

1. 结构变量和 APL 简介

从 PCS 7 V7.1 开始，推出了全新的高级过程库（Advanced Process Library，简称 APL）。和 PCS 7 标准库相比，APL 提供了更为方便、灵活和丰富的控制功能。

APL 中的功能块广泛使用了结构变量来定义块引脚。结构变量（Struct）是一种复杂数据类型（Complex Data Type），一个结构变量由多个元素（Element）构成，元素可以是任何一种数据类型。图 1 显示了 FB 块引脚中的一个结构变量的定义。其中，PV 是一个结构变量，其中包含两个元素 Value 和 ST。Value 为 Real 类型，ST 为 Byte 类型。

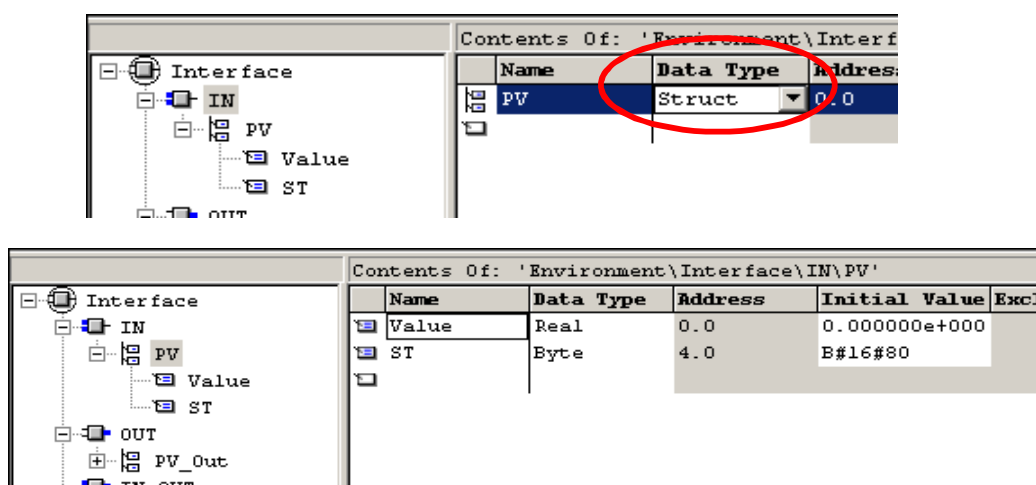


图 1 结构变量在 FB 块中的定义

与简单数据类型相比，结构变量将多个元素进行了“打包”，因此可以在单个引脚中包含更多的信息，例如，同时包含过程值和其信号状态（质量代码），或者同时包含一个量程的上限和下限。

结构变量简化了 CFC 中功能块的用户界面，通过结构变量之间的一个连线就可以传递多个相关的数值；结构变量把一组相关的数值组合成单个引脚，增强了程序的可读性；APL 利用结构变量的特点实现了一些特殊的功能，使过程控制更加方便灵活。

下面将详细介绍结构变量在 PCS 7 V7.1 中的功能特性和使用方法。

本文所述内容均基于 PCS 7 V7.1 SP2 版本。

2. 结构变量在 APL 中的应用实例

在 APL 中，功能块的设计充分利用了结构变量的优势。下面列举了 APL 中借助结构变量实现的一些典型应用。

2.1 信号状态 ST

APL 中大量使用了带有信号状态（ST）的结构变量。这一类结构变量具有固定的格式，由两个元素构成，第一个元素是信号的数值，其数据类型为简单数据类型（BOOL 或 REAL），元素名称为 Value，第二个元素为信号状态（旧称质量代码），数据类型为 BYTE，元素名称为 ST。

图 2 显示了 APL 中一个典型的带有信号状态的结构变量。

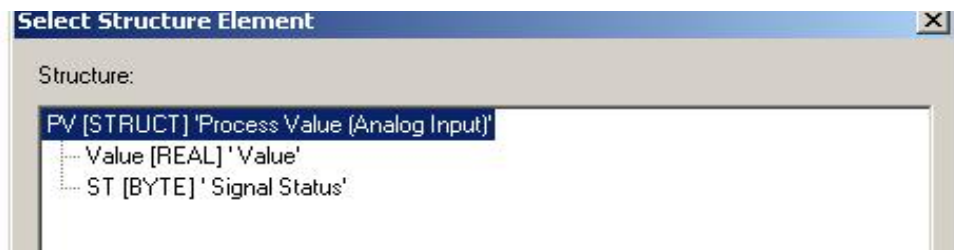


图 2 带有信号状态的结构变量

在 PCS 7 标准库中，信号状态和信号值分别定义为两个引脚（例如，CTRL_PID 块的引脚 PV_IN 为过程值，QC_PV_IN 为信号状态），因此在使用时需要分别连接。APL 将信号状态包含在结构变量中，一方面简化了使用，另一方面避免了由于引入信号状态而导致的引脚的数量的增加，所以在 APL 的块引脚中更广泛的使用了信号状态，增强了整个过程控制的安全性。

2.1.1 信号状态用于 OS 显示

信号状态可以直观的在 OS 上的图标、面板中显示。这样操作员不仅可以看到过程值，还可以了解该值的实际状态。如图 3 所示。

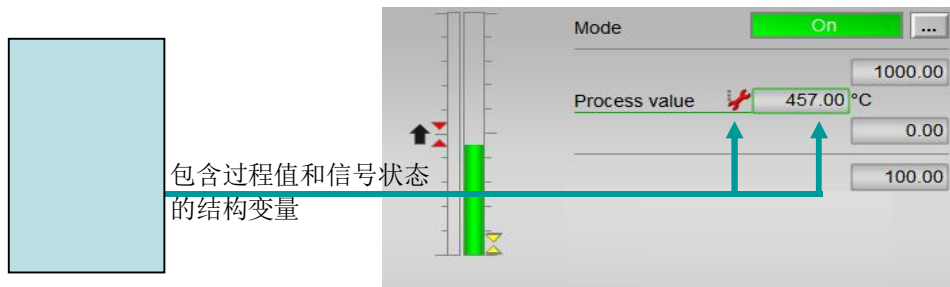




图 3 信号状态在 OS 中的显示

APL 中为信号状态定义了不同的数值以及不同的状态图标。例如，当 ST 值为 16 进制的 80H 时表示好值，60H 表示模拟值 ，00H 表示坏值 ，等等。详细的定义请参考 APL 帮助文档“ PCS 7 - Manual for Advanced Process Library V7.1” 1.2.3.3 节。

2.1.2 信号状态的获得和传递

直接从过程 I/O 获得的过程值通过通道驱动块（如 Pcs7DiIn）连接到控制功能块（如 MotL）。这些过程 I/O 的状态由自动生成的模块驱动程序获得，通过通道驱动块根据这些状态计算出信号状态 ST 的值，并传递给控制功能块。

非直接过程值的信号状态由相应的控制功能块计算得出。例如，连锁块 Intlk02 的逻辑输出值 Out 的信号状态就不是直接来自过程 I/O，而是根据各个输入值的信号状态按照一定的规则计算得出的。

关于信号状态计算的详细规则可以参考 APL 帮助文档“ PCS 7 - Manual for Advanced Process Library V7.1” 1.2.3.3 节。

在过程控制的信号流中包含了信号状态，控制功能块的算法也包含了信号状态。因此增强了过程控制的安全性。如图 4 所示，连锁块的两个输入均为 1，AND 逻辑输出为 1，如果只考虑信号值，则不应该触发连锁（0 信号表示连锁有效）。如果结合了信号状态，由于连锁的第一个输入的信号状态为坏值（此处为 28H），这个状态也传递到了连锁的输出端，因此在 MotL 块上连锁被触发，同时，面板中也会显示相关状态。

关于信号状态对连锁的影响，可以参考 APL 帮助文档“ PCS 7 - Manual for Advanced Process Library V7.1” 1.2.4.4 节。

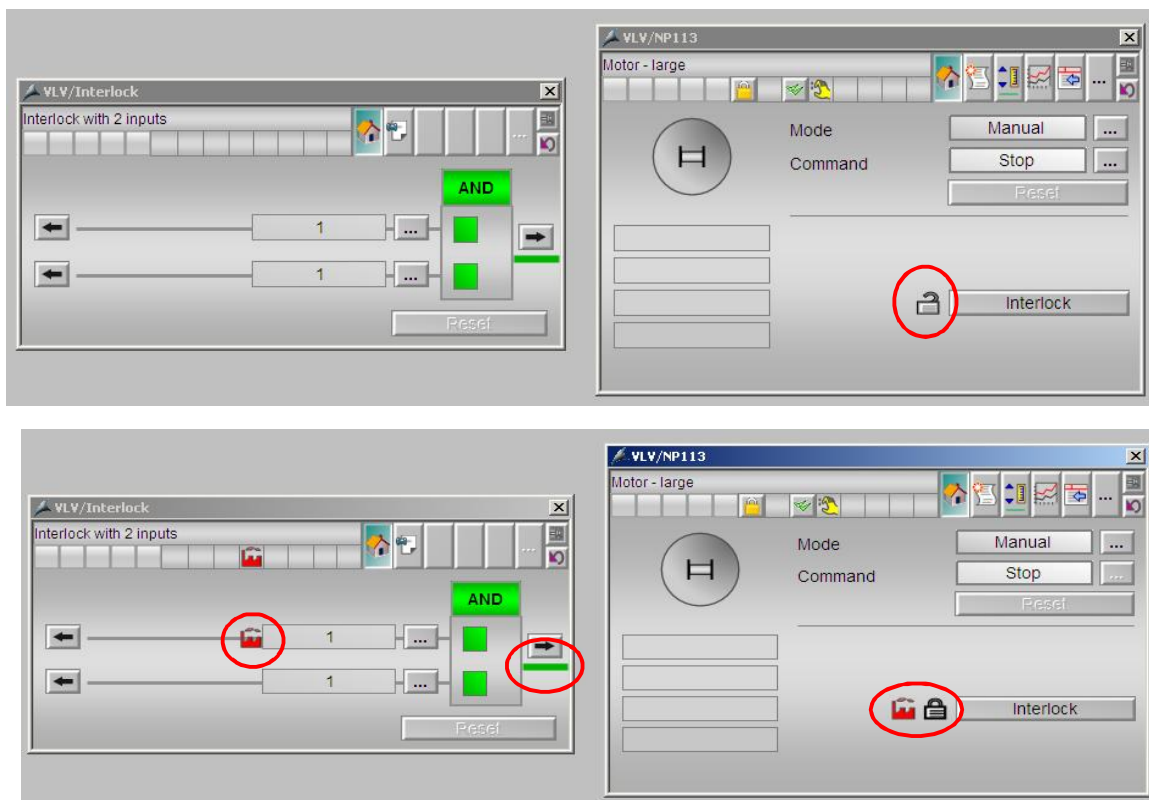


图 4 从安全性考虑，信号状态可以影响连锁

2.1.3 特殊信号状态值 FF

在 APL 中，利用一个特殊的信号状态值 FF 实现了在 OS 操作面板中自动隐藏相关对象的功能。例如，MotL 块有三个用于连锁保护的输入引脚，分别是 Permit, Intlock 和 Protect，用户可根据工艺要求使用其中的一个或多个。

如图 5 所示，Intlock 引脚在 CFC 中进行了连接，就会自动在 MotL 的 OS 面板上显示出 Intlock 的按钮和图标，Permit 和 Protect 引脚没有连接，在 OS 中就没有显示。

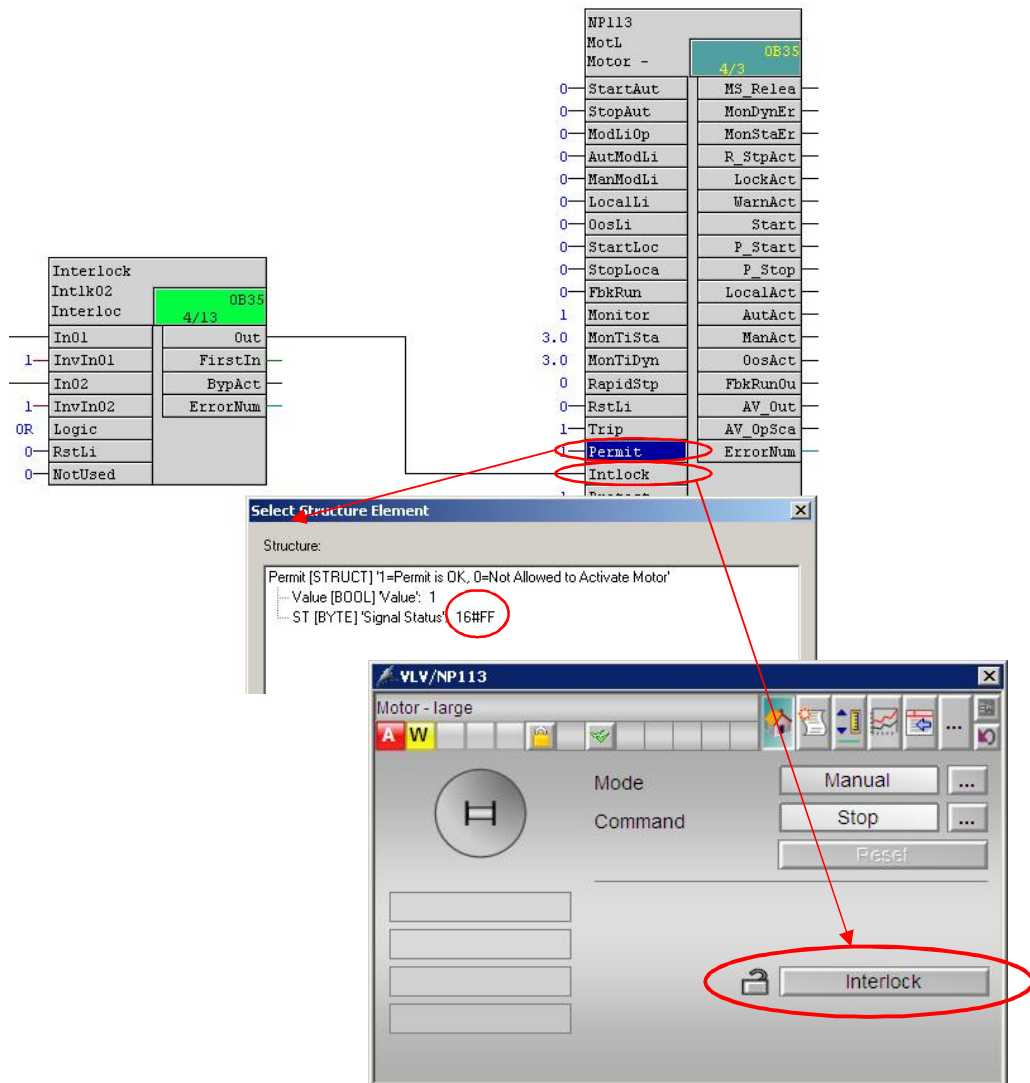


图 5 信号状态 FF 在 OS 中自动隐藏相关对象

这种“不连线就不显示”的功能是利用特殊的信号状态 FF 实现的。在 APL 的 OS 面板中，某些对象的信号状态值是 FF 时就会隐藏，在 CFC 中，这一类引脚的 ST 值缺省设置为 FF。所以当没有连线的时候，这些对象就不会显示，而一旦连线之后，其 ST 值来自于输入，这个值不等于 FF 时，面板上的对象就显示出来了。

2.2 模拟量的量程

在 PCS 7 标准库中，模拟量量程上下限一般通过两个引脚分别设置（例如，CH_AI 块的引脚 VHRANGE 和 VLRANGE）。在 APL 中，上下限通过一个结构变量设置。除此之外，APL 中通道驱动块还可以将输入的量程传递到输出端，这样就可将量程传递给其他功能块。

如图 6 所示，在通道块 Pcs7AnIn 中，通过结构变量 Scale 设置量程，该量程经过 ScaleOut 传递给模拟量监视块 MonAnL 的 PV_OpScale 用做 OS 面板中的棒图上下限。

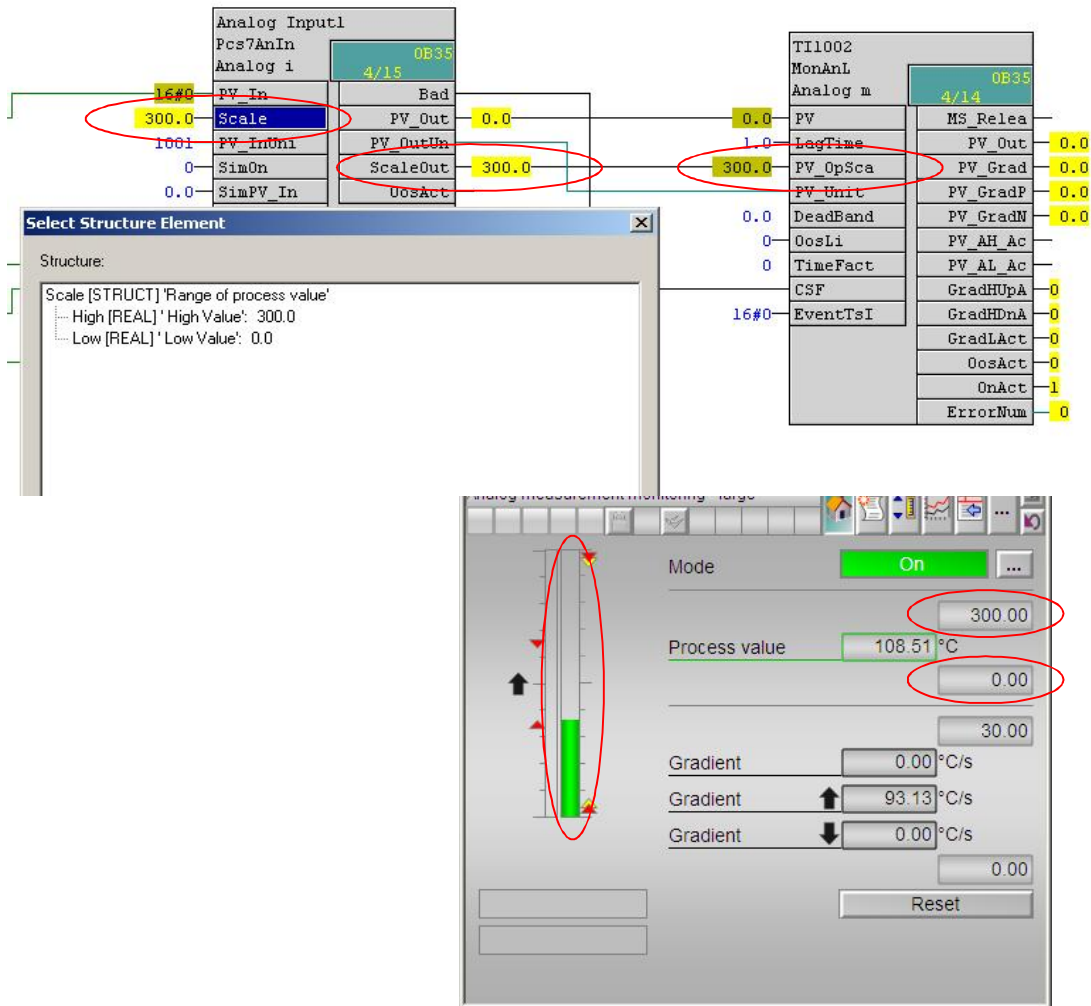


图 6 结构变量 Scale

2.3 Feature 和 OS_Perm

在 APL 中，很多功能块具有一个名为 Feature 的引脚。Feature 是一个包含 32 个 BOOL 元素的结构变量。每一个 BOOL 值都对应一个设置选项，用来选择块的不同功能特性。关于 Feature 的详细介绍请参考 APL 帮助文档“PCS 7 - Manual for Advanced Process Library V7.1” 1.5 章。图 7 显示了 VML 块的 Feature 引脚。

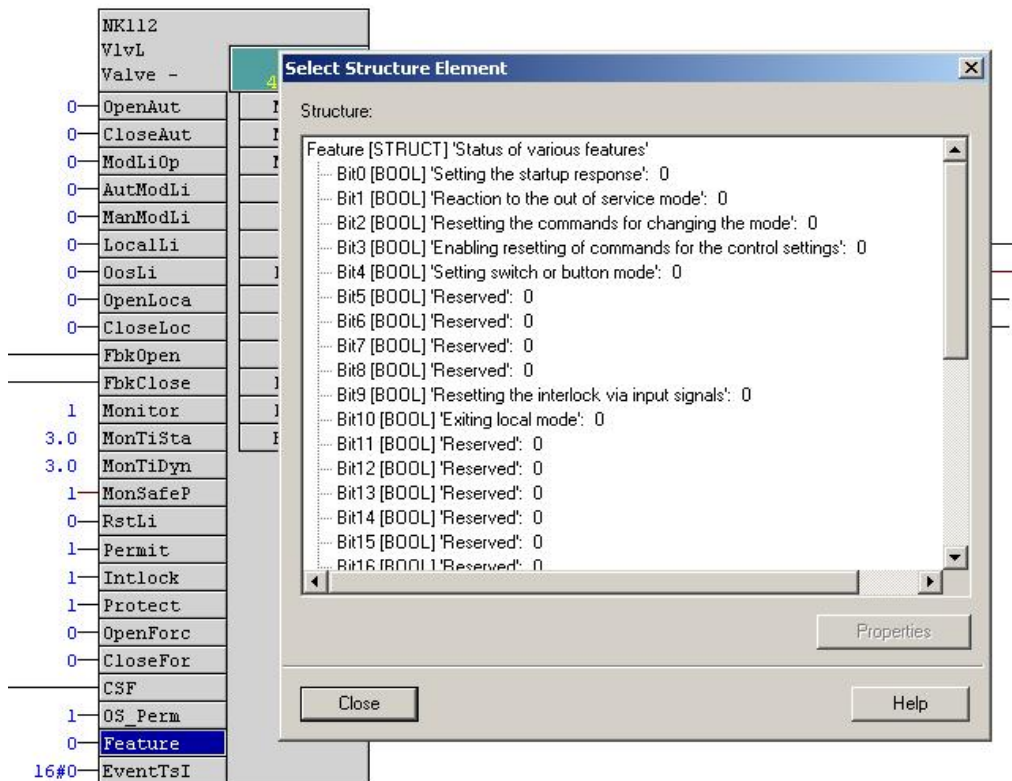


图 7 结构变量 Feature

在 APL 中，很多带有 OS 监控功能的块都具有一个名为 OS_Perm 的引脚。OS_Perm 是一个包含 32 个 BOOL 元素的结构变量。单个 BOOL 值用于设置是否允许操作员在 OS 上操作某个具体的命令。详细定义请参考相关功能块的帮助文档。

图 8 显示了 VlvL 块的 OS_Perm 引脚。

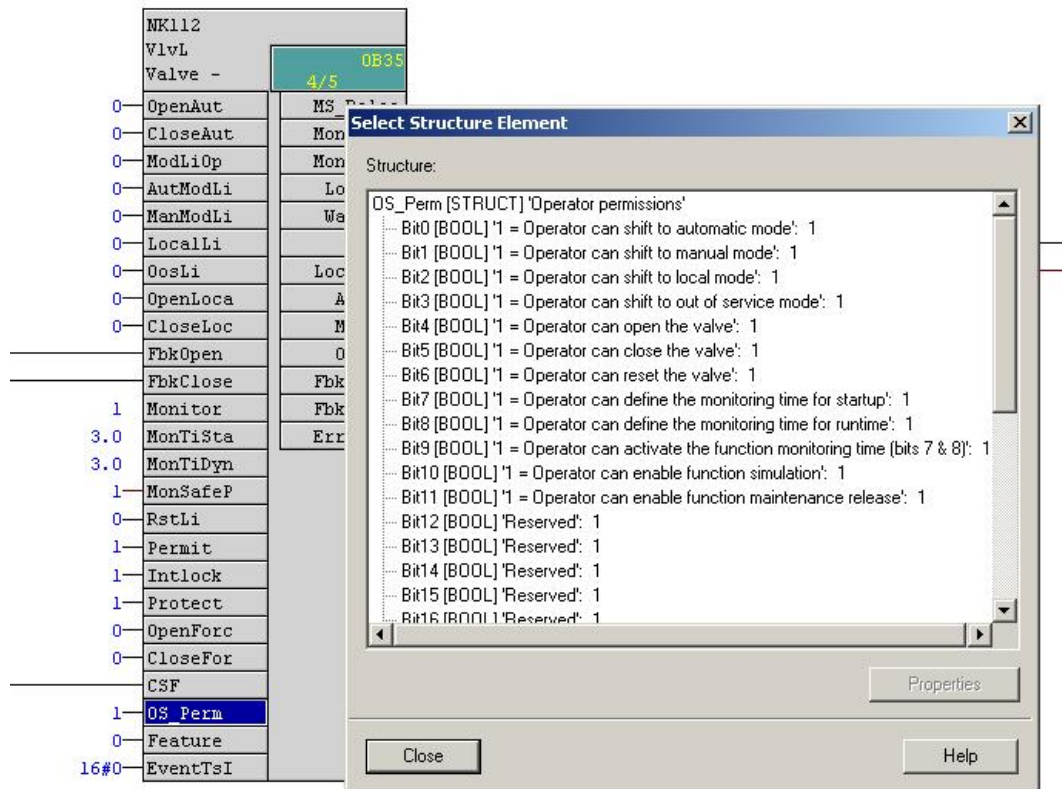


图 8 结构变量 OS_Perm

Feature 和 OS_Perm 都是由 32 个 BOOL 元素构成的结构变量。与直接使用 DWORD 类型相比，使用结构变量能够让设置更为直观，而且用户可以单独设置每一个位。如图 7、8 所示，用户在 CFC 中就可以直接查看和修改每一个 BOOL 值。

2.4 其他结构变量

在 APL 中，还有一些结构变量用于在不同的块之间传递数据。例如，附加模拟量块 AV 和 MotL 块之间通过引脚 AV_Tech 相连。AV_Tech 是一个结构变量，包含了附加模拟量的大量信息。如图 9 所示。

由于使用了结构变量，大量数据的交换只需一根连线，简化了用户界面，方便了操作。

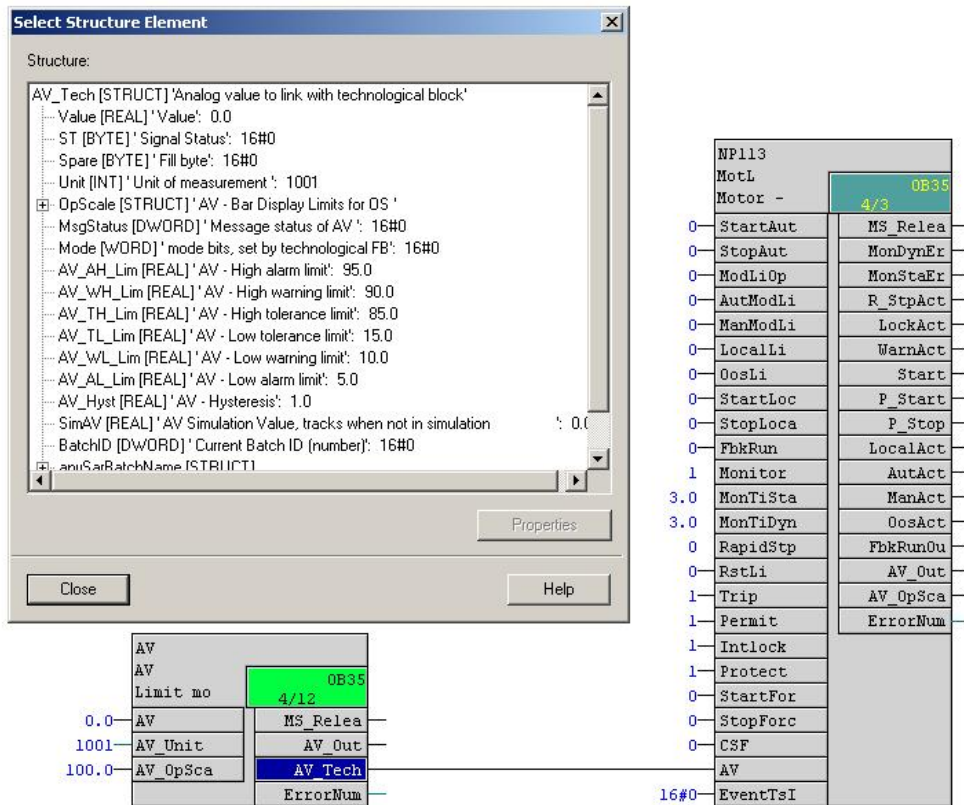


图 9 结构变量 AV_Tech

3. 结构变量在 CFC 中的使用

第 2 章介绍了结构变量在 APL 中的一些应用实例。我们看到了使用结构变量所带来的好处。

在使用结构变量时，与简单变量有一些不同。接下来的几章将分别介绍在 CFC、Process Object View、SFC 等环境中如何使用结构变量。

3.1 结构变量之间的连接

在 CFC 中，连接两个结构变量的前提条件是它们具有相同的结构。相同的结构是指，两个结构变量定义的元素的名称、数据类型、顺序都必须完全一致。如图 10 中 PV_Out 和 PV 之间的连接。

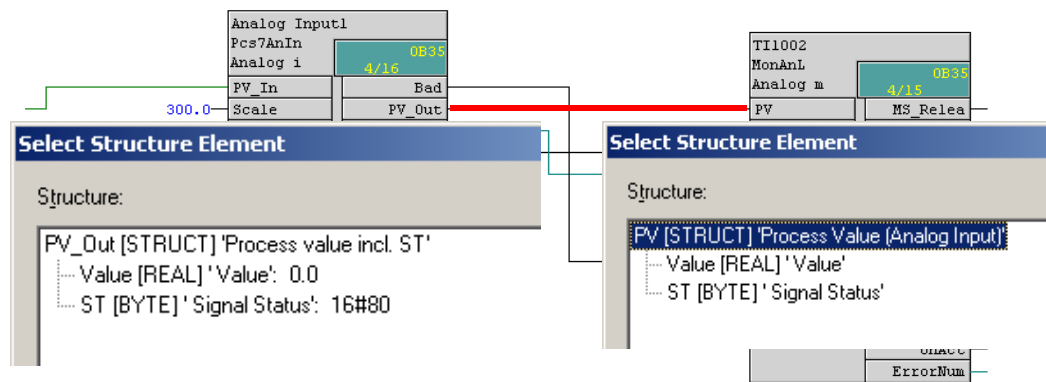


图 10 具有相同结构的两个结构变量互连

3.2 结构变量和非结构变量之间的连接

一般情况下，结构变量和非结构变量之间不能直接连接。只有在两种特殊情况下，才允许直接连接。

1) ANY 型的变量可以和结构变量直接连接。例如图 8 中，AV 块的 AV_Tech 引脚是一个结构变量，而 MotL 块的 AV 引脚是一个 ANY 型的变量。

2) 符合第 2.1 节所描述的带有信号状态的结构变量可以和相应类型的非结构变量直接连接。即：结构变量必须只包含两个元素，第一个元素为 Value，数据类型为 REAL 或者 BOOL，第二个元素为 ST，其数据类型为 BYTE。这种结构变量可以直接连接到与其 Value 类型一致（REAL 或者 BOOL）的非结构变量（PCS 7 V7.1 SP1 以上版本支持）。如图 11 所示。图中 OpenAut、FbkOpenOut 和 FbkCloseOut 都是结构变量，分别直接连接到了 AND 和 OR 块的 BOOL 型引脚上。

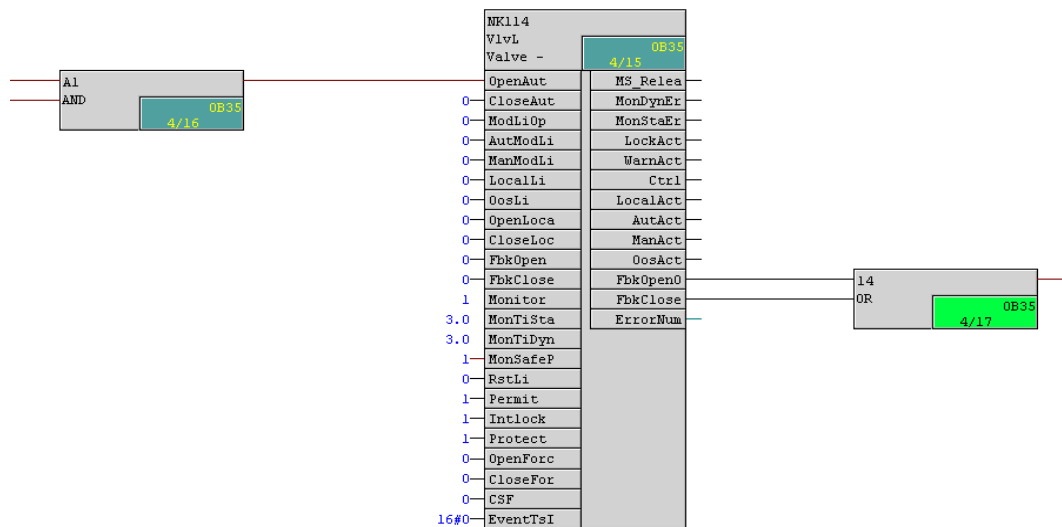


图 11 结构变量和非结构变量的直接连接

这种直接连接有一些限制，下列几种情况下无法使用这种直接连接：

跨 AS 的程序互联

连接到全局变量（例如全局运算符，全局 DB 块）

连接到运行组的使能信号

连接到一个 CFC Chart 的接口

在无法直接将结构变量和非结构变量互连的情况下，可以使用 APL 提供的转换块实现连接。APL 提供的转换块见图 12。

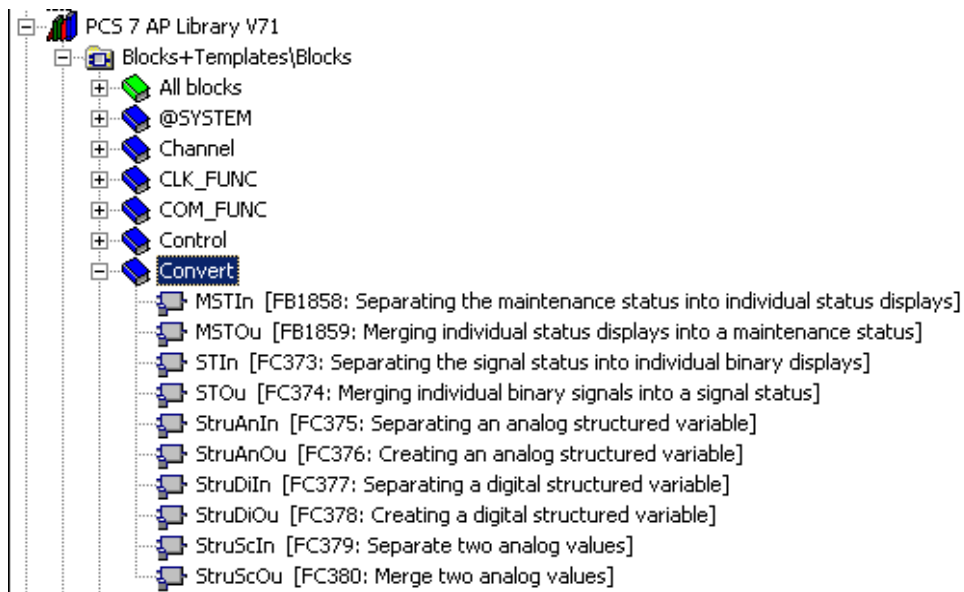


图 12 APL 中的转换块

图 13 显示了转换块的使用。

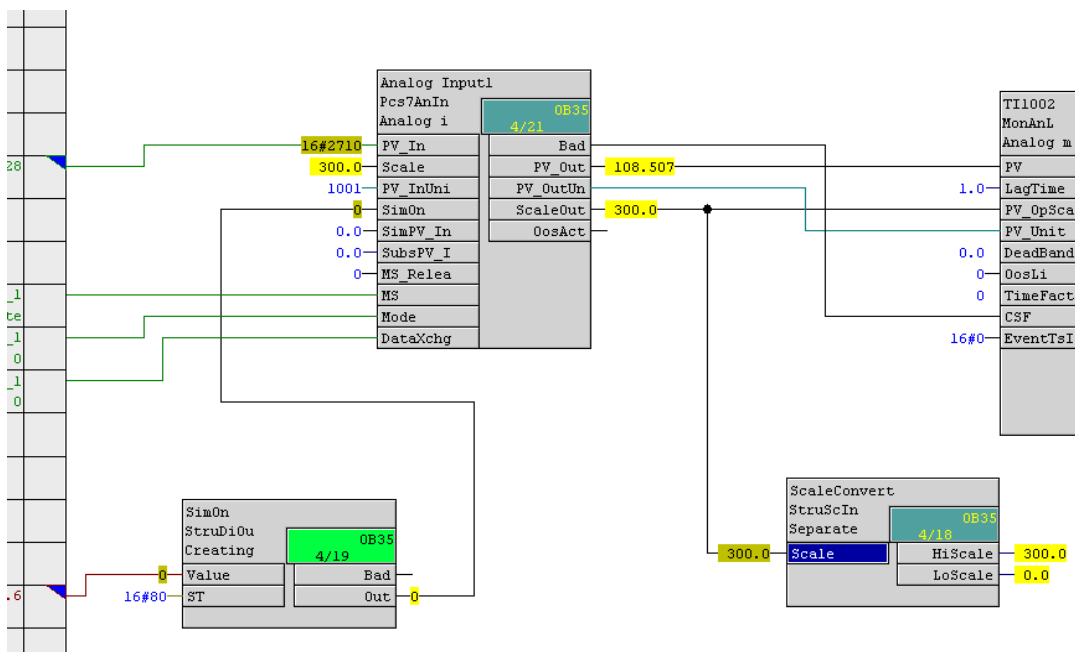


图 13 使用转换块和非结构变量互连

StruDiOu 块将一个 BOOL 型变量转换为结构变量连接到 Pcs7AnIn 的 SimOn 引脚。StruScIn 将结构变量 ScaleOut 转换为两个 REAL 型的输出。

为了减少这种转换，建议尽量避免 APL 库和 PCS 7 标准库混合使用。

3.3 结构变量的取反

在 CFC 中，BOOL 类型的输入引脚可直接通过右键菜单→Invert 进行取反。

由两个元素 Value (BOOL) 和 ST (BYTE) 组成的结构变量也可以同样的方式取反 (PCS 7 V7.1 SP1 以上版本支持)。如图 14 所示。

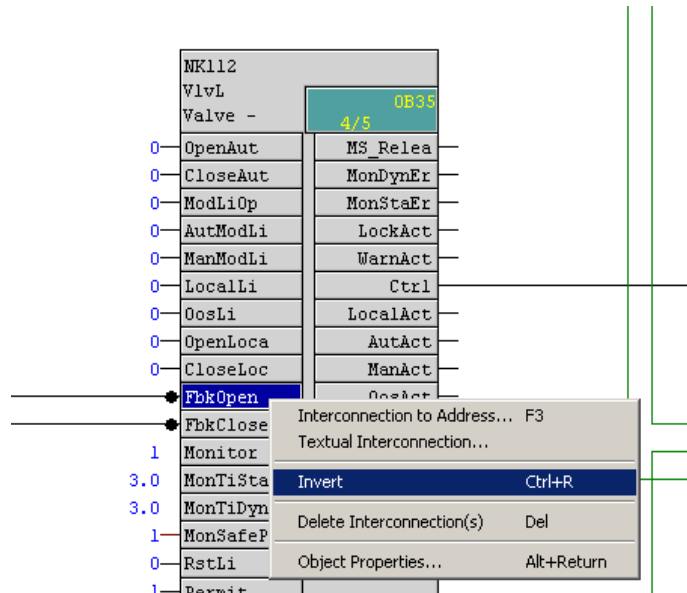


图 14 Value 为 BOOL 类型的结构变量的取反

4. 结构变量在 SFC 中的使用

双击打开一个步 (Step) 或者转移条件(Transition)，点击 Browse 按钮浏览到相应的结构变量，点击右键→Open Structure，在弹出窗口中选择结构变量的元素。如图 15 所示。

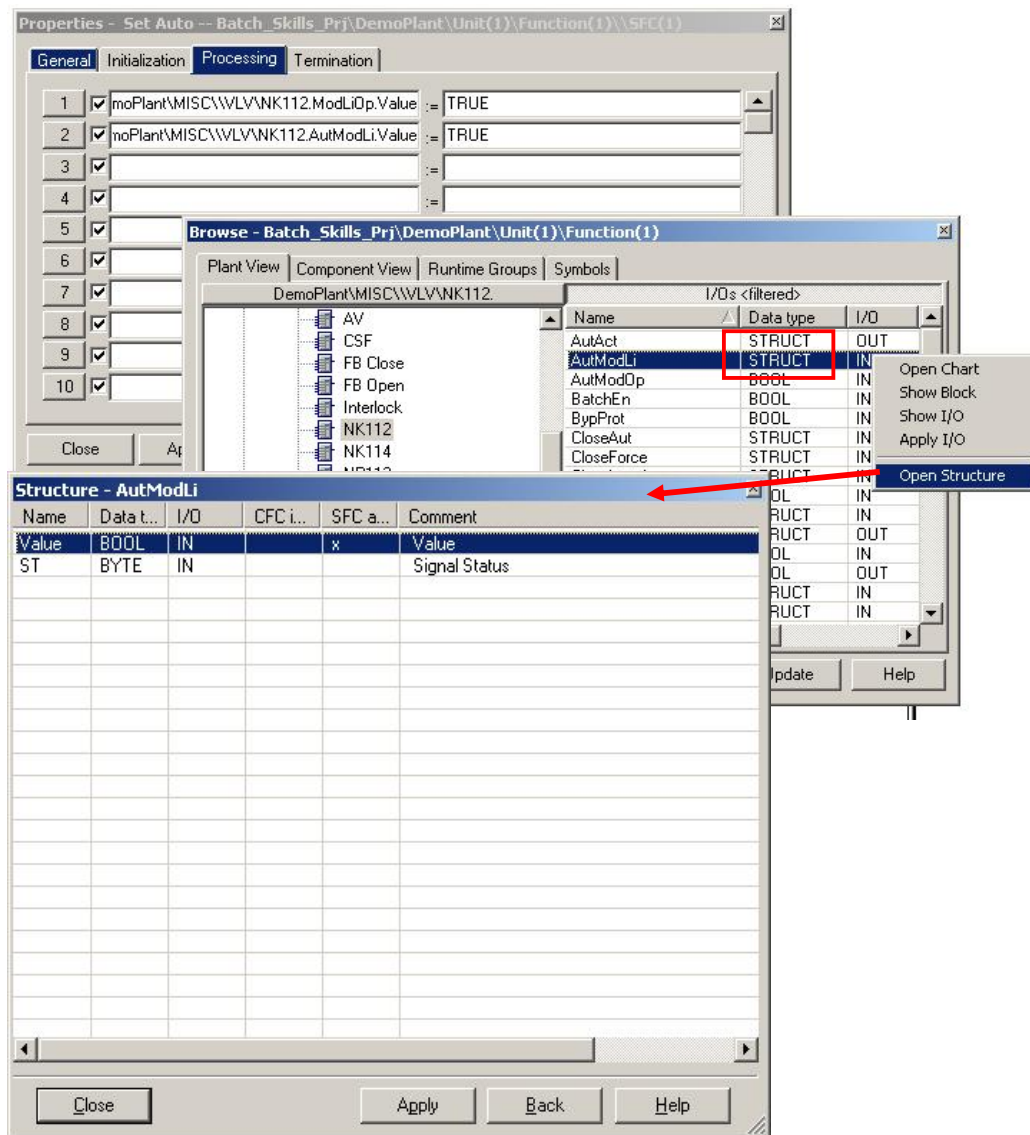


图 15 在 SFC 中使用结构变量的元素

结构变量的元素在 SFC 中的显示格式为：结构变量名.元素名

在转移条件中，只能使用结构变量中的元素，而不能使用结构变量本身。如果直接插入所选的结构变量，则会出现错误提示对话框。

在步（Step）中，既可以给结构变量中的元素赋值，也可以给结构变量赋值。如果想给结构变量赋值，只需双击选中的结构变量即可插入。可以将一个结构变量的值赋给另一个结构变量，当然，前提是二者的结构定义完全相同。如图 16 所示。

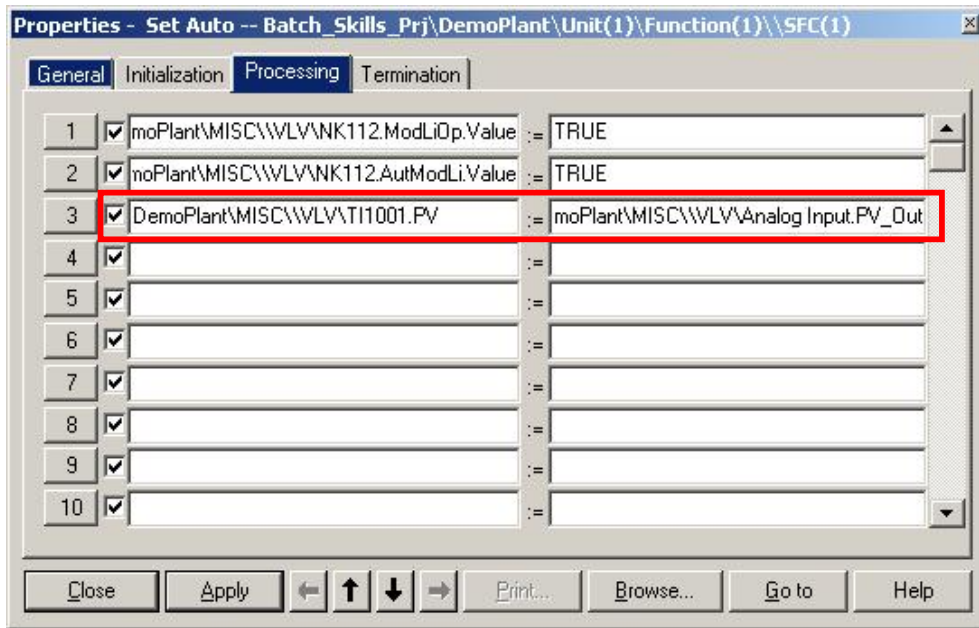


图 16 在步中可以直接给结构变量赋值

另外一种更为便捷的使用方式是，同时打开 SFC 和 CFC 编辑窗口，直接把需要连接的变量从 CFC 窗口拖拽到 SFC 的步（Step）或者转移条件(Transition)中，编辑器会默认插入相应结构变量的 Value 元素。如图 17 所示。

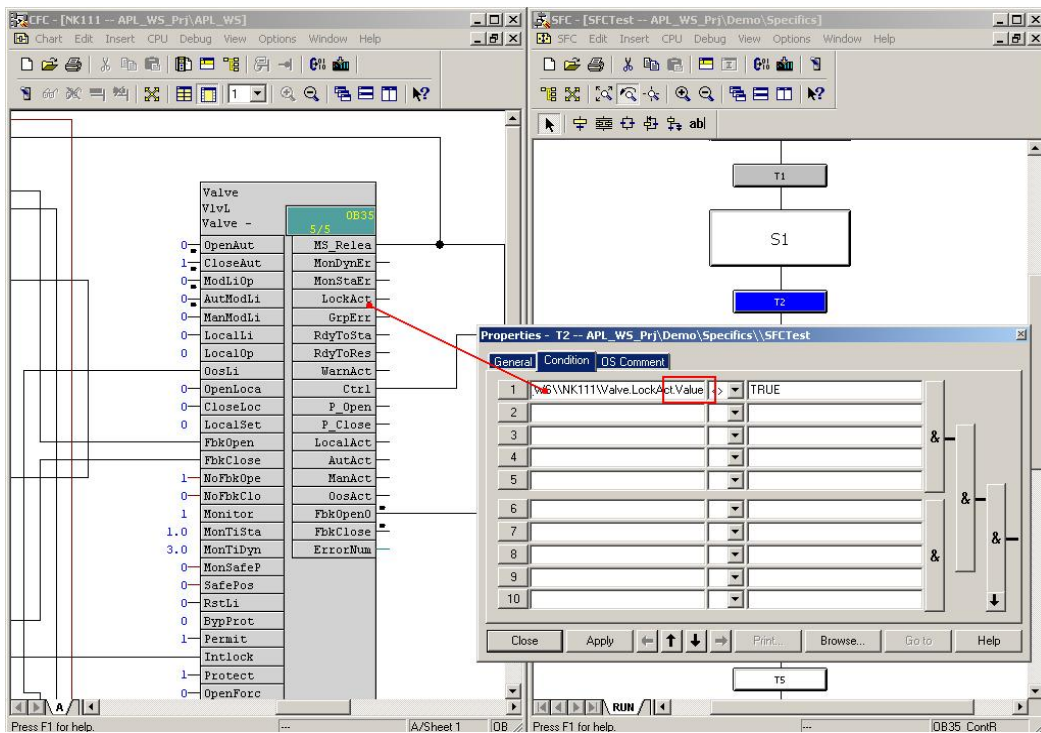


图 17 把 CFC 中的结构变量拖拽到 SFC 中

5. 结构变量在 SFC Type 中的使用

SFC Type 具有可自定义的接口。在 SFC Type 的自定义接口中也可以使用结构变量。

如图 18 所示，在接口区域展开结构变量，选择其中的元素，直接拖拽到组态界面中即可。

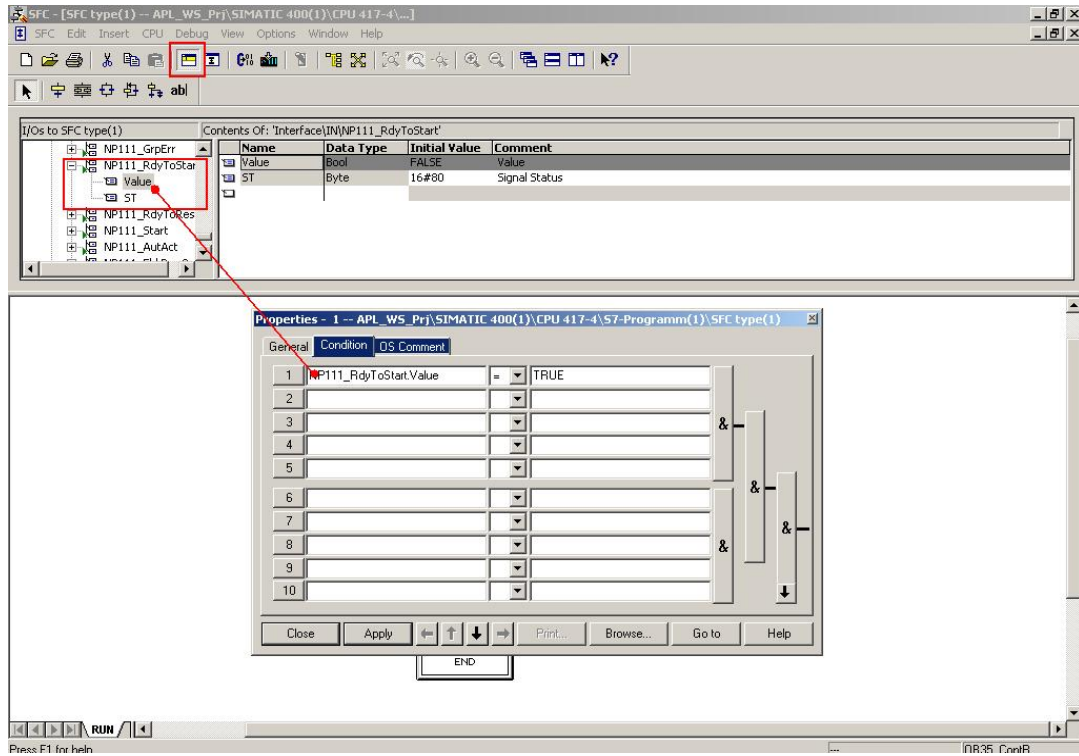


图 18 使用 SFC Type 接口区域的结构变量

在 SFC Type 自定义接口的块触点（Block Contact）如果是一个包含结构变量的功能块类型，例如 APL 库中的 MotL 块，那么在 SFC Type 中通过拖拽的方式引用该块触点时，弹出的变量选择对话框中既包含了结构变量，也包含了其中的元素，用户可根据需要选择。如图 19 所示。

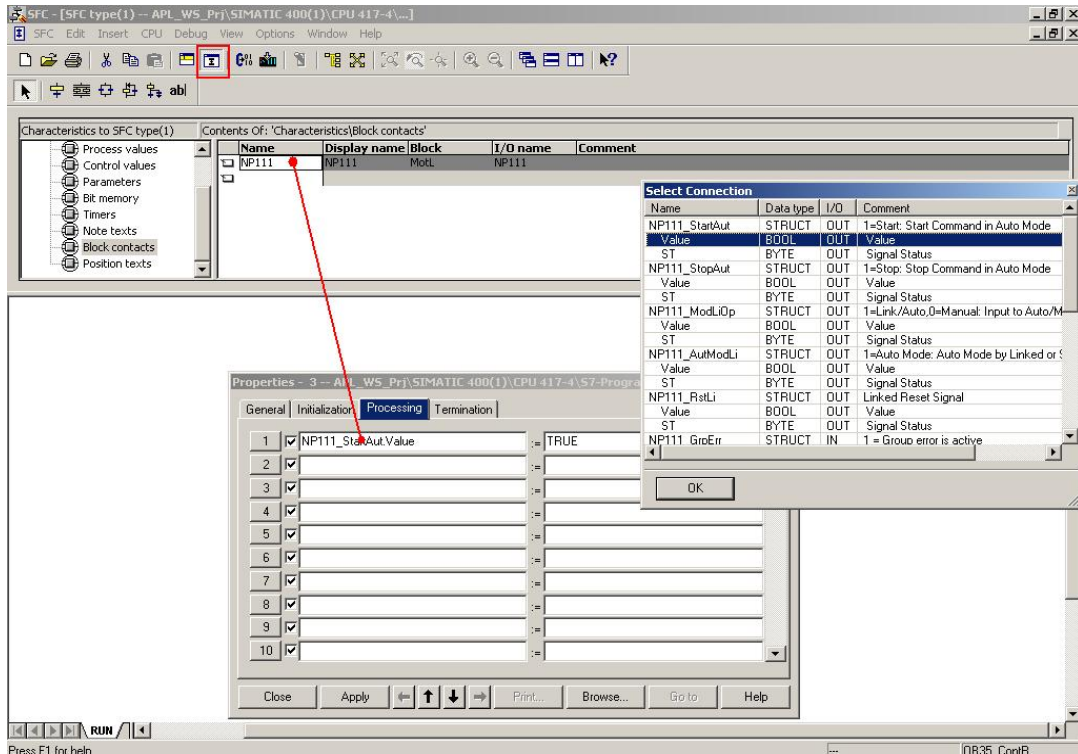


图 19 使用 Block Contact 中的结构变量

6. 结构变量的属性

6.1 CFC 中查看和修改结构变量的属性

结构变量由多个元素构成，因此对结构变量的操作也需要针对其包含的元素分别进行。

在 CFC 中，双击一个结构变量引脚，首先弹出结构变量对话框，然后再选择其中某个元素进行单独的操作。如图 20 所示。

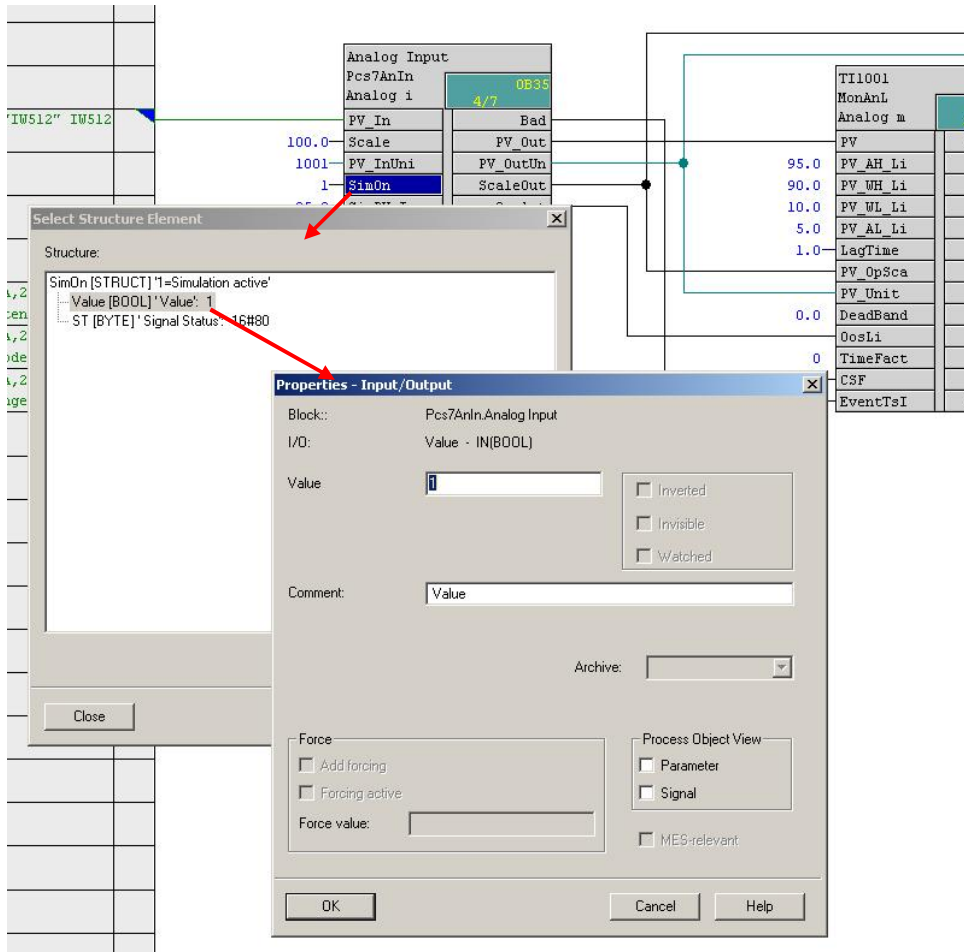


图 20 查看和修改结构变量中单个元素的属性

在 CFC 的块引脚上显示的是结构变量中第一个元素的数值。

进入在线调试状态后，当鼠标移动到某个结构变量上时，在弹出的提示框中会显示结构变量的完整信息。如图 21 所示。

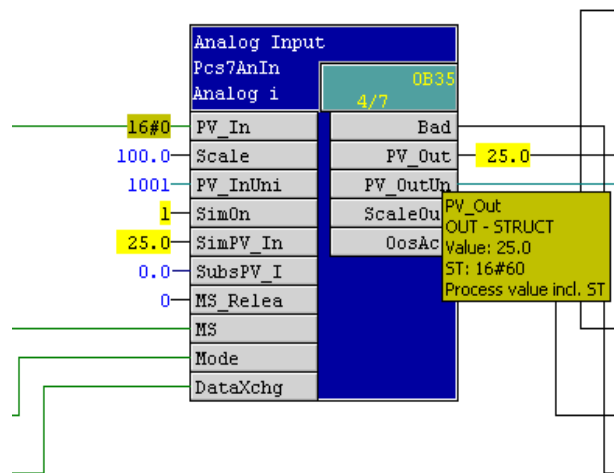


图 21 调试状态下显示结构变量的完整信息

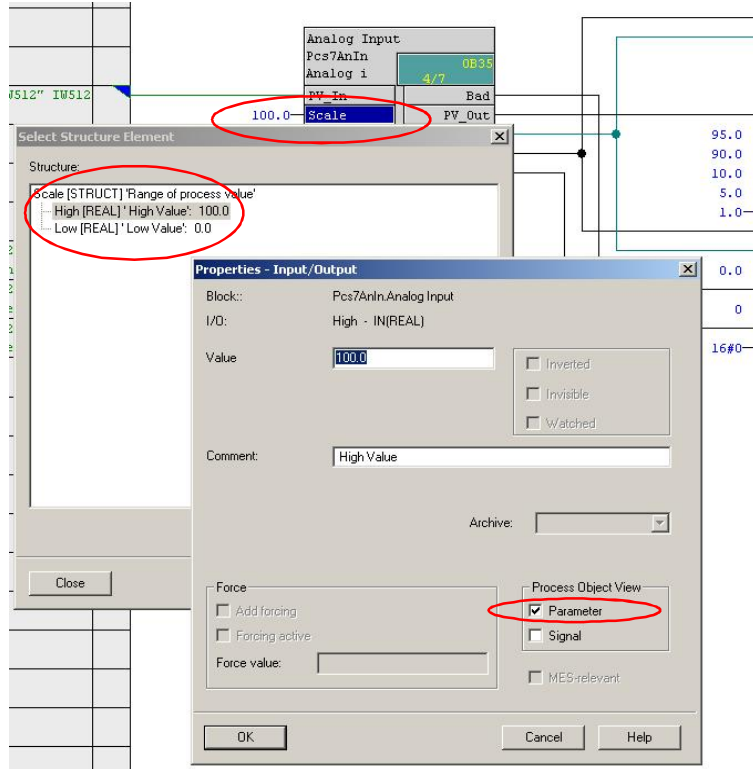
6.2 在 Process Object View 中查看和修改结构变量的属性

使用 Process Object View (POV) 可以集中进行批量参数修改。(关于过程对象视图的详细介绍可参考网上文档:

<http://ad.siemens.com.cn/download/searchResult.aspx?searchText=A0149>)

块引脚是否能在 POV 中显示, 取决于其 Parameter 或者 Signal 属性。对结构变量来说, 这一属性是针对其中的元素单独设置的。在 POV 中显示的是结构变量中具有 Parameter 或者 Signal 属性的元素。

如图 22 所示。Scale 引脚的两个元素 High 和 Low 分别设置了 Parameter 属性。在 POV 中可以看到 Scale 结构变量和其中的两个元素。元素以 结构变量名.元素名 的格式显示, 例如 Scale.High。



Filter by column: I/O name Display: Scale Filter general:											
Hierarchy	Chart	Chart comment	Block	Block comment	I/O name	I/O comment	Process tag...	Category	Value	Unit	
1	DemoPlant\MISC\	VLV	Analog Input	Analog input driver	Scale	Range of process value					
2	DemoPlant\MISC\	VLV	Analog Input	Analog input driver	Scale.High	High Value			100.0		
3	DemoPlant\MISC\	VLV	Analog Input	Analog input driver	Scale.Low	Low Value			0.0		
4	DemoPlant\MISC\	VLV	Analog Input1	Analog input driver	Scale	Range of process value					
5	DemoPlant\MISC\	VLV	Analog Input1	Analog input driver	Scale.High	High Value			300.0		
6	DemoPlant\MISC\	VLV	Analog Input1	Analog input driver	Scale.Low	Low Value			0.0		
7	DemoPlant\MISC\	VLV	ScaleConvert	Separate two analog values	Scale	Scale Value					
8	DemoPlant\MISC\	VLV	ScaleConvert	Separate two analog values	Scale.High	High Value					
9	DemoPlant\MISC\	VLV	ScaleConvert	Separate two analog values	Scale.Low	Low Value					

图 22 POV 中显示结构变量

6.3 功能块开发过程中设置结构变量的属性

如果用户需要开发自定义的功能块，则需要为功能块的引脚设置相应的属性，例如，用 S7_m_c 属性控制引脚是否可以上传到 OS 上生成监控变量。（关于自定义功能块的详细内容，请参考网上文档：

<http://ad.siemens.com.cn/download/searchResult.aspx?searchText=A0163>)

当块引脚是一个结构变量时，需要为其中的元素定义相关的属性。区别于简单类型变量的属性，这种属性名称中带有 x，例如 S7_xm_c。定义的格式如图 23 所示。

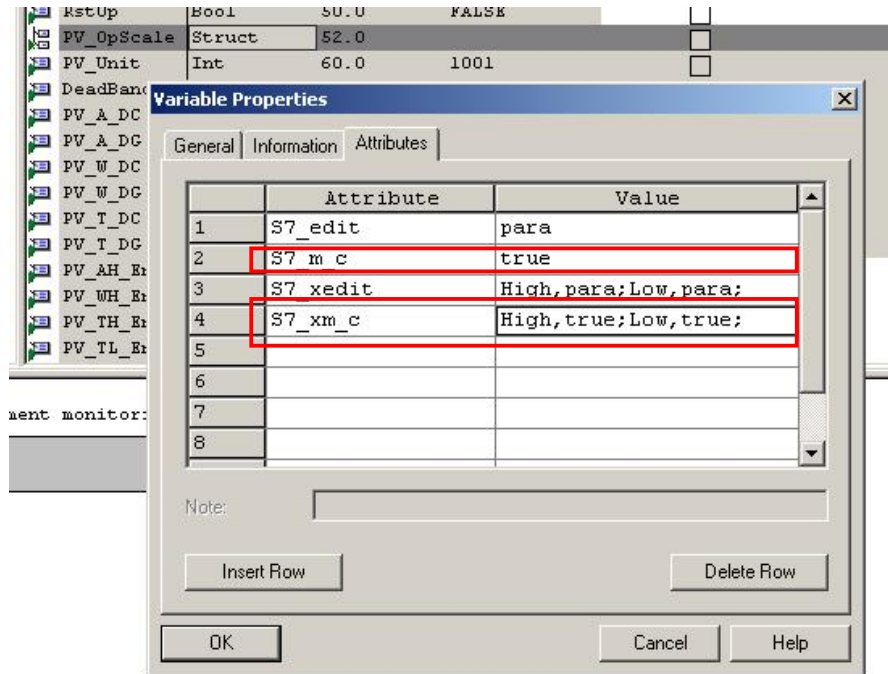


图 23 结构变量的属性定义

上图显示了 MonAnL 块的 PV_OpScale 引脚的属性定义。其中 S7_m_c 定义了结构变量自身的属性，S7_xm_c 定义了结构变量中元素 High 和 Low 的属性。

S7_xm_c 的格式为：元素名称 1,属性值;元素名称 2,属性值; ...

元素名称和属性值之间用“,”分隔，每个元素定义结束后用“;”结尾。

上述定义完成后，编译 OS，在 OS 上获得的变量名称如图 24 所示。



图 24 结构变量上传到 OS 上的变量名称

结构变量上传到 OS 后并不是一个结构，而是每个元素分别生成一个变量。变量名称格式为：结构变量名#元素名