控制的块可以在 WinCC 中以块图标方式显示。用户可以在此指定所使用的图标。
- **Messages** 此按钮将打开一个组态消息的对话框，用户可以在此编辑消息文本，此文本可以用于传送消息给 MIS/MES 管理系统。

3.3.2. 修改输入/输出的数量

如果程序块拥有可变数量，同数据类型的输入，例如 NAND, OR 等等，用户可以通过使用 "Edit >Number of I/Os..."菜单命令来更改输入的数量。下图为更改 AND 功能块输入数量的界面，默认输入数量为 2，最大为 120。

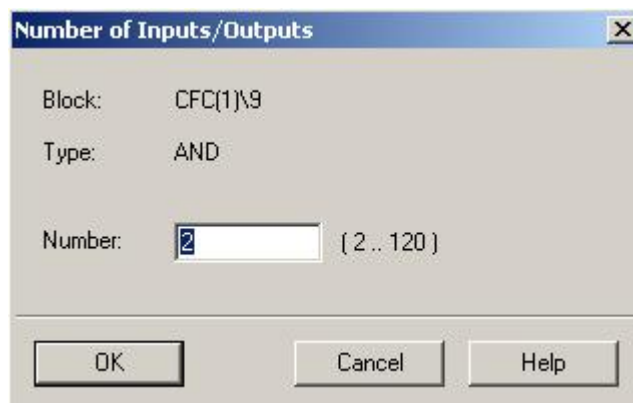


图 3-11: 更改 AND 功能块输入数量

3.4. 修改输入/输出的属性

用户可以在块的属性对话框中的输入/输出栏中，查看并修改所有输入/输出的属性，也可以在图表中双击某个输入/输出点，单独编辑其属性。

Properties - Block -- CFC(1)\3													
General		Inputs/Outputs											
#	Name	I/O	Type	Value	Comment	Invisible	Watched	Identifier	Unit	Text 0	Text 1	Parameter	Signal
1	EN	IN	BOOL	1		<input checked="" type="checkbox"/>	<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>
2	MAN_ON	IN	BOOL		Select: 1=...	<input type="checkbox"/>	<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>
3	PVPER_ON	IN	BOOL	0	Enable: 1=...	<input checked="" type="checkbox"/>	<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>
4	P_SEL	IN	BOOL	1	1=Proporti...	<input type="checkbox"/>	<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>
5	I_SEL	IN	BOOL	1	Enable: 1=...	<input type="checkbox"/>	<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>
6	INT_HOLD	IN	BOOL	0	Integral A...	<input type="checkbox"/>	<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>
7	I_ITL_ON	IN	BOOL	0	Initialize In...	<input type="checkbox"/>	<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>
8	D_SEL	IN	BOOL	1	1=Derivati...	<input type="checkbox"/>	<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>
9	SAMPLE_T	IN	REAL	1.0	Sample Ti...	<input checked="" type="checkbox"/>	<input type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>
10	SP_INT	IN	REAL		Internal se...	<input type="checkbox"/>	<input checked="" type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>
11	PV_IN	IN	REAL		Process V...	<input type="checkbox"/>	<input checked="" type="checkbox"/>					<input type="checkbox"/>	<input type="checkbox"/>

图 3-12: 块输入/输出的属性

- # 输入/输出的编号
- Name 输入/输出的名称
- Value 用户可以在此设置此变量的数值
- Invisible 如果用户不希望显示此输入/输出变量，则可选择此项，这样可以减少块的显示大小
- Watched 可以允许用户监视并修改变量值
- Identifier 可以显示非 BOOL 类型变量的标识，此标识文本将被用于在 OS 中显示
- Text0 对于 BOOL 类型变量，当其为 0 时，用于显示的文本，例如：Close
- Text1 对于 BOOL 类型变量，当其为 1 时，用于显示的文本，例如：Open
- Parameter 如果用户希望此变量作为参数，可选择此项，并可对其赋值
- Signal 如果用户希望此变量作为信号，可选择此项，其数值来自于连接关系

3.4.1. 取反一个块的输入

对于块的 BOOL 类型的输入，S7-CFC 提供了取反功能。用户可以双击某 BOOL 输入名称后，在属性中选择 Inverted 选项，在输入取反之后，“0”将变为“1”；“1”将变为“0”。

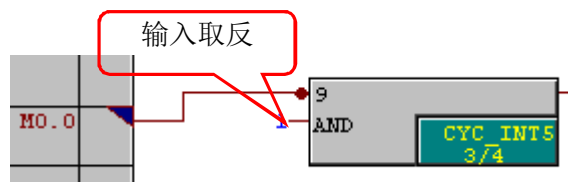


图 3-13: 取反输入

注意事项:

- 如果复制一个取反输入的信号到另一个输入，新输入信号不会自动取反。用户必须自己明确是否取反
- 如果移动一个取反输入的连接到一个非取反的输入，新输入信号不会自动取反，取反将

被取消。

- 对于图表的输入/输出是不可以取反的，用户在连接中加入一个 NOT 块来解决这一问题。

3.4.2. 使用数值标识

S7-CFC 提供了数值标识功能。当用户定义了不同数值定义的标识后，即可使用。如下图中的输入变量 IN_BOOL，用户可以定义当输入数值为“ 0” 时显示“ close”；当输入数值为“ 1” 时显示“ open”。如下图中的输入变量 IN_WORD，用户可以定义当输入数值为“ 2” 时显示“ in123456”。

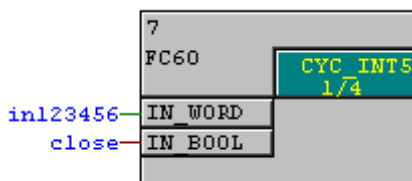


图 3-14: 用户定义标识后的显示

当用户在 LAD/STL 编程环境下生成 FB/FC 时，对于其输入参数的属性，用户可以添加数值标识。下图为 FC60 的 BOOL 类型输入参数的属性修改界面。对于 BOOL 输入类型，允许的参数范围仅包括“S7_string_0” 及 “S7_string_1”。

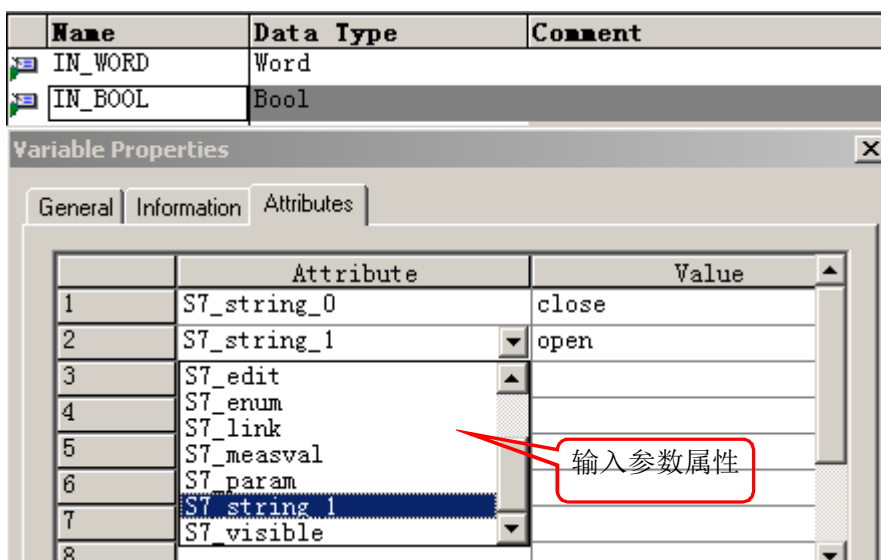


图 3-15: BOOL 输入参数属性

当用户在 LAD/STL 编程环境下生成 FB/FC 时，对于其输入参数的属性，用户可以添加数值标识。下图为 FC60 的 WORD 类型输入参数的属性修改界面。对于 BYTE, INT, DINT, WORD, DWORD 输入类型，允许的参数范围为从“S7_string_2” 至“S7_string_25”。

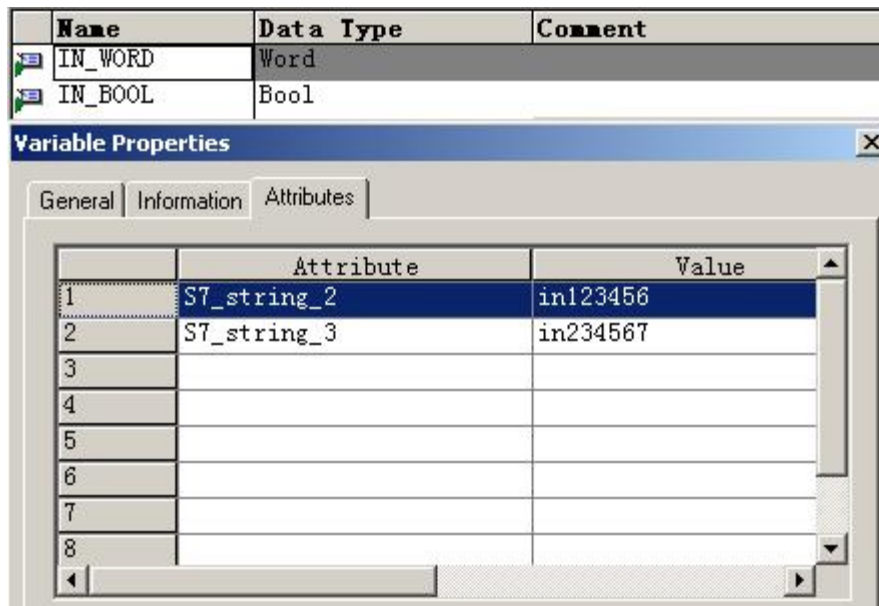


图 3-16: WORD 输入参数属性

注意事项:

- "S7_string_0" 及 "S7_string_1"可以用于 BOOL 输入类型, 可以为最大 16 个字符的文本
- "S7_string_2" 至"S7_string_25"可以用于 BYTE, INT, DINT, WORD, DWORD 输入类型, 可以为最大 8 个字符的文本
- "S7_enum" : 枚举数据类型, 主要用于 PCS7 中, 其使用的数值定义需要在 Configuring enumerations 中定义。可以用于 BOOL, BYTE, INT, DINT, WORD, DWORD 输入类型, 可以为最大 16 个字符的文本。其可以替代"S7_string_0" 至"S7_string_25"的作用, 由于其主要用于 PCS7 中, 超出了本文讨论的范围, 此处不做详细介绍。

3.4.3. 定义数据归档

具有 archiving 和 long_term archive 的变量将在编译 OS 时自动生成归档变量

- "No archiving" (S7_archive:= 'false')输入/输出不会被归档
- "Archiving" (S7_archive:= 'shortterm') 在 OS 上短期归档
- "Long-term archiving" (S7_archive:= 'longterm') 在归档服务器长期归档, 并且存储在 CD, DVD, MOD, 磁带等等。

所有具备 OCM (可以被监视及控制) 特性的数据 (数据类型为 BOOL, BYTE, WORD, DWORD, INT, DINT 及 REAL) 都可以被归档。

3.5. Interconnections (连接关系)

在一个 CFC 的 Chart 中, 一个 Interconnections (连接关系) 可以连接:

- 一个 block/chart 的输出与相同的或其它的 block/chart 的一个或多个输入, 输入与输出的数据类型必须一致。

- 一个 block 的输出与一个 run-time group（仅 BOOL 数据类型）
- 一个 chart 内的 block 的输入/输出与 chart 的输入/输出
- 一个 block 的输出与 CFC 外部的数据，例如共享地址

用户可以在 Chart 中，左键点击某个块的输入/输出管脚，并脱拽鼠标，在需要连接的其它管脚上方松开鼠标左键，即完成了新建连接关系的操作。

另外，用户也可以通过右键点击某个 block 的输入/输出时，组态与其它对象的连接关系，如下图所示（注意：点击输入或输出后，显示的内容有所不同）。

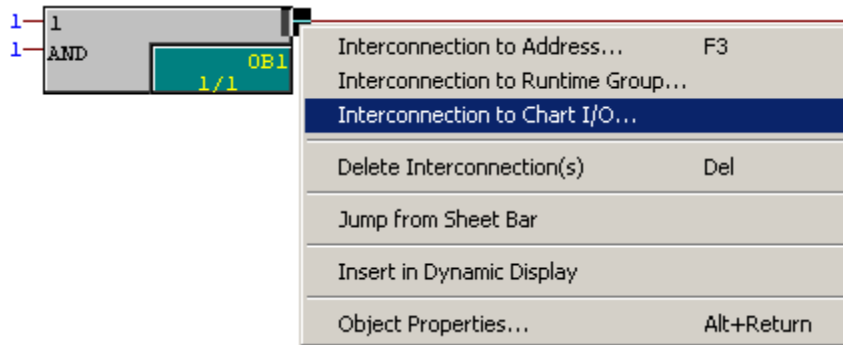


图 3-17: 组态连接关系

3.5.1. 与共享地址的连接

当用户在连接组态界面下选择了“ Interconnection to Address...”后，即可以组态共享地址连接，例如：共享数据块，I/O 信号，存储位，定时器，计数器。用户可以从下拉列表中选择地址，也可以手动输入地址，地址的数据类型与 block 的输入/输出的数据类型要一致。

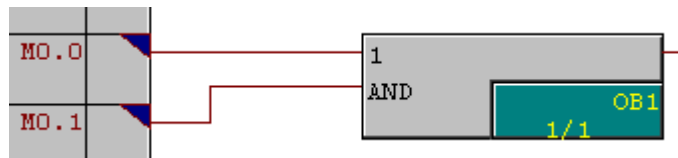


图 3-18: 组态共享地址连接

3.5.2. 与 Run-Time Groups 的连接

用户可以动态地使能或禁止。用户可以将一个 Block 的输出连接到一个 Run-Time Group，这样就可以控制其是否执行。在这种情况下，Run-Time Group 属性“ Active” 的设置将不被考虑。

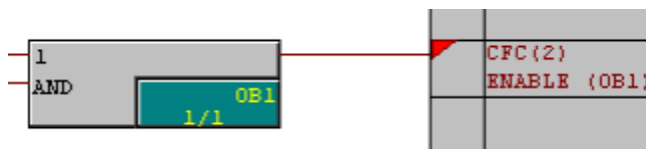


图 3-19: 组态 Run-Time Groups 连接

3.5.3. 文本连接

用户可以对一个 block/chart 的输入组态文本连接，此连接指向 CFC 中的另外一个

block/chart 的输出。如果用户输入的连接关系不正确（例如连接目标不存在，类型不匹配）则此连接处于“打开”的状态，此连接只有在闭合状态下才变为一个真正的连接。

当用户可以右键点击某个块的输入管脚，并选择“Textual Interconnection...”后，即可以组态文本连接。其格式为：(chart\block. I/O)，如果连接关系正确，CFC 自动更改为标准格式。例如下图中的输入的文本为 CFC (1)\1. ERR，此连接关系实际为 CFC (1) (A1)\1. ERR（即 CFC(1) 的 Sheet A，Sheet A 中 6 个区域中的区域 1，块编号为 1，输出管脚标识为 ERR）。

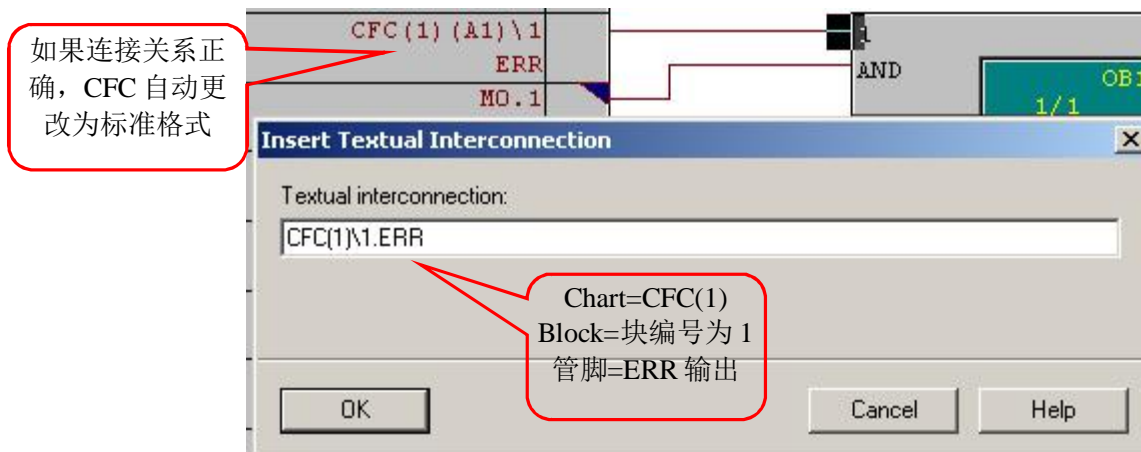


图 3-20: 组态文本连接

如果用户输入了一个错误的文本连接，则系统使用黄色的三角标志显示，直到用户更改正确为止。

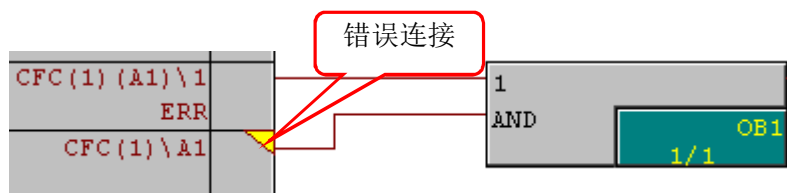


图 3-21: 错误文本连接

3.5.4. 组态数据结构之间的连接

通过使用数据结构，用户可以将任意数据打包为一个结构体，并将其定义为一个 block 的输入/输出。在 Block 的显示部分，一个结构拥有一个输入/输出名称，数据类型为“ST”。

一个结构可以包括多个元素（最大嵌套深度为 8 层）：

- 基本数据类型（BOOL, WORD...）
- 结构

一个结构体包括如下信息：

- 数据类型（基本数据类型，否则为“ST”）
- 名称
- 数值（仅仅与基本数据类型相关）

连接

一个 block 结构类型的输入/输出仅仅可以与另一个结构体相连接，而不可以与基本数据类型连接。连接的结构体必须要一致，包括结构体内的基本数据类型，顺序，名称，都要一致，但结构体的名称可以不同。结构体内的基本元素不可以单独与另一个结构体内的基本元素进行连接，结构体只可以作为整体与其它结构体相连接。

与共享地址的连接

一个 block 结构类型的输入/输出不可以与一个完整数据块进行连接 (DBx)，其可以与数据块中的第一层数据结构相连接 (DBx.name_st)。与之相类似，对于基本数据类型，其只可以与数据块中的第一层基本元素相连接 (DBx.element)。

3.6. 运行时间属性

块的运行时间属性决定了此块在整个 CPU 结构中的运行顺序。此属性决定了一个目标系统的响应时间，死区时间，以及与时间相关系统的稳定性。当每个 block 被插入时，系统都将赋予其默认的运行时间属性。

3.6.1. 块的运行时间属性

块的运行时间属性以彩色背景显示

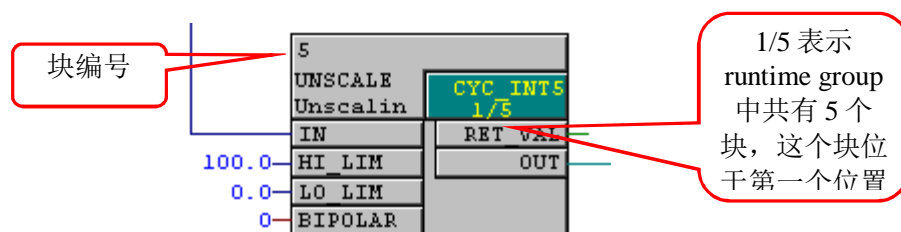


图 3-22: 块的运行时间属性

3.6.2. CPU 的运行时间属性

用户可以通过菜单“ Edit>Run Sequence...”切换至 run-time editor 。在此界面下，用户可以查看 CPU 的运行时间属性。此界面下分为三层结构：任务/runtime group/block。用户可以展开/收起这些分层目录来查看整个项目的运行时间属性。也可以选择某个 runtime group/block 后，使用左键托拽的方式来改变其所在 任务/runtime group 中的顺序。

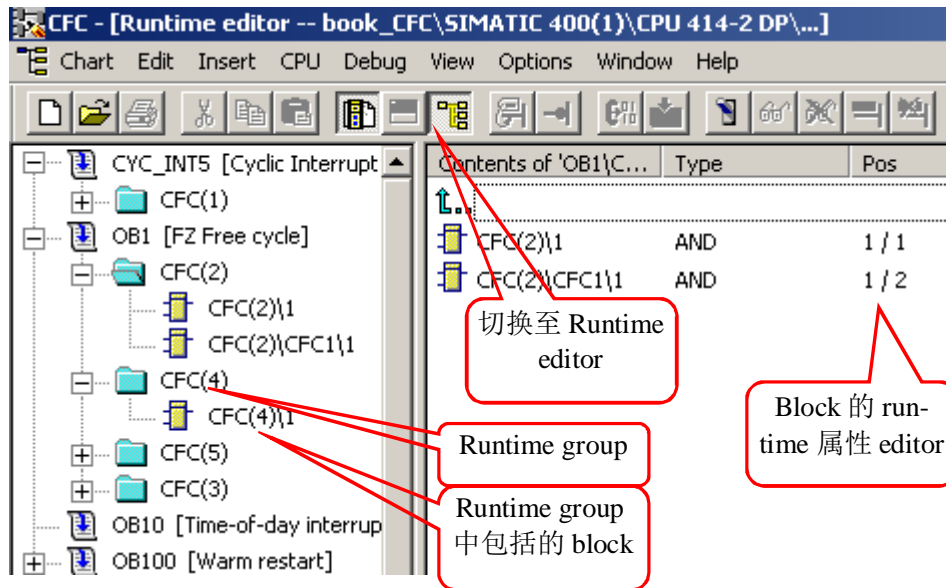


图 3-23: CPU 的运行时间属性

3.6.3. 生成 Runtime Group

当用户使用右键点击某个任务后，选择“ Insert Runtime Group...”后，可以添加新的 Runtime Group，另外，当用户在 S7program/charts 目录下插入一个新的 CFC 程序时，系统将默认在 OB35 中生成一个新的 Runtime Group。如果用户不更改 chart 的插入点，用户所新添加的 chart 都将位于 OB35 当前的 Runtime Group 中。如果用户希望在其它任务中插入新的 chart，可以选择此任务（例如 OB1），并选择“ Predecessor for the Next Chart Insertion...”，则当用户在 S7program/charts 目录下插入一个新的 CFC 程序时，此 chart 将被添加到当前任务位置例如（OB1）。

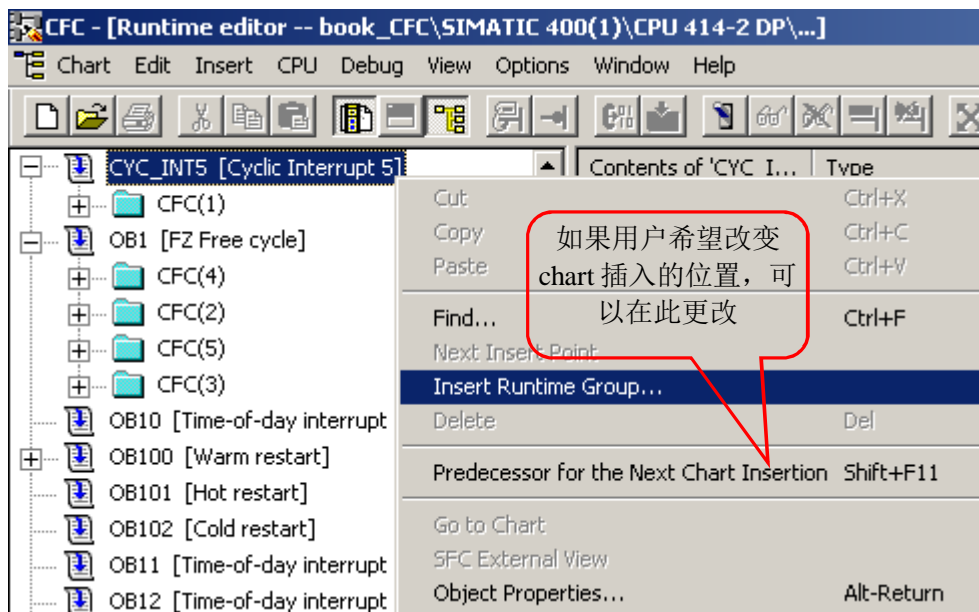


图 3-24: 添加 Runtime Group 及安装点

3.6.4. Runtime Group 的属性

当用户使用右键点击某个 runtime group 后，选择其属性，可以看到如下界面，其中包括如下重要属性：

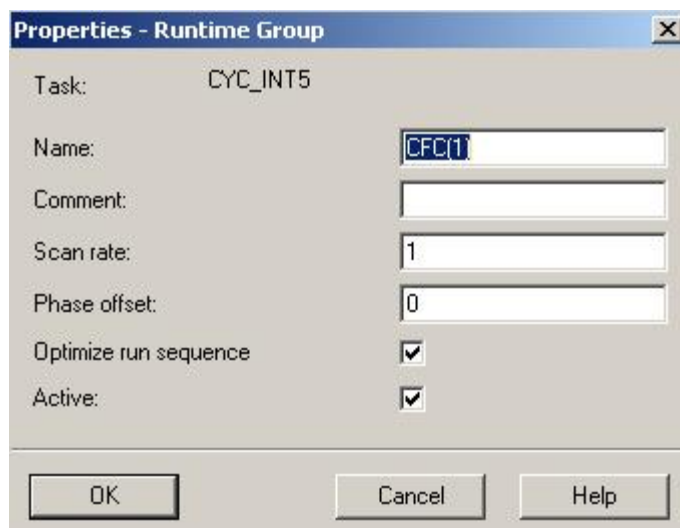


图 3-25: runtime group 运行时间属性

- Scan rate: 决定当任务被执行时，runtime group 是否在每个任务执行周期，还是间隔多个任务周期被执行。数值范围：共 16 个数值可供选择， 2^0 ， 2^1 ， 2^2 ，... 2^{15} 。例如：OB33 的执行周期为 500 ms，则可能的 scan rate 为：1s, 2s, 4s, 8s, 16s。
- Phase offset: 数值范围：大于等于 0，小于所选择的 Scan rate。例如：OB33 的执行周期为 500 ms，Scan rate 为 16, Phase offset 为 3，则此 runtime group 执行的时间为 1.5s; 9.5s; 17.5s 等等。
- Active: 用户可以决定是否激活此 runtime group

3.7. 编译

用户可以将 chart 编译为一个程序或者一个 FB。默认情况下，chart 将被编译为一个程序。CFC 使用的编译器为 SCL。

3.7.1. 将 chart 编译为程序

用户可以通过菜单“Chart > Compile > Charts as Program...”进行编译，或者直接点击工具栏中的编译图标来编译程序。

当编译时，CFC 程序将被镜像为 FC 及 DB。

需要生成 FC 的情况：

- 每个使用的 OB 需要一个 FC
- 每个 Runtime Group 需要一个 FC

需要生成 DB 的情况：

- 需要为每个 FB 的背景数据块生成一个 DB
- 为了保存每个 FC 的中间结果，需要生成 CFC 内部 DB, 每种数据类型需要一个 DB。当其达到最大长度时（4 Kbytes），在添加新的 DB

注意：在用户在编译界面中如果没有选择“ Make Textual Interconnections”，这些连接不会转换为真正的连接。这些相关的参数将使用默认值。

3.7.2. 将 chart 编译为 FB

用户可以通过菜单“ Chart > Compile > Charts as Block Type...” 将 chart 编译为一个 FB。

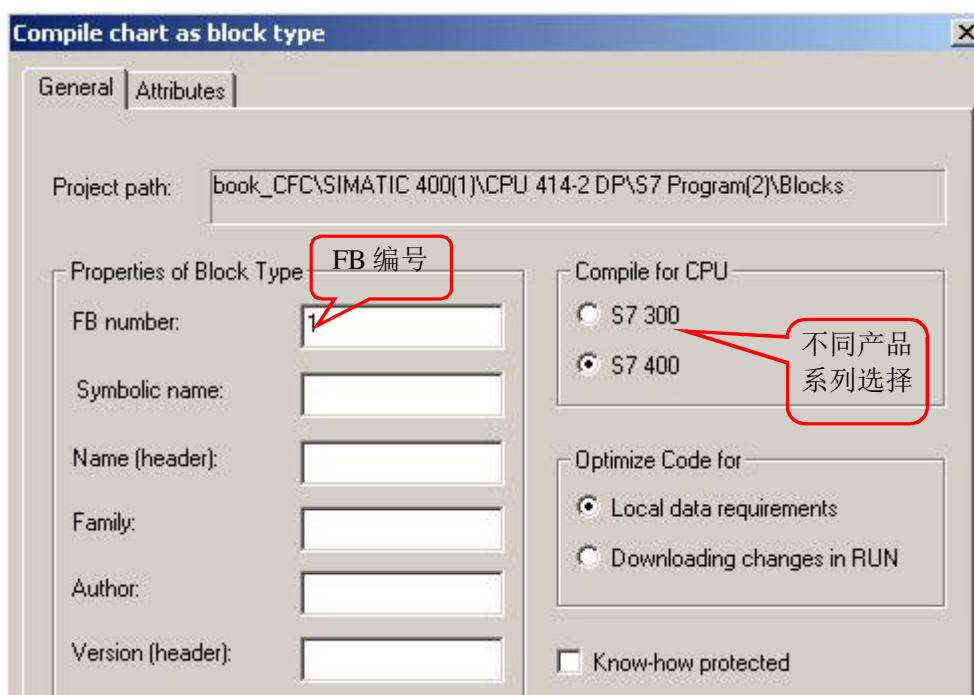


图 3-26: runtime group 运行时间属性

Compile for CPU

由于 S7 300 / S7 400 拥有不同的启动 OB，所以此处的选择将会使编译生成的 FB 代码不同。

Optimize code for (优化方式)

- Local data requirements: (本地堆栈要求)

当用户修改 chart 时，这种代码优化方式不增加本地数据堆栈的需求。所有的临时变量都将被存储于背景数据块（VAR 区域）。然而当用户更改 chart 时，这种方式导致了背景数据块时间标签的改变，因此，用户无法实现在线模式下下载。

- Download changes in RUN

当用户修改 chart 时，这种代码优化方式将把临时变量都存储于 VAR_TEMP 区域中。而 VAR_TEMP 将使用 CPU 的本地数据堆栈。这样当用户更改 chart 时，不会导致背景数据块时间

标签的改变。因此，用户在多数情况下可以实现在线模式下下载，但此种方法的缺点是增加了本地数据堆栈的需求。

Know-how protection

当用户使能此项时，仅仅当 SCL 源文件存在时，生成的块可以被察看及修改。

3.8. 下载用户程序至目标系统

对于 CFC 生成的程序，由于 CFC 可以保证 AS(控制器)中的数据一致性，所以用户必须使用 CFC 编辑器下载程序至目标系统。对于复制/下载离线 Blocks 文件夹中的 blocks 至在线文件夹的操作，是不允许的。如果用户程序还没有进行过编译，系统将提示用户进行编译操作。

3.8.1. 下载完整程序 (Entire program)

用户可以在 CPU 处于 STOP 或 RUN-P 模式下，进行下载完整程序 (Entire program) 操作。在用户确定提示对话框后，CPU 中的所有块将被清除，并切换到 STOP 模式。在下载成功后，系统将提示用户是否启动 CPU。

3.8.2. 仅下载改变 (Changes only)

用户可以在 CPU 处于 RUN-P 模式下，进行仅下载改变 (Changes only) 操作。为了防止 CPU 进入 STOP 状态，系统将进行一系列检查。但无法确保 CPU 肯定不会切换到 STOP 模式。当项目中存在时间戳不一致的情况下，无法进行 Change only 下载。

注意事项:

- 如下情况用户可能执行仅下载改变操作：
 - 用户至少执行过一次完整下载
 - 下载已经下载过的程序结构
- 在CFC的“ Options>Customize>Compile/Download...” 界面中，提供了“ Compress” 选项，如果用户选择了此选项，则与前面的CPU压缩空间不同，每次编译时，CFC将压缩FC/DB的编号范围，此时就无法在进行仅下载改变 (Changes only) 操作。
- 下载完整程序并不意味着要编译完整程序，如果程序已经编译并被下载到了CPU中，用户可以进行仅下载改变 (Changes only) 操作
- 用户可以随意编译程序 (changes only or entire program)，此操作不影响用户是否可以仅下载改变 (Changes only) 的操作。
- 如果一个仅下载改变 (Changes only) 的操作失败，用户可以重新尝试下载剩余部分
- 如果用户改变了程序，并且在其它系统或PLCSIM上进行测试，随后则用户不可以在原CPU上进行仅下载改变 (Changes only) 的操作。

3.8.3. 防止 CPU 进入 STOP 状态

在编译及下载的操作时，系统将进行一系列一致性检查来防止CPU在下载时进入STOP状态。

编译

- 进行包括块层次调用及时间标签的一致性检查，
- 检查需要的最大本地数据堆栈并与所组态CPU所拥有的资源相对比，如果超出用户在（Options > Customize > Compilation）中设定的数值或CPU资源的100%，则产生一个警告。
- 检查嵌套深度，如果超出，则停止编译并报警。
- 检查OB，确保其调用的FC已经被CFC生成，否则将报警。
- 在编译之后，检查S7通信的背景数据块，并与CPU允许的最大通讯任务数量来对比，如果超出用户在（Options > Customize > Compilation）中设定的数值或CPU资源的100%，则产生一个警告。

所有的警告消息将中止后续的下载操作。

下载

- 进行包括块层次调用及时间标签的一致性检查，
- 检查需要的最大本地数据堆栈并与在线CPU所拥有的资源相对比，如果超出用户在（Options > Customize > Compilation）中设定的数值则，则产生一个警告。如果超出CPU资源的100%，则取消下载。
- 检查需要的存储空间。如果超出CPU资源的100%，则取消下载。如果超出用户在（Options > Customize > Compilation）中设定的数值则，则产生一个警告，并提示三种操作选项：
 - 压缩CPU内存 compress the CPU memory
 - 跳过压缩并继续下载（责任用户自负）skip compression and continue the download (responsibility of the user)
 - 取消下载cancel the download
- 如果超出CPU资源的100%，则取消下载。

类型导入（Type import）

当一个被调用的块的时间标签显示其拥有不同版本时，系统将弹出此块调用层次的对话框，用户可以在此更新此块。当此块更新后，调用此块的其它块可能将无法执行。

例如：FB61 调用 FB46，当这两个块都重新导入更新后，调用 FB46 的 FB202 如果没有被更新，FB202 可能会访问修改过的参数。因此用户可以通过菜单“ Options > Chart Reference Data... > Block Call Hierarchy”，来查找相应块调用的层次，并导入相应的块。

复制/移动（Copying/moving）

当用户复制/移动多重背景的功能块时，需要同时复制所调用的块。

选项：包括用户数据块 (Include user data blocks)

当用户下载时，如果此选项被选择，CFC 区域外的数据块将被按照如下规则处理：

- 如果是新添加或时间标签不同的数据块，则它们将被下载
- 如果在 S7 program 文件夹下不存在，则将删除 CPU 中的数据块

当用户下载时，如果此选项未被选择，则 CFC 区域外的数据块将被按照如下规则处理：

- 如果数据块在 S7 program 文件夹下存在，但在 CPU 中不存在，下载将被终止，并报警。
- 如果在 S7 program 文件夹下不存在，但 CPU 中存在数据块，则系统报警；如果数据块的时间标签不一致，系统也将报警。用户可以自行修改相关数据块。

3.9. 回读 Chart

3.9.1. 修改过的数据 Modified Data

当激活的 chart 中的参数被 OS 或操作面板修改后，用户可以通过回读 (read back) CFC charts 的方式来保存这些参数数值。

3.9.2. 回读 Reading Back

用户可以通过菜单“Chart > Read Back...”调用回读功能，如果用户选择“Program on the CPU”作为数据源，则 CPU 中当前的块及参数被回读到 chart 文件夹中。

另外用户也可以通过 SIMATIC Manager 的“Upload”命令，事先上载 CPU 中的数据，并存储到离线文件夹中。此时，用户 CFC 的回读界面中就可以选择“Program Offline”作为数据源。

以上两种情况下，用户都可以选择是否回读所有块的所有输入参数，或者仅仅会读具备系统属性“operator control and monitoring (S7_m_c:='true')”的参数，或者具有回读属性的参数(S7_read_back := 'true')。

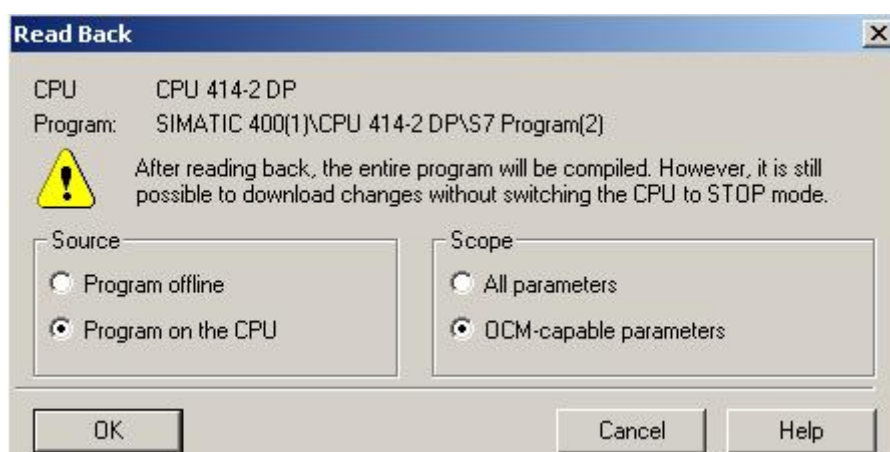


图 3-27: 回读 Reading Back

3.9.3. 回读之后

为了保证一致性，在回读操作后，整个程序将被自动重新编译。这并不意味着完整下载程序，

用户可以进行仅下载改变操作。在回读操作之后，用户可以通过菜单“ Options > Logs > Read Back”，来查看那些 I/O 被回读，以及什么地方发生了错误。

注意事项：

- 仅仅 Inout, input 参数可以被回读 (output 参数不可以)
- 当回读时，参数类型 ANY, ARRAY, POINTER, UDT 被忽略
- 对于 SFC 有些操作的 Input 参数不会被回读
- 如果 CPU 中的数值被更改，并且被回读，这些修改的数值将被标志为已经下载，当用户下载改变至 CPU 时，这些数值不包括在内。
- 回读可以读取完整块（例如 BATCH 块），这种情况下系统属性 S7_read_back := 'false'
- 对于 H CPUs，如果 H CPU 处于 solo 模式，将出现选择对话框，用户可以选择读取的 CPU，如果 CPU 处于冗余模式，此对话框不出现。
- 当进行硬件下载操作时，处于工作内存的参数将会被装载内存中的数据所覆盖。如果用户在下载硬件组态之前进行了回读操作，则可以使用完整下载的方式来解决。
- 如果选择回读“ OCM-capable parameters”，FC 的 input 将被忽略。
- 如果选择回读“ All parameters”，与 chart I/Os 有连接的 FC 的 input 将被回读。

3.10. 测试及调试

CFC 编辑器提供了监控功能，激活的 chart 中的输入/输出在测试模式下可以被监控，默认更新周期为 2 秒。用户可以通过菜单“ Debug > Test Settings...” 来修改此数值。

3.10.1. 测试模式 Test Modes

测试模式包括两种方式：

- Process Mode: 用于块的在线动态显示的通信负荷受到限制，仅仅导致 CP 或总线上存在有限的通信负荷。当测试模式被激活后，所有的块都处于“ watch off” 状态，如果用户希望监控某个块，或者输入/输出，可以单独选择此对象后，然后通过菜单“ Debug > watch on” 来使能监控功能。
- Laboratory Mode: 对比于 Process Mode，此方式通信负荷不受限制，当测试模式被激活后，所有的块都处于“ watch on” 状态。

当用户选择具体测试模式之后，可以通过菜单“ Debug > Test Mode” 来激活测试模式

3.10.2. 监控块的输入/输出

用户可以通过点击工具栏中的图标来决定是否监控 chart 中的某个对象。当数值被监控时，有三种显示情况：

- 黄色背景的黑色星号：数值改变为动态显示
- 黄色背景的黑色数值：数值读取于 CPU

- 红色背景的####: 数值无法显示

当用户使能测试模式后，即可更改未连接的输入信号的数值。可以双击某个输入管脚后，进行修改，界面可参考下图：

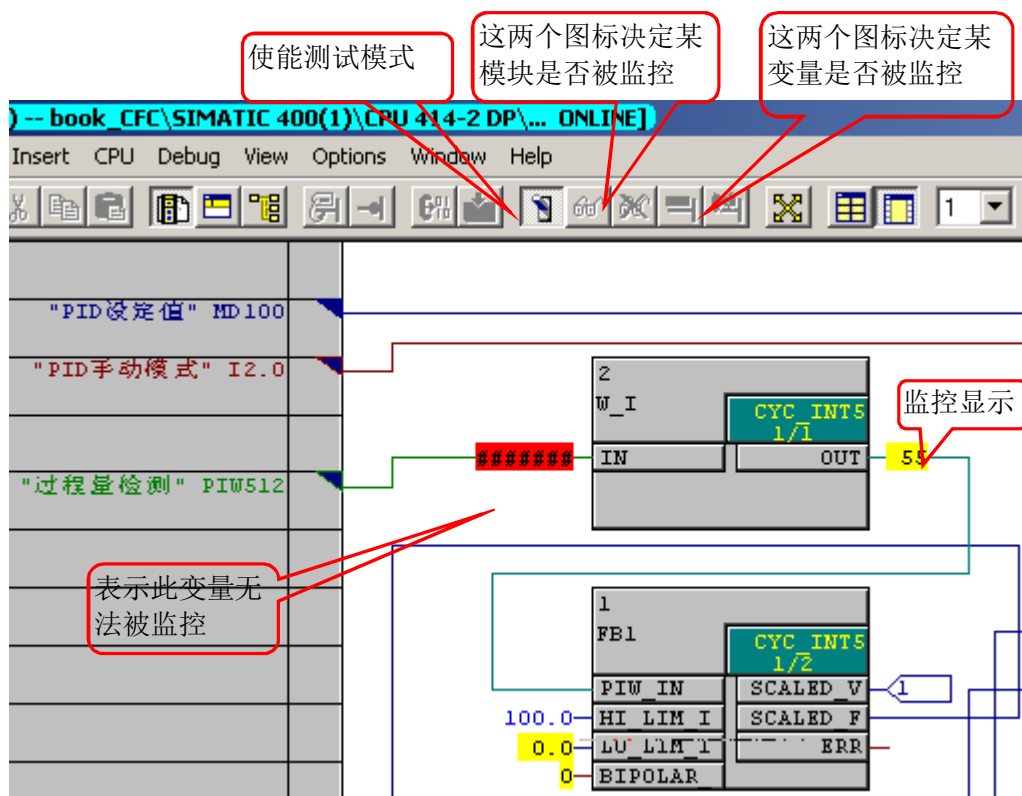


图 3-28: 监控界面

注意事项:

用户无法监视如下信息:

- 未存储在DB中的block 的输入/输出
- 未连接的FC或BOP（基本操作，例如AND，OR）等的输入，
- 数据类型为STRING, DATE_AND_TIME, ANY的输出

在测试模式下，如果功能块（FB）的EN=0，则功能块连接的输入不显示，其显示当功能块最后一次被调用时的数值；当功能块的EN由 0 变为 1 时，这些数值将被刷新。而对于FC及BOP的输入，一直显示所连接的数据源的数值。

3.10.3. 动态显示（Dynamic Display）

在测试模式下，block 及 chart 的输入/输出可以在独立窗口中被动态显示。可支持基本数据类型（BOOL, WORD 等）及结构中的基本数据类型。用户通过菜单：“View -> Dynamic Display”中可以打开此窗口。

Status	Watch	Chart	Block	I/O	Value	Unit	Comment
	<input checked="" type="checkbox"/>	CFC(1)	1	SCALED_VAL	0.198929		
	<input checked="" type="checkbox"/>	CFC(1)	3	P_SEL	1		1=Proportional Act...
	<input type="checkbox"/>	CFC(1)	3	INT_HOLD			Integral Action Hold
	<input checked="" type="checkbox"/>	CFC(1)	3	I_SEL	1		Enable: 1=Integral...

在不删除变量的情况下，是否显示此变量

变量的数值

图 3-29: 动态显示窗口

当用户希望监视某个变量时，可以通过右键点击某变量，选择“Insert in Dynamic Display”，则此变量将被添加到动态监视窗口，同时动态监视窗口也将被打开。当此窗口被关闭时，所显示的变量也将被记录，当下次打开此窗口时，这些变量还将被显示。

4. CFC 应用于虚拟工程

4.1. 虚拟工程工艺要求

下面将以一个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：

- ✧ 采集模拟量输入数据
- ✧ 输入滤波选择
- ✧ PID 控制（手动/自动切换控制）
- ✧ 模拟量输出

4.2. 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

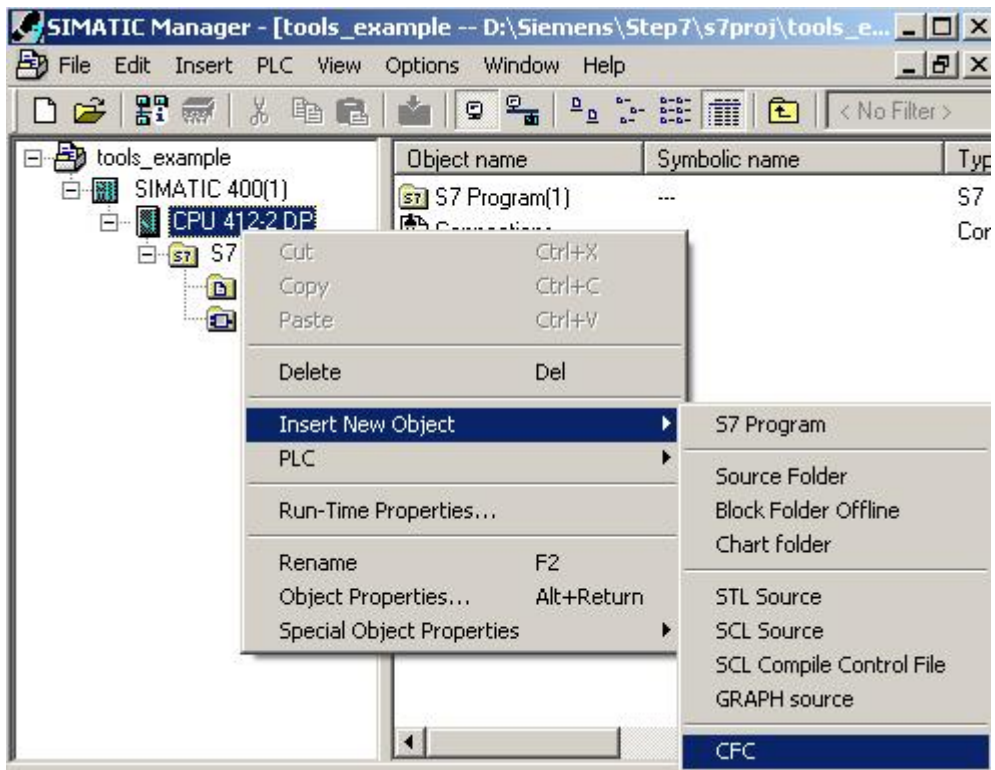


图 4-1: 添加 CFC 程序

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

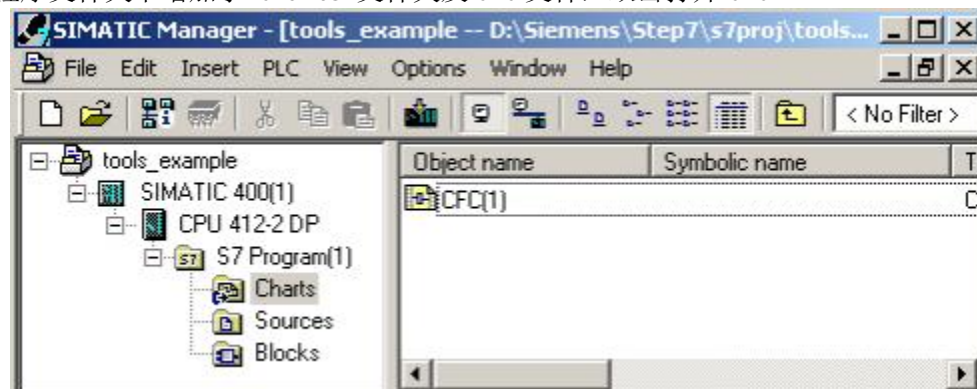


图 4-2: 打开 CFC

3) CFC 页面视图

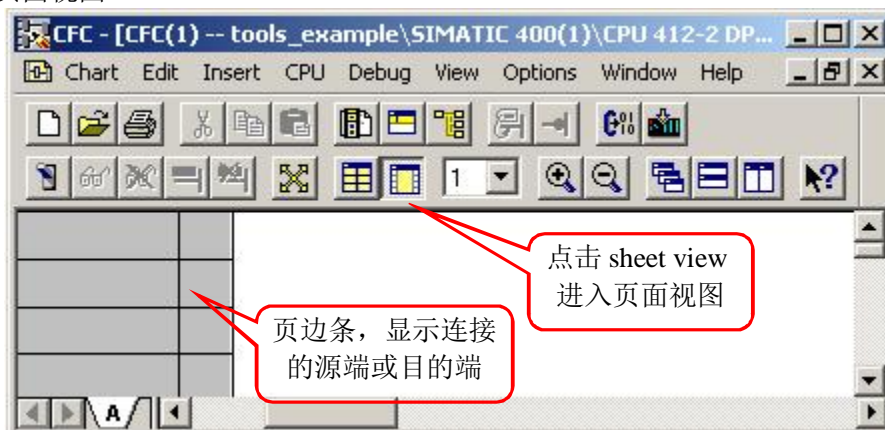


图 4-3: CFC 页面视图

4) CFC 当前所处位置

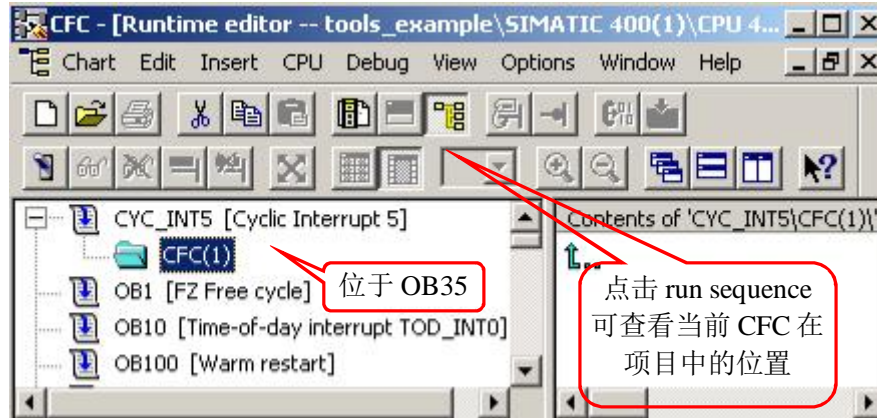


图 4-4: CFC 当前所处位置

5) 调用其它功能块

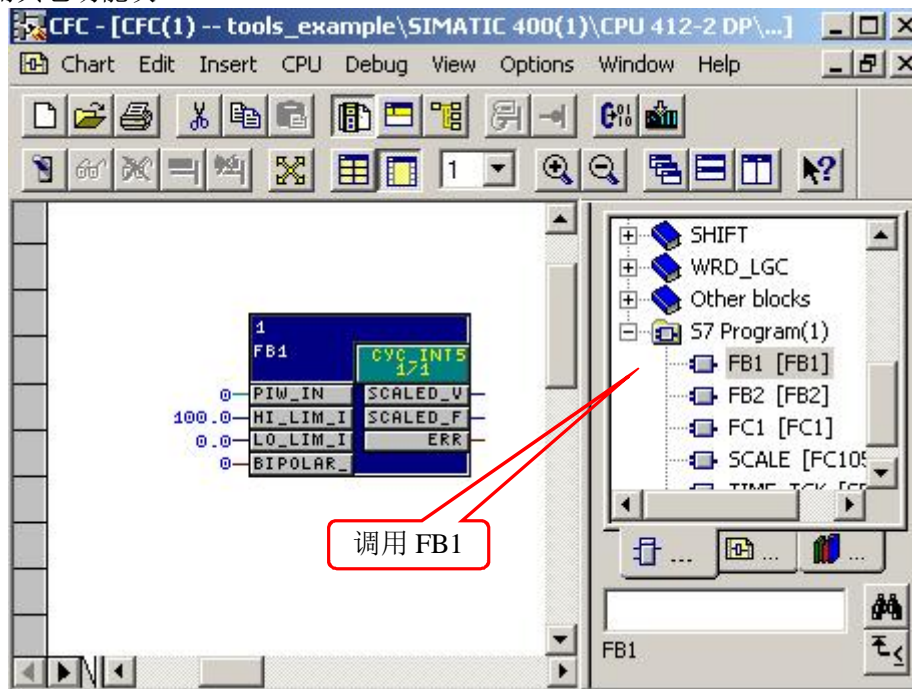


图 4-5: CFC 调用其它功能块

6) 调用分配地址连接

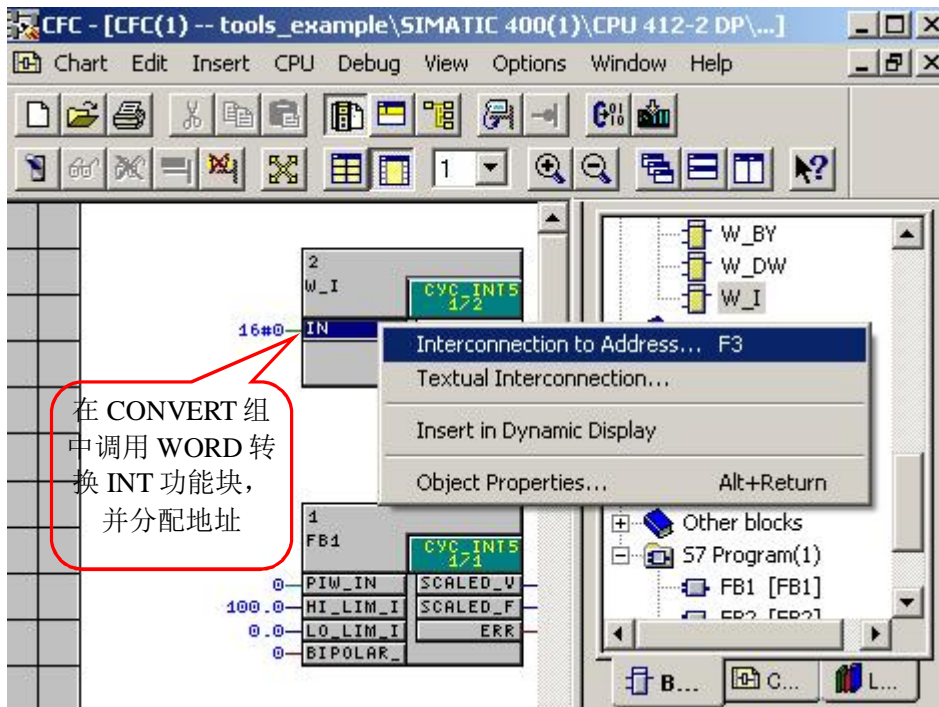


图 4-6: 分配地址连接

7) 内部连接只要数据类型一致, 直接拖拽就可建立连接

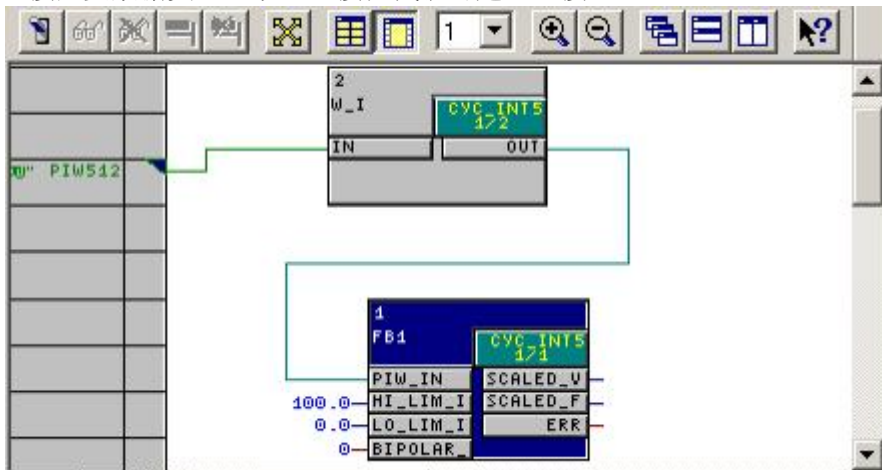


图 4-7: 内部连接

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

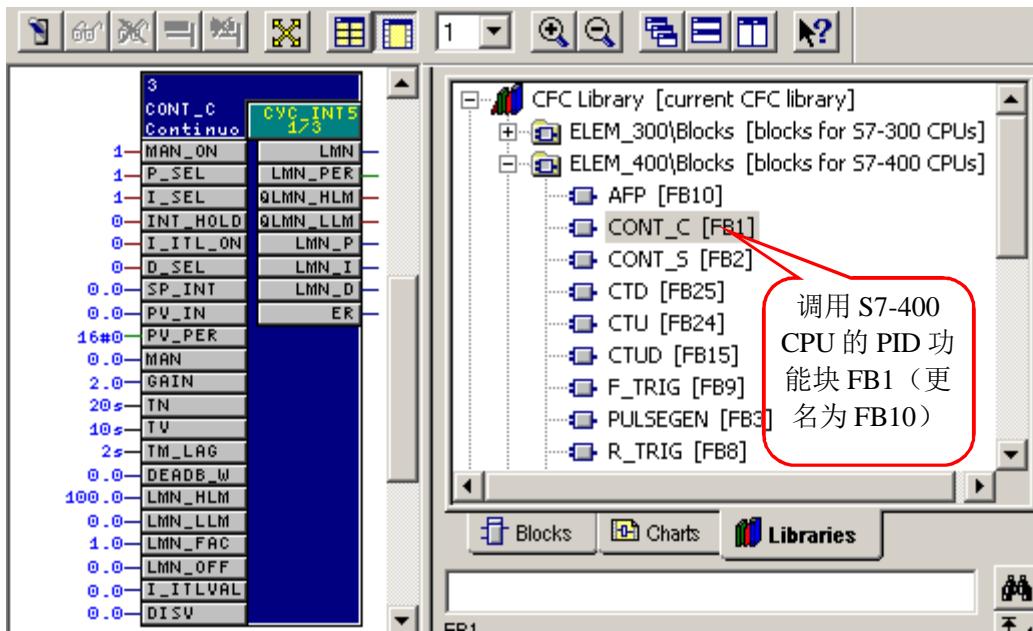


图 4-8: 调用 CFC 程序库

- 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。

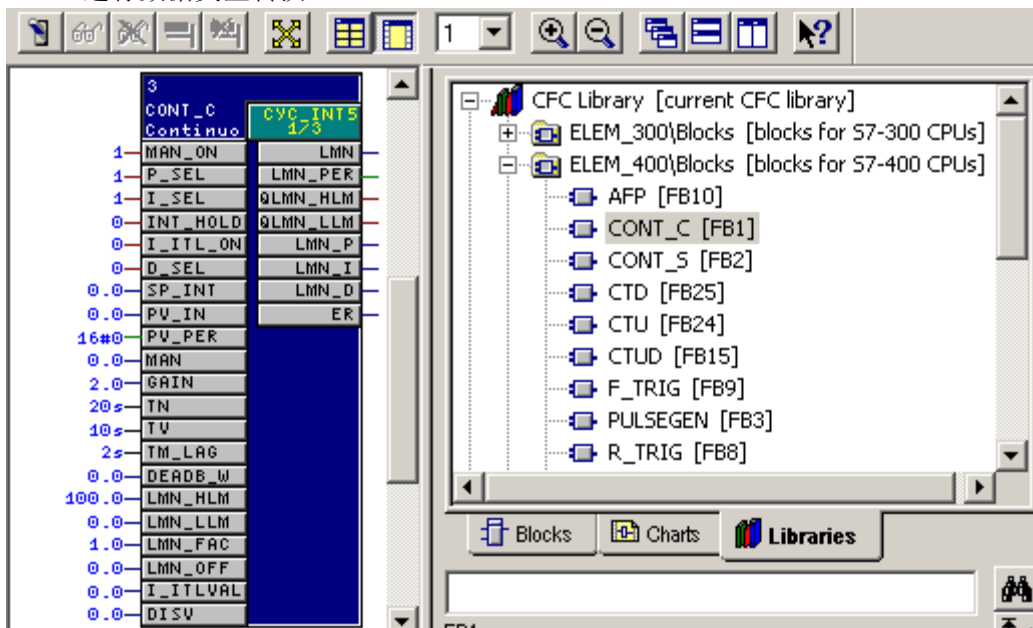


图 4-9: 调用 STEP7 标准程序库

10) 模拟量输入滤波

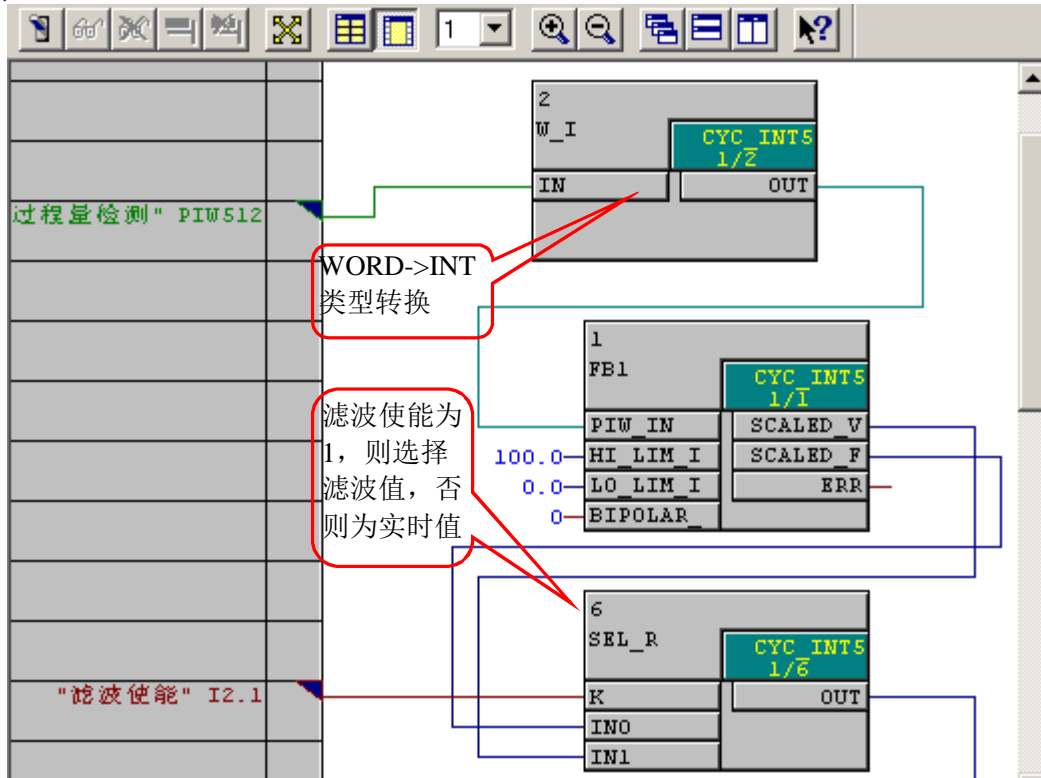


图 4-10: 模拟量输入滤波

11) 手动/自动切换无扰动处理, 将 LMN 值送至 MAN

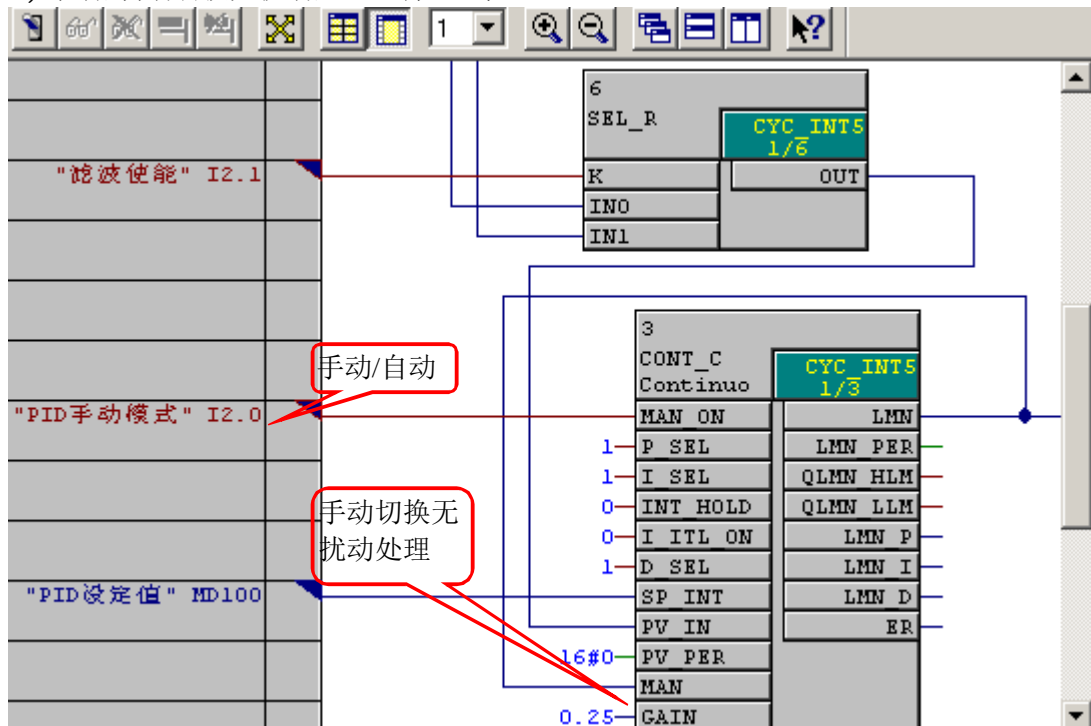


图 4-11: 手动/自动切换无扰动处理

12) 输出处理

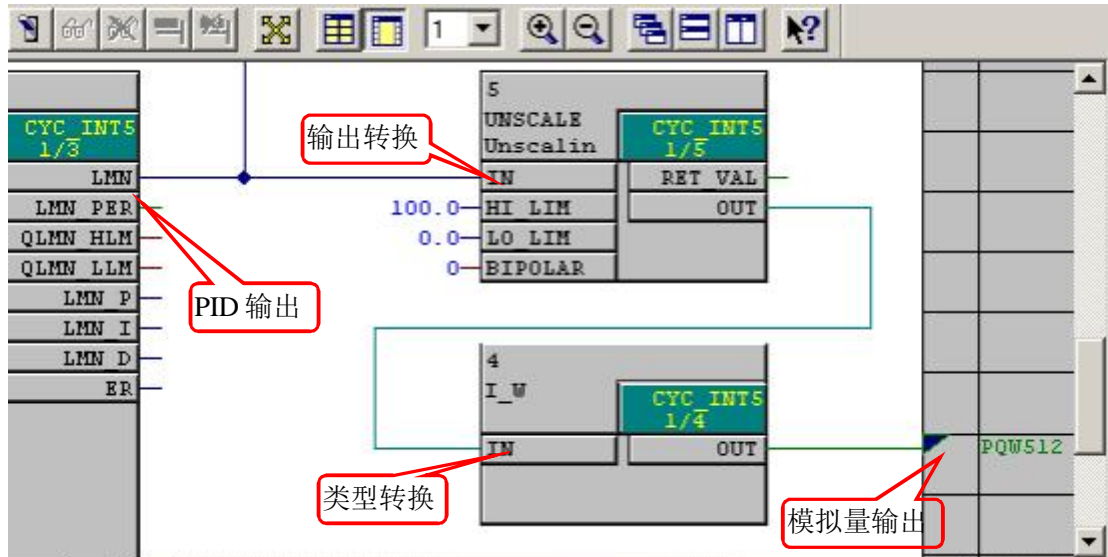


图 4-12: 输出处理

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

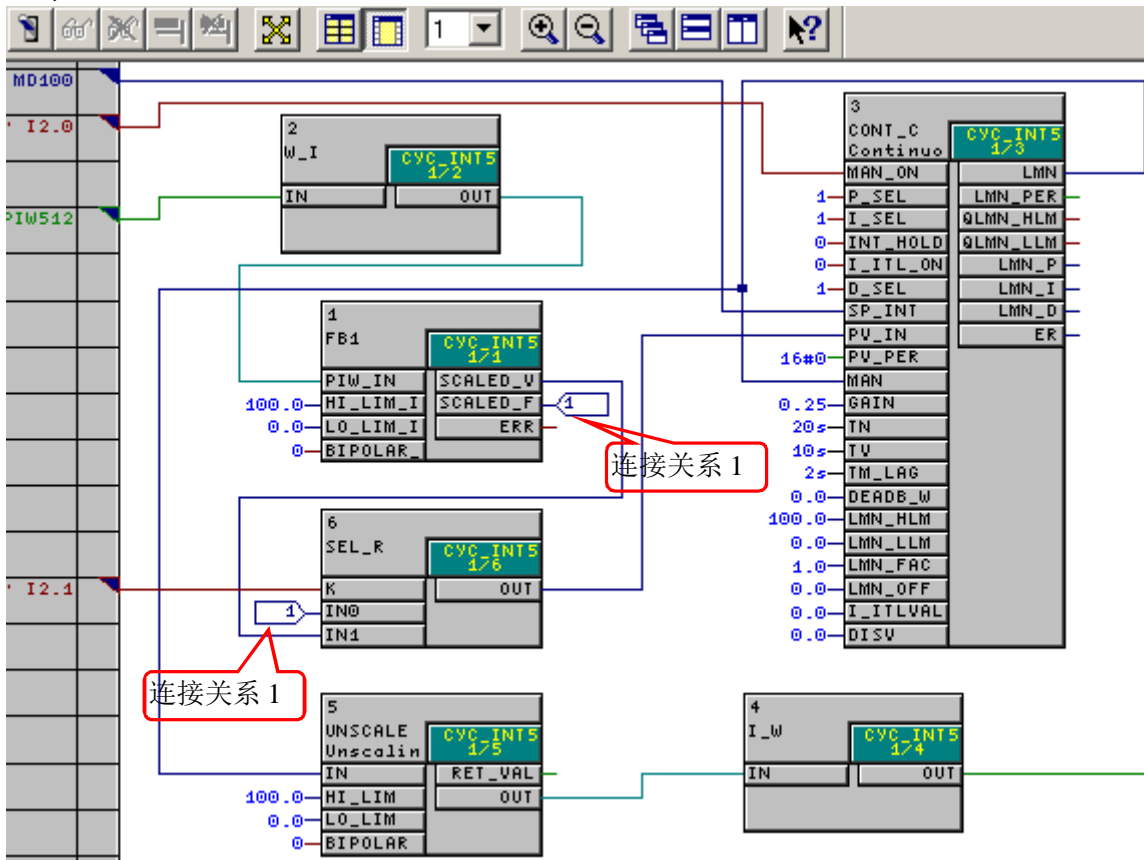


图 4-13: 程序图概览

- 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。

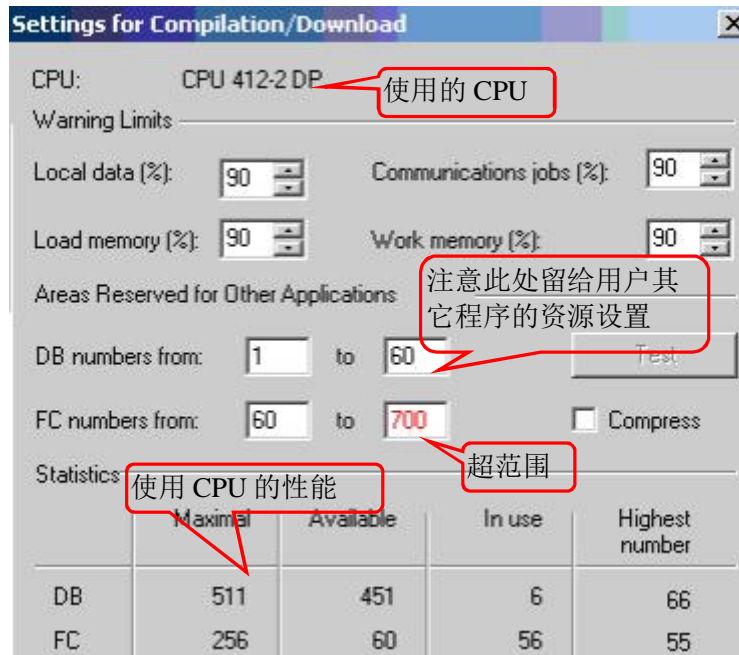


图 4-14: 编译/下载设置

- 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。

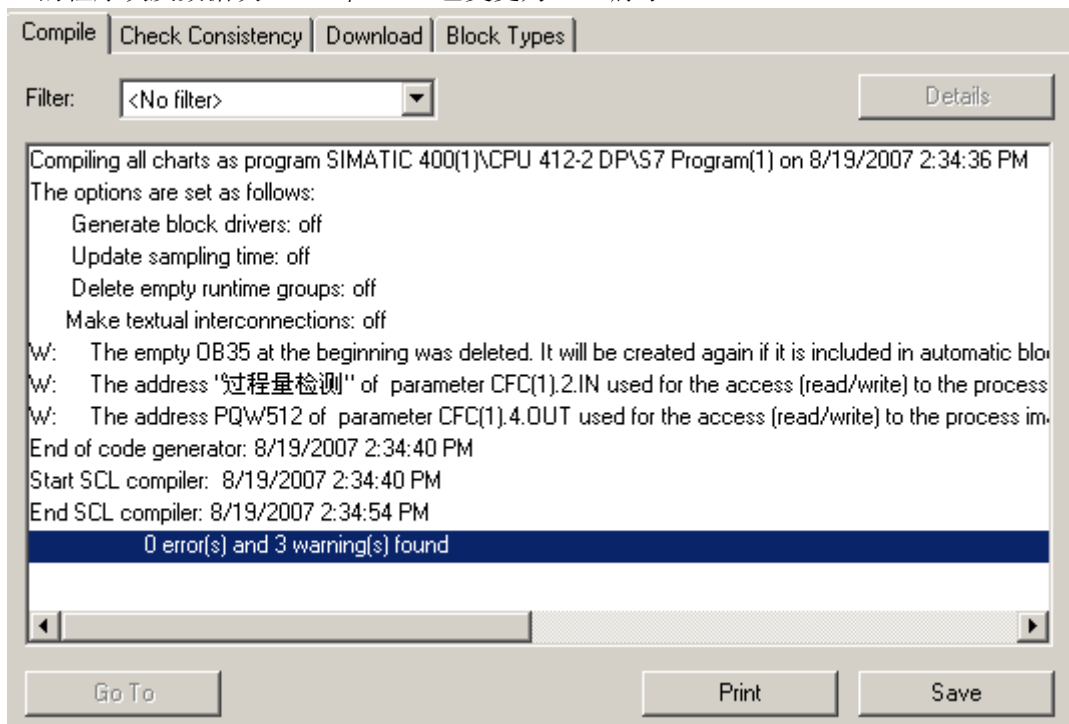


图 4-15: 编译程序

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

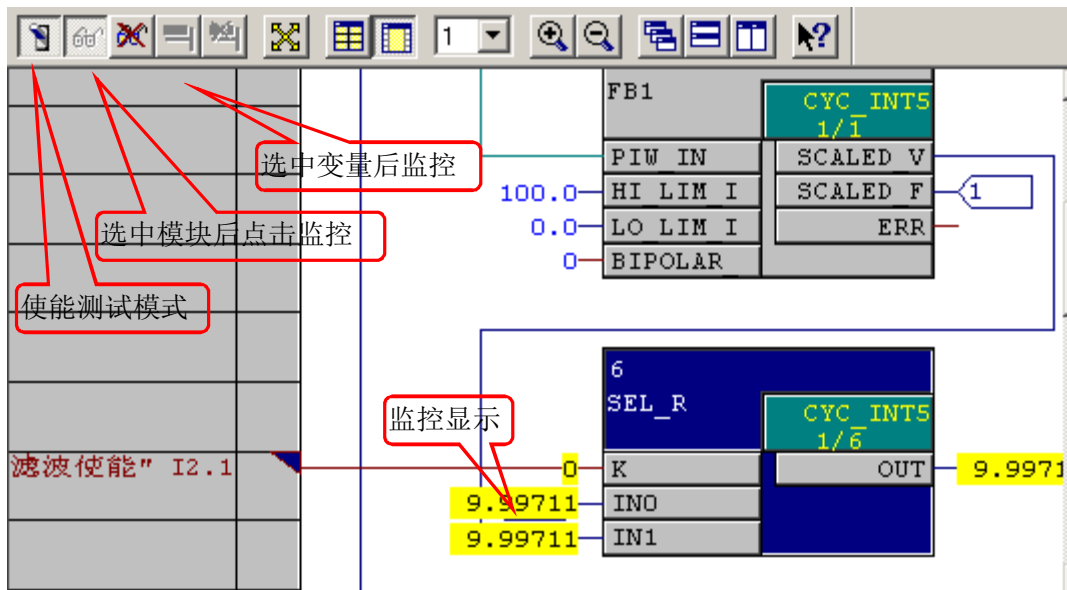


图 4-16: 监控程序

17) 在 View -> Dynamic Display 中可以监控变量

Status	Chart	Block:	I/O	Value	Unit	Comment
<input checked="" type="checkbox"/>	CFC(1)	3	PV_IN	9.99711		Process Variable I...
<input checked="" type="checkbox"/>	CFC(1)	3	SP_INT	8.0		Internal setpoint

图 4-17: 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，请按照本文提供的的地址连接下载 CFC 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。

5. S7-CFC 常见问题

5.1. 与监控调试相关问题

5.1.1. 问题：在 S7-CFC 中如何监控单独的 FC ？

问题：在 S7-CFC 中如何监控单独的 FC ？

解答：当编译 CFC chart 时，在中间步骤中系统会创建带有“ @SCLFileFromCFC_SFC...” 名称的 S7-SCL 源程序，用户可以将这个 S7-SCL 源程序单独编译，并且必须使能编译选项中的“ Create debug info”。在 SCL 编辑器中将程序下载到 CPU，在 S7-SCL 源程序中必要位置设置并激活断点，用户可以使用断点调试功能，同时在在 CFC 图中打开在线视图进行监控。

注 意 事 项：
在用 S7-SCL 编译器编译后，不能通过 CFC 装载，因为它显示项目不一致并建议重新进行编译。对于这种调试方法，并不建议使用。

5.1.2. 问题：对于内部嵌套功能块连接有何限制？

问题：对于内部嵌套功能块连接有何限制？

解答：对于内部嵌套功能块连接需要遵循如下规则：

- 内部嵌套的功能块的 I/O 不能与 Chart 的 I/O 相连接。
- 一个 block 的 IN 类型参数可以与 chart 的 IN 或 IN_OUT 类型的参数相连接。
- 一个 block 的 ANY 类型的 I/O 参数可以与 chart 的基本数据类型的 I/O 相连接（指针除外）。
- 不同的逻辑控制器类型，各数据类型之间互连的兼容性各不相同。如果不兼容，则连接过程中会显示相应的错误信息。
- 用户不能为 Chart 的输出分配参数，用户可以通过为其相连接的功能块的输出分配参数，相应的值即被赋予到相应的 Chart 输出中。
- FC 的结构化输出变量不能与 Chart I/O 相连接

5.1.3. 问题：编译时出现信息“ SCL 编译器不能执行”或“块中符号名称太多”？

问题：如何避免编译 CFC 程序时，出现信息“ SCL 编译器不能执行”或“块中符号名称太多”？

解答：在 CFC 中生成的 S7-SCL 源代码不能在 S7-SCL 中直接打开和编译。上述现象和信息可能会在大的 CFC 项目中出现，并且在有大量 S7-SCL 源代码的情况下也会出现。

SCL 编译器的注意事项:

当使用 S7-SCL 编译器编译 chart 时, 请确认在一个任务或运行系统组中没有安装太多块; 否则, 块就会变得太多和太大, 导致 S7-SCL 编译器出现上述错误信息。可通过分割块为几个运行系统组以避免这一情况。在编译过程中会产生一些小的函数 (FC) 。在 S7-SCL 编译器中大的 CFC 结构的编译过程需要花很长时间; 在某些情况下, 可能会有几分钟没有进程指示。

5.1.4. 问题: CFC 中最多能够显示多少个输入/输出?

问题: CFC 中最多能够显示一个 block 的多少个输入/输出?

解答: 大约 160 个输入及 160 个输出可以被显示。

5.1.5. 问题: 在 LAD 编辑器中启用监控功能时, 为什么会出现信息“ 30:484” ?

问题: 在 LAD/FBD/STL 编辑器中启用“ Monitor” 功能时, 为什么会出现信息“ 30:484” (在线—离线差异)?

解答: 如果在“ Monitor” 功能被启用后, 信息“ 30:484” (在线—离线差异) 出现在 LAD/FBD/STL 编辑器中, 是由于在线块和离线块间的时间标签不同引起的。由于涉及块顺序的在线信息不能分配到离线块单独的程序单元中, 块状态不能显示。

下述情况会发生所描述的现象 :

- a) 完成离线块(LAD/STL/FBD 编辑器)中的改变, 将变化载入到 CPU 中, 不保存变化, 关闭离线块, 然后再次离线打开这个块并启用“ Monitor” 功能。
- b) 完成离线块(LAD/STL/FBD 编辑器)中的改变, 将变化载入到 CPU 中, 离线保存它, 关闭块, 然后将块的另一版本复制到离线程序中或载入到 CPU 中, 当重新打开离线块时启用“ Monitor” 功能。
- c) 完成离线块(LAD/STL/FBD 编辑器)中的改变, 将其载入到 CPU 中, 然后将块的另一版本复制到 CPU 中并在块仍打开时启用“ Monitor” 功能。

请注意在以上假设 b) 和 c) 中块也可从其它应用中载入。记住使用 CFC 时, 装载管理完全由 CFC 保证; 也就是说, 除了 CFC 装载功能外, 不应当使用其它功能来装载块到 CPU (例如通过 SIMATIC 管理器或 LAD/STL/FBD 编辑器装载单独的块, 除非这些块是 S7-CFC 之外不相关的块)。CFC 装载管理包含块目录中所有块, 不仅是 CFC 的块! 由 CFC 创建的程序必须总是由 CFC 下载到 PLC, 因为只有这个装载功能能够保证配置数据与 PLC 数据一致。

5.2. 与使用技巧相关问题

5.2.1. 问题: 如何将 OB 的本地数据传送给 CFC 的 chart 使用?

问题: 如何将 OB 的本地数据传送给 CFC 的 chart 使用?

解答：在 CFC 中，用户可以将 CFC chart 编译成功能块，并型在 OB 中进行调用，对于 OB 的本地数据（例如 OB1 的扫描时间）可以将其赋值给 chart 的 I/O, 以供其使用。 用户可以按照如下步骤来实现此功能：

- 通过 CFC 菜单 "View > Chart I/Os" 定义 chart 接口，分配输入、输出、输入输出的变量名和数据类型。这些参数的数据类型必须与需要使用的 OB 本地变量的数据类型一致（例如：临时变量"OB1_MIN_CYCLE"的数据类型为 INT）。
- 在 CFC 中，通过菜单命令"Chart > Compile > Chart as Block..." 打开 "Compile chart as block type"对话框。
- 在"General"项中分配 FB 号，在"Compile for CPU"下选择 S7-300 或者 S7-400
- 确认 "OK"。组态的 CFC chart 就会被编译成一个功能块。
- 在 STEP7 项目中，打开 OB1，调用 CFC 块。CFC 块将会在"FB blocks"中找到。给 FB 块分配一个背景数据块，将本地变量赋值给此 FB 的输入/输出。

5.2.2. 问题：如何将“ MUL_R” 块和“ DWORD” 的操作数互连到一起？

问题：如何才能将“ MUL_R” 块和一个数据类型为“ DWORD” 的数据块的操作数互连到一起？

解答：“ MUL_R” 块将输入相乘，在输出上输出乘积。只允许使用 REAL 数据类型来设置此块的输入和输出。如果要使用“ MUL_R” 块来组态“ DWORD” 类型的数据块的操作数，则必须在输入上放置一个“ DW_R” 块（用于转换），并在输出上放置一个“ R_DW” 块（用于转换回到原类型）。

注意事项：

- 在 CFC 中不同的数据类型相连各不相同。然而，所有的数据类型总共可分为两类（基本数据类型，复合数据类型）。所有数据类型相连的原则就是他们必须相互兼容。
- ANY 型数据类型可以与上述的所有数据类型相连
- 在 CFC 中不支持数组类型（ARRAY）数据。也就是说，在 CFC 图中不支持带有数组型连接参数的功能块。
- 功能块（例如“与”门）的输出不能与图形连接的 IN_OUT 参数相连

5.2.3. 问题：如何添加 S7-CFC 之外的 DB 和 FC ？

问题：在使用 S7-CFC 编程时，如何添加 S7-CFC 之外的 DB 和 FC？

解答：在 S7-CFC 的编译选项中，为用户预留了一部分资源，如果用户希望在 blocks 文件夹中添加其它非 S7-CFC 类型的其它程序，可以使用这个编号范围。在此编号范围内的 FC/DB 不受 S7-CFC 编译的影响，用户可以自由选择编写这些块的语言，单独下载/调试。

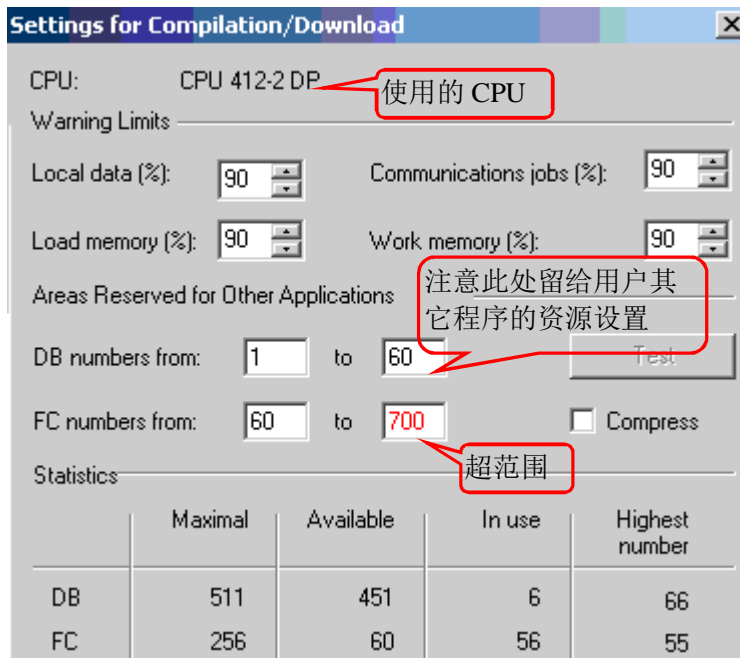


图 5-1: 编译/下载设置

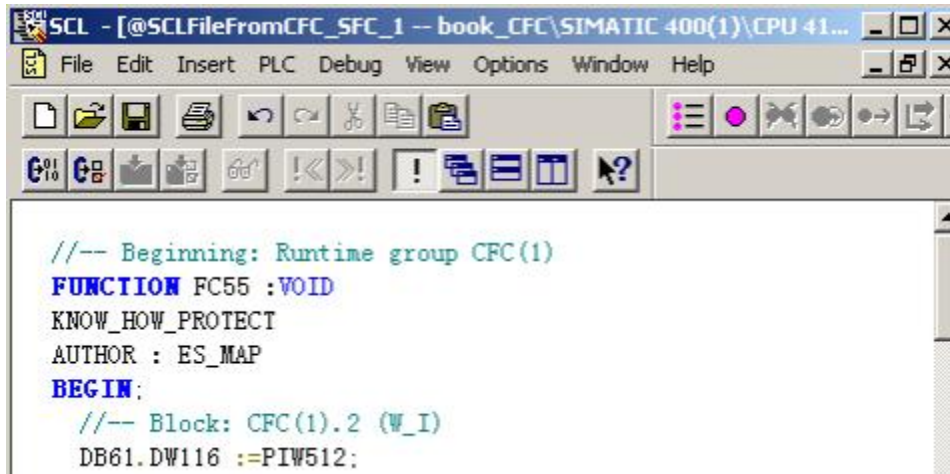
5.2.4. 问题：如何启用 CFC chart 的加密功能？

问题：如何启用 CFC chart 的加密功能？

解答：可以启用 CFC chart 的加密功能分为两种情况。

1. 情况 1: chart 作为完整程序时启用专有知识保护

- 打开一个 CFC chart
- 使用菜单功能 Chart > Compile > charts as program
- 编译后，源程序文件夹中自动生成一个“@SCLFileFromCFC_SFC...”名称的 S7-SCL 源程序
- 打开 SCL 源文件并通过删除“KNOW_HOW_PROTECT”行中的//注释符号激活块保护。
- 编译更改后的 SCL 源文件，产生的功能便受到保护。块文件夹中将出现一个锁定符号表示功能受到保护。
- 最后，将文件夹“charts”从 SIMATIC 管理器中删除。注意：删除 CFC chart 和文件夹“charts”后，无法再对块进行更改。



```
//-- Beginning: Runtime group CFC(1)
FUNCTION FC55 :VOID
KNOW_HOW_PROTECT
AUTHOR : ES_MAP
BEGIN:
  //-- Block: CFC(1).2 (W_I)
  DB61.DW116 :=PIW512;
```

图 5-2: 使能加密功能

2. 情况 2: chart 作为功能块时启用专有知识保护

- 打开一个 CFC chart
- 使用菜单功能 Chart > Compile > Charts as block type
- 在随后打开的窗口中，选择选项“ Know-how-Protecti on” 并点击 OK 确认。
- 在对块进行编译期间，产生一个 SCL 源文件和一个受保护的块。
- 在最后一步中，将 CFC 图和 SCL 源文件删除。

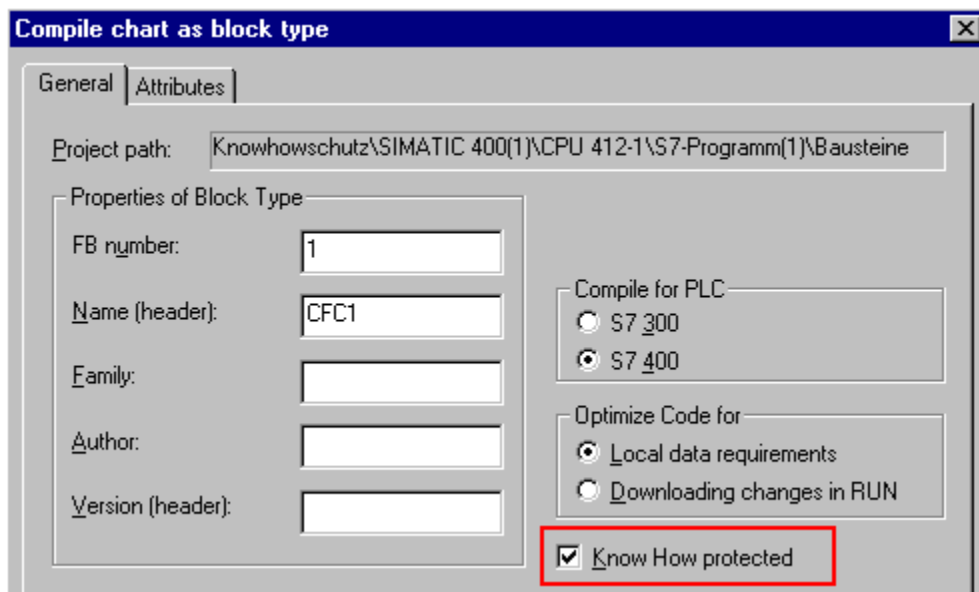


图 5-3: 使能加密功能