

声明块接口

9.1 块接口概述

简介

接口中包含有块所用局部变量和局部常量的声明。这些变量可分为以下两组：

- 在程序中调用时构成块接口的块参数。
- 用于存储中间结果的局部数据。

变量声明可用于定义程序中的块的调用接口，以及块中需使用的变量/常量名称和数据类型。

函数块接口中还定义了分配给函数块的背景结构。

块接口的表格或文本表示

系统默认，接口显示为表格视图。在 SCL 中，接口也可显示为文本视图。

另请参见：

选择块接口的表格或文本式表示方式 (页 9612)

块参数

下表列出了块参数的类型：

类型	区域	函数	可用于
输入参数	Input	其值由块读取的参数。	函数、函数块和某些类型的组织块
输出参数	Output	其值由块写入的参数。	函数和函数块
输入/输出参数	InOut	调用时由块读取其值，执行后又由块写入其值的参数。	函数和函数块
返回值	返回	返回到调用块的值。	功能

根据打开块的类型，可能显示其它区域。

9.1 块接口概述

局部数据

下表列出了局部数据的类型：

类型	区域	函数	可用于
临时局部数据	Temp	用于存储临时中间结果的变量。只保留一个周期的临时本地数据。如果使用临时局部数据，则必须确保在要读取这些值的周期内写入这些值。 注： 如果在具有标准访问权限的函数 (FC) 中使用临时本地数据，请在使用前初始化该数据。否则，这些值可能为随机数。	函数、函数块和组织块 注： 临时局部数据不显示在背景数据块中。
静态局部数据	Static	用于在背景数据块中存储静态中间结果的变量。静态数据会一直保留到被覆盖，这可能在几个周期之后。该代码块中，作为多重实例调用的块名称，也将存储在静态局部数据中。	函数块
常量	常量	在块中使用且带有声明符号名的常量。	函数、函数块和组织块 注： 局部常量不显示在背景数据块中。

参见

- 在程序中使用变量 (页 112)
- 关键字和保留标识符 (页 106)
- 块接口中的有效数据类型 (页 9608)
- 设置局部变量的保持性 (页 9611)
- 常量的基本知识 (页 113)
- 块调用时的参数传递 (页 91)
- 在调试过程中使用设定值 (页 9706)

9.2 声明块接口的规则

9.2.1 声明块接口的一般规则

使用块参数

在块内使用块参数时，应遵循以下规则：

- Input 参数只能读取。
- Out 参数只可写入。
- In/out 参数可读取和写入。

说明

每个块的最大参数数量

在所有常规应用中，形参所允许的最大数量已足够。精确数量则取决于所选择的数据类型、声明子节和其它因素。

如果超出最大数量，则在编译时将收到一条消息。此时，可将多个参数组合为一个 PLC 数据类型 (UDT) 或组合为一个全局数据块 (DB)，作为块参数进行传递。

有关调用块时结构传递的更多信息，敬请访问以下 FAQ：

S7-1500 中进行块调用时，为何使用一个整体进行参数传递，而非许多单个元素？

 <https://support.industry.siemens.com/cs/ww/en/view/67585079> (<https://support.industry.siemens.com/cs/cn/zh/view/67585079>)

将默认值赋给块参数

可以将默认值赋给函数块接口中的特定参数。赋值的可能性取决于声明区域和参数的数据类型。如果未指定默认值，则使用既定数据类型的预定义值。例如，BOOL 类型的预定义值为“false”。

下表列出了可以赋默认值的参数：

参数类型	区域	可指定默认值		
		基本数据类型	结构化数据类型	参数类型
Input 参数	Input	√	√	-
Output 参数	Output	√	√	-
In/out 参数	InOut	√	- ⁽¹⁾	-
静态局部数据	Static	√	√	-

9.2 声明块接口的规则

参数类型	区域	可指定默认值		
		基本数据类型	结构化数据类型	参数类型
临时局部数据	Temp	-	-	-
常量	Constant	√	-	-

(1) 例外：在优化访问的块中，特定条件下可以选择使用 PLC 数据类型作为默认值。

说明

有关变量名称中可用字符的更多信息，敬请访问西门子工业在线支持网站中的以下条目：
在 STEP 7 (TIA Portal) 中，标识符和操作数应在何时加“引号”？
 <https://support.industry.siemens.com/cs/ww/en/view/109477857> (<https://support.industry.siemens.com/cs/cn/zh/view/109477857>)
有关项目级变量统一命名的建议信息，请参见“编程样式指南”：
 <https://support.industry.siemens.com/cs/ww/en/view/81318674> (<https://support.industry.siemens.com/cs/cn/zh/view/81318674/en>)

说明

调用和寻址命名空间中的程序元素

在程序代码中，命名空间中的程序元素使用符合 IEC 标准的格式来表示：

- 程序元素的名称不加引号。
- 命名空间位于程序元素名称之前，二者用点号分隔。

有关命名空间格式的详细信息，请参见“对命名空间中的程序元素进行分类 (页 9503)”

参见

- 在程序中使用变量 (页 112)
- 关键字和保留标识符 (页 106)
- 声明预定义的实参 (页 9626)

9.2.2 块接口中的有效数据类型 (S7-300, S7-400)

S7-300/400 中块接口的有效数据类型

下表列出了接口各区域可分配的参数数据类型。

区域	标准 数据类型	ARRAY STRUCT STRING DT	参数类型	VOID	POINTER	ANY
组织块						
Temp	√	√	-	-	-	√
常量	√	√ ⁽³⁾	-	-	-	-
函数块						
Input	√	√	√	-	√	√
Output	√	√	-	-	-	-
InOut	√	√ ⁽¹⁾	-	-	√	√
Static	√	√	-	-	-	-
Temp	√	√	-	-	-	√
常量	√	√ ⁽³⁾	-	-	-	-
函数						
Input	√	√ ⁽¹⁾	√	-	√	√
Output	√	√ ⁽¹⁾	-	-	√	√
InOut	√	√ ⁽¹⁾	-	-	√	√
Temp	√	√	-	-	-	√
返回	√	√	-	√	√	√ ⁽²⁾
常量	√	√ ⁽³⁾	-	-	-	-
⁽¹⁾ 只能将 STRING 定义为 254 个字符的标准长度。 ⁽²⁾ 在 SCL 中，ANY 不允许作为函数值。 ⁽³⁾ 不允许使用数据类型为 ARRAY 或 STRUCT 的常量。						

9.2 声明块接口的规则

参见

有效数据类型概述 (页 255)

块调用时的参数传递 (页 91)

9.2.3 块接口中的有效数据类型 (S7-1200, S7-1500)

S7-1200 中块接口的有效数据类型

下表列出了接口各区域可分配的参数数据类型。

区域	标准 数据类型	ARRAY STRUCT STRING / WSTRING DT	ARRAY [*]	VOID	VARIANT
组织块					
Temp	✓	✓ ⁽⁵⁾	-	-	✓
常量	✓	✓ ^{(1) (2)}	-	-	-
函数块					
Input	✓	✓	-	-	✓
Output	✓	✓	-	-	-
InOut	✓	✓ ⁽¹⁾	✓ ⁽⁴⁾	-	✓
Static	✓	✓	-	-	-
Temp	✓	✓ ⁽⁵⁾	-	-	✓
常量	✓	✓ ^{(1) (2)}	-	-	-
函数					
Input	✓	✓ ⁽¹⁾	✓ ⁽⁴⁾	-	✓
Output	✓	✓ ⁽¹⁾	✓ ⁽⁴⁾	-	✓
InOut	✓	✓ ⁽¹⁾	✓ ⁽⁴⁾	-	✓
Temp	✓	✓ ⁽⁵⁾	-	-	✓
返回	✓	✓ ⁽³⁾	-	✓	-

区域	标准数据类型	ARRAY STRUCT STRING / WSTRING DT	ARRAY [*]	VOID	VARIANT
常量	✓	✓ (1) (2)	-	-	-

(1) 不能在这些区域中声明 STRING 和 WSTRING 的长度。不允许声明 MyString[3] 格式。在这些区域中，仅允许在具有优化访问权的块中声明 WSTRING。

(2) 不允许使用数据类型为 ARRAY 或 STRUCT 的常量。

(3) WSTRING 数据类型的函数值不得超过 1022 个字符。

(4) 固件版本 V4.2 及更高版本中，ARRAY[*] 可用于具有优化访问的块中。

(5) 要快速访问“Temp”部分声明的结构化数据类型（如，ARRAY、STRUCT、PLC 数据类型等），不要访问整个结构，而是访问单独的基本结构元素。

S7-1500 中块接口的有效数据类型

下表列出了接口各区域可分配的参数数据类型。

区域	标准数据类型	ARRAY STRUCT STRING / WSTRING DT	ARRAY[*]	参数类型	VOID	DB_ANY	POINTER	ANY	VARIANT	REF_TO	NVT ⁽¹⁰⁾
组织块											
Temp	✓	✓ (9)	-	- (4)	-	✓	-	✓ (3)	✓	✓	✓
常量	✓	✓ (1) (5)	-	-	-	-	-	-	-	-	-
函数块											
Input	✓	✓	-	✓	-	✓	✓	✓	✓	-	✓
Output	✓	✓	-	-	-	✓	-	-	-	-	✓
InOut	✓	✓ (1)	✓ (7)(8)	- (4)	-	✓	✓	✓	✓	-	✓
Static	✓	✓	-	-	-	✓	-	-	-	-	✓
Temp	✓	✓ (9)	-	- (4)	-	✓	-	✓ (3)	✓	✓	✓
常量	✓	✓ (1) (5)	-	-	-	-	-	-	-	-	-

9.2 声明块接口的规则

区域	标准数据类型	ARRAY STRUCT STRING / WSTRI NG DT	ARRA Y[*]	参数 类型	VOID	DB_A NY	POINT ER	ANY	VARIA NT	REF_T O	NVT ⁽¹⁰⁾)
函数											
Input	✓	✓ ⁽¹⁾	✓ ⁽⁷⁾	✓	-	✓	✓	✓	✓	✓	✓
Output	✓	✓ ⁽¹⁾	✓ ⁽⁷⁾	-	-	✓	✓	✓	✓	✓	✓
InOut	✓	✓ ⁽¹⁾	✓ ⁽⁷⁾	- ⁽⁴⁾	-	✓	✓	✓	✓	-	✓
Temp	✓	✓ ⁽⁹⁾	-	- ⁽⁴⁾	-	✓	-	✓ ⁽³⁾	✓	✓	✓
返回	✓	✓ ⁽⁶⁾	-	-	✓	✓	✓	✓ ⁽²⁾	-	✓	✓
常量	✓	✓ ^{(1) (5)}	-	-	-	-	-	-	-	-	-
<p>(1) 不能在这些区域中声明 STRING 和 WSTRING 的长度。不允许声明 MyString[3] 格式。在这些区域中，仅允许在具有优化访问权的块中声明 WSTRING。</p> <p>(2) 在 SCL 中，ANY 不允许作为函数值。</p> <p>(3) ANY 只能用于“Temp”区域中的标准访问块。</p> <p>(4) “INSTANCE”参数类型是“TEMP”和“InOut”区域中唯一允许出现的例外。</p> <p>(5) 不允许使用数据类型为 ARRAY 或 STRUCT 的常量。</p> <p>(6) WSTRING 数据类型的函数值不得超过 1022 个字符。</p> <p>(7) 固件版本 V2.0 及更高版本中，ARRAY[*] 可用于具有优化访问的块中。</p> <p>(8) 仅当未设置属性 “在 IDB 中设置”(Set in IDB) 时，ARRAY[*] 才可用。</p> <p>(9) 要快速访问“Temp”部分声明的结构化数据类型（如，ARRAY、STRUCT、PLC 数据类型等），不要访问整个结构，而是访问单独的基本结构元素。</p> <p>(10) 命名值数据类型 (NVT) 仅在软件单元中可用。</p>											

参见

- 有效数据类型概述 (页 255)
- 块调用时的参数传递 (页 91)

9.2.4 设置局部变量的保持性

简介

函数块将其数据存储在一个背景中。要防止在发生电源故障时数据丢失，可以将数据标记为保持性。此类数据存储保持在保持性存储区中。设置保持性的选项取决于函数块的访问设置类型。

一般访问块的保持性特性

对于一般访问的数据块，无法设置各变量的保持性特性。只能在所分配的背景中将其定义为具有保持性。该块包含的所有变量随后会被视为具有保持性。

优化块访问的保持性

对于可优化访问的数据块，可以定义各个变量的保持性。

对于结构化数据类型的变量，保持性设置将始终应用于整个结构。无法对结构中的各个元素进行任何单独的保持性设置。

不能在“InOut”部分中创建结构化数据类型的保持性变量。ARRAY、STRUCT 或 STRING 等结构化数据类型的输入/输出参数通常是非保持性的。

可使用下列设置：

- 保持性
即使在发生电源故障后，变量或结构的值仍存在。
- 非保持性
若发生电源故障，变量或结构的值丢失。
- IDB 设置
可为背景数据块设置保持性。然而，在背景数据块中设置而非应用的设置，对于通过“IDB 设置”选择的所有变量而言，是基本的设置。

参见

局部变量和局部常量的属性 (页 9654)

块访问的基本知识 (页 63)

9.3 选择块接口的表格式或文本式表示方式

块接口的表格或文本表示

系统默认，接口以表格视图显示。在 SCL 中，接口也可显示为文本视图。

SCL 块的接口表示方式可统一设置。如果选择文本表示法作为默认表示法，则之后所有新建块的接口都将显示在文本视图中。

在文本视图中，只能使用以下元素：

- 各区域之间的注释信息或某一区域的开头处的注释信息
- 一种类型的多个区域（如 VAR_INPUT）。

说明

故障安全块和系统块

对于故障安全块和系统块，该接口仅显示为表格格式。

设置新块的接口显示方式

要组态新块的表示方式，请按以下步骤操作：

1. 在“选项”(Options) 菜单中，选择“设置”(Settings) 命令。
“设置”(Settings) 窗口将显示在工作区中。
2. 在区域导航中，选择“PLC 编程 > SCL”(PLC programming > SCL) 组。
3. 在“新块的默认设置 > 块接口”(Default settings for new blocks > Block interface) 中，为接口选择指定视图。

创建新块时，该块的接口将以所选视图形式显示。

说明

导入某个块后的接口显示

导入某个块后，该块的接口将显示在表格视图中。

参见

块接口的布局 (页 9613)

文本块接口的结构 (页 9631)

9.4 声明块接口

9.4.1 块接口的布局


块接口的布局

下图显示了块接口的结构。具体的列数和区域取决于块类型。

名称	数据类型	默认值	保持性
▼ Input			
■ My_Input1	Bool	false	未保留
▼ Output			
■ My_Output1	Bool	1	保留
▼ InOut			
■ <Hinzufügen>			
► Static			
► Temp			
▼ Constant			
■ My_Constant1	Int	22	

各列的含义

下表列出了各列的含义。可根据需要显示或隐藏各列。显示的列数取决于 CPU 系列和打开对象的类型。

列	说明
	单击该符号，可将元素拖放到程序中作为操作数。
名称	元素的名称。
数据类型	元素的数据类型。
偏移	块中变量的相对地址。仅标准访问的块中，才显示该列。 注： SIMATIC 系统库中的许多指令都具有“优化块访问”属性，因此不占用任何固定存储器地址。即使将这些指令用作标准访问块中的多重实例，这些指令也不显示偏移量。

9.4 声明块接口

列	说明
默认值	<p>代码块接口中可预分配给特定变量的值，或局部常量的值。</p> <p>对于变量，可选择是否指定默认值。如果未指定任何值，则使用指定数据类型的预定义值。例如，BOOL 类型的预定义值为“false”。</p> <p>在相应的背景数据块中，将变量的默认值应用为初始值。在背景数据块中，可使用实例特定的起始值对这些值进行替换。</p> <p>而在块接口中常量通常为声明的默认值。这些常量不显示在背景数据块中，也无法在此指定实例特定的值。</p>
保持性	<p>将变量标记为具有保持性。</p> <p>即使在关断电源后，保持性变量的值也将保留不变。</p> <p>该列仅在具有优化访问特性的函数块接口中显示。</p>
在 HMI 工程组态中可见	指示 HMI 选择列表中是否默认显示变量。
从 HMI/OPC UA/Web API 可访问	指示在运行过程中 HMI/OPC UA/Web API 是否可访问该变量。
从 HMI/OPC UA/Web API 可写	指示在运行过程中是否可从 HMI/OPC UA/Web API 写入变量。
设定值	<p>将变量标记为设定值。设定值是指调试过程中可能需要进行微调的值。</p> <p>该列仅适用于函数块的接口中。</p>
监控	指示该变量的过程诊断是否创建有监视。
注释	说明元素的注释信息。

参见

- 在程序中使用变量 (页 112)
- 关键字和保留标识符 (页 106)
- 块接口中的有效数据类型 (页 9608)
- 设置局部变量的保持性 (页 9611)
- 常量的基本知识 (页 113)
- 选择块接口的表格式或文本式表示方式 (页 9612)
- 块调用时的参数传递 (页 91)
- 在调试过程中使用设定值 (页 9706)

9.4.2 在块接口中声明局部变量和局部常量

要求

块接口已打开。

操作步骤

要声明一个变量或常量，请按以下步骤操作：

1. 在块接口中，选择需声明的部分。
常量必须在“Constant”部分进行声明。
2. 在“名称”(Name) 列中，输入元素名称。
3. 在“数据类型”列中输入所需的数据类型。在输入过程中，系统将提供自动完成功能。
4. 对于常量，在“默认值”(Default value) 列中输入一个值。
5. 可选：更改在块接口其它列中显示的属性。

结果

创建该元素。

语法检查

每次输入后都会执行语法检查，并且找到的任何错误都将以红色显示。无需立即更正这些错误，可以继续编辑并在以后更正所有错误。但是，如果变量声明包含语法错误，将无法编译程序。

说明

如果更改了块接口，则程序中的块调用可能变得不一致。若可能，系统会自动更新调用位置。如果无法自动更新，则必须手动更新那些不一致的块。

另请参见

在 LAD 中更新块调用 (页 9801)

在 FBD 中更新块调用 (页 9863)

参见

声明块接口的规则 (页 9605)

在程序中使用变量 (页 112)

常量 (页 113)

9.4 声明块接口

- 声明 ARRAY 数据类型的变量 (页 9617)
- 声明 STRUCT 数据类型的变量 (页 9619)
- 声明 STRING 和 WSTRING 数据类型的变量 (页 9621)
- 声明基于 PLC 数据类型的变量 (页 9621)
- 声明更高级别的变量 (页 9622)
- 声明多重背景 (页 9623)
- 声明参数实例 (页 9624)
- 有关起始值的基本信息 (页 9677)
- 设置局部变量的保持性 (页 9611)
- 局部变量和局部常量的属性 (页 9654)

9.4.3 在程序编辑器中声明局部变量

要求

程序编辑器已打开。

步骤

要声明局部变量，请按以下步骤操作：

1. 在程序中插入指令。
“<???”、“<??.?>”或“...”字符串表示操作数的占位符。
2. 使用要创建的变量的名称替换操作数占位符。
3. 选择元素的名称。
若要声明多个元素，请为要声明的所有元素选择名称。
4. 在快捷菜单中，选择“定义变量”命令。
将打开“定义变量”对话框。显示已输入元素名称的声明表。
5. 要声明局部变量，请选择以下区域之一：
 - 局部输入
 - 局部输出
 - 局部输入/输出
 - 局部静态
 - 局部临时

6. 在其它列中，输入数据类型和注释。
7. 单击“定义”按钮完成输入。

结果

声明将直接写入块接口并在整个块内有效。

说明

如果更改了块接口，则程序中的块调用可能变得不一致。若可能，系统会自动更新调用位置。如果无法自动更新，则必须手动更新那些不一致的块。

另请参见：

在 LAD 中更新块调用 (页 9801)

在 FBD 中更新块调用 (页 9863)

参见

在程序中使用变量 (页 112)

关键字和保留标识符 (页 106)

有关起始值的基本信息 (页 9677)

局部变量和局部常量的属性 (页 9654)

设置局部变量的保持性 (页 9611)

编辑块接口的表格 (页 9627)

声明块接口的规则 (页 9605)

9.4.4 声明 ARRAY 数据类型的变量

要求

块接口已打开。

操作步骤

要声明一个 ARRAY 数据类型的变量，请按以下步骤操作：

1. 在块接口中，选择需声明的部分。
2. 在“名称”(Name) 列中，输入变量的名称。

9.4 声明块接口

- 3. 在“数据类型”列中，单击数据类型选择按钮。
将打开可选用的数据类型的列表。
- 4. 选择“Array”数据类型。
“数组”(Array) 对话框随即打开。
- 5. 在“数据类型”(Data type) 文本框中，指定 ARRAY 元素的数据类型。
- 6. 在“ARRAY 限值”(ARRAY limits) 输入字段中，指定各维度的上限和下限值。
一维 ARRAY 的示例：
Array[0..3] of Bool
三维 ARRAY 的示例：
Array[0..3, 0..15, 0..33] of Bool
带有可变限值的一维 ARRAY 示例：
Array[*] of Bool
带有可变限值的三维 ARRAY 示例：
Array[*, *, *] of Bool
局部常量作为 ARRAY 限值时，一维 ARRAY 的示例：
Array[#My_local_const1..#My_local_const2] of Bool
全局常量作为 ARRAY 限值时，一维 ARRAY 的示例：
Array["My_global_const1".."My_global_const1"] of Bool
- 7. 确认输入。
- 8. 可选：更改将在块接口其它列中显示的变量属性。

这时变量已创建完成，但保持折叠状态。要展开 ARRAY，请单击变量前的三角形标记。请注意，为了清楚起见，不能展开包含超过 10000 个元素的 ARRAY。

说明

ARRAY[*] 的可用性

对于固件版本 V4.2 及更高版本的 S7-1200 系列 CPU 以及固件版本 V2.0 及更高版本的 S7-1500 系列 CPU，ARRAY[*] 可用于优化块中。在函数 (FC) 中，可在所有声明子区域中使用 ARRAY[*]。在函数块 (FB) 中，ARRAY[*] 只能声明为“InOut”区域中的 in-out 参数。

输入 ARRAY 元素的默认值

要为 ARRAY 中各元素预分配默认值，请按以下步骤操作：

- 1. 单击 ARRAY 数据类型变量前的三角形符号。
该 ARRAY 随即打开，且各 ARRAY 元素在单独的行中显示。
- 2. 在“默认值”(Default value) 列中，输入所需的值。

为 ARRAY 添加注释

要为整个 ARRAY 添加注释，可在 ARRAY 声明的最上面一行中输入注释信息。该注释信息将用作所有低层级 ARRAY 元素的预设注释。

要为 ARRAY 的各元素指定具体的注释信息，请按以下步骤操作：

1. 单击 ARRAY 数据类型变量前的三角形符号。
该 ARRAY 随即打开，且各 ARRAY 元素在单独的行中显示。
2. 在“注释”(Comment) 列中，输入指定值。

在扩展模式下显示 ARRAY

在“扩展模式”下，所有 ARRAY 默认展开显示。

要启用扩展模式，请按下列步骤操作：

1. 单击工具栏中的“扩展模式”(Expanded mode) 按钮。

说明

“扩展模式”显示的限制

根据计算机上可用的 RAM 大小，扩展模式下的最大限值为：

- RAM < 8 GB：最多可显示 10000 行。
 - RAM ≥ 8 GB：最多可显示 40000 行。
-

参见

ARRAY (页 322)

声明块接口的规则 (页 9605)

在程序中使用变量 (页 112)

关键字和保留标识符 (页 106)

局部变量和局部常量的属性 (页 9654)

设置局部变量的保持性 (页 9611)

9.4.5 声明 STRUCT 数据类型的变量

要求

块接口已打开。

9.4 声明块接口

操作步骤

要声明 STRUCT 数据类型的变量，请按以下步骤操作：

1. 在块接口中，选择需声明的部分。
2. 在“名称”(Name) 列中，输入变量的名称。
3. 在“数据类型”(Data type) 列中，输入“Struct”。输入时系统将提供自动完成功能。
在新变量后将插入空的缩进行。
4. 在第一个空行中插入第一个结构元素。
在该元素后会再插入一个空行。
5. 选择结构元素的数据类型。
6. 可选：更改在块接口其它列中显示的结构元素的属性。
7. 对于所有其它结构元素，重复执行步骤 4 到 6。
不必显式完成该结构。结构会以输入的最后一个元素结束。
8. 要在结构后插入新变量，请在结构结尾后留一个空行，然后在第二个空行中启动新变量。

结果

创建数据类型 STRUCT 的变量。

说明

S7-1200/S7-1500：一个数据块中最多包含 252 个结构 (STRUCT)

S7-1200/S7-1500 系列 CPU 中，一个块中最多支持 252 个结构 (STRUCT)。如需使用更多结构，则可使用 PLC 数据类型 (UDT) 代替“STRUCT”数据类型。

参见

STRUCT 声明 (页 9636)

声明块接口的规则 (页 9605)

在程序中使用变量 (页 112)

关键字和保留标识符 (页 106)

局部变量和局部常量的属性 (页 9654)

设置局部变量的保持性 (页 9611)

9.4.6 声明 STRING 和 WSTRING 数据类型的变量

要求

块接口已打开。

操作步骤

要声明 STRING 或 WSTRING 数据类型的变量，请按以下步骤操作：

1. 在块接口中，选择需声明的部分。
2. 在“名称”(Name) 列中输入一个变量名称。
3. 在“数据类型”(Data type) 列中输入“STRING”或“WSTRING”。输入时系统将提供自动完成功能。
4. 可选：在关键字 STRING 或 WSTRING 后面，使用方括号指定字符串的最大长度。如果未指定最大长度，则字符串的默认长度为 254 个字符。
最大长度为 4 的 WSTRING 示例：
WSTRING[4]
使用局部常量定义字符串最大长度的示例：
STRING[#My_local_const1]
使用全局常量定义字符串最大长度的示例：
STRING["My_global_const1"]

结果

应用 STRING 或 WSTRING 数据类型的变量。

参见

STRING 和 WSTRING 声明 (页 9645)

9.4.7 声明基于 PLC 数据类型的变量

要求

PLC 数据类型在当前 CPU 中进行声明。

9.4 声明块接口

操作步骤

要声明基于 PLC 数据类型的变量，请按以下步骤操作：

1. 在块接口中，选择需声明的部分。
2. 在“数据类型”列中输入 PLC 数据类型。在输入过程中，系统将提供自动完成功能。创建该变量。
3. 如果在声明 PLC 数据类型过程中已为 PLC 数据类型中的变量定义了默认值或注释，则这些默认值或注释将灰显。这些值可在块接口中更改。更改后的值将显示为黑色，且仅应用于特定的使用点。

说明

若更新或删除在块接口中使用的 PLC 数据类型，则接口会变得不一致。要消除这种不一致性，必须重新编译程序。

另请参见“更新块接口 (页 9660)”

9.4.8 声明更高级别的变量

简介

要访问声明变量内的数据区域，则可以通过附加声明来覆盖所声明的变量。这样，可以选择对不同数据类型的已声明变量进行寻址。例如，可以使用 BOOL 的 ARRAY 对 WORD 数据类型变量的各个位寻址。

覆盖变量

要覆盖具有新数据类型的变量，请按以下步骤操作：

1. 打开块接口。
2. 在接口中，选择要覆盖的具有新数据类型的变量。
3. 单击工具栏上的“添加行”按钮。
在要覆盖的变量之后插入一行。覆盖变量必须直接在行中被覆盖的变量之后声明。
4. 在“名称”(Name) 列中，输入变量的名称。
5. 在“数据类型”列中输入“AT”条目。在这一步骤中，系统将提供自动完成功能。
将以下内容添加到“名称”列的条目中。
“AT<Name of the higher-level tag>”
6. 再次单击数据类型选择按钮，为新变量选择数据类型。
创建该变量。它指向与更高等级变量相同的数据，但是会为该数据设定新的数据类型。

删除覆盖

要删除变量的覆盖，请按以下步骤操作：

1. 在块接口中，选择较高级别的变量。
2. 在快捷菜单中，选择“删除”命令。

覆盖删除。

参见

使用 AT 覆盖变量 (页 129)

9.4.9 声明多重背景

要求

要调用的函数块存在于项目树中，并且具有多重实例功能。

声明多重实例

要将待调用的函数块声明为多重实例，请按以下步骤操作：

1. 打开调用函数块的块接口。
2. 在“静态”区域的“名称”列中，输入块调用的名称。
3. 在“数据类型”列中，输入要调用的函数块的符号名称。

说明

当在程序段中编写块调用时，程序编辑器将自动声明多重实例，然后在“调用选项”话框中指定要将该块作为多重实例调用。

将多重实例声明为 ARRAY

要声明多重实例的 ARRAY，请按以下步骤操作：

1. 打开调用函数块的块接口。
2. 在“静态”(Static) 区域的“名称”(Name) 列中，输入 ARRAY 的名称。
3. 在“数据类型”(Data type) 列中，选择 ARRAY 数据类型。
“数组”(Array) 对话框随即打开。
4. 在“数据类型”(Data type) 输入字段中，输入应定义实例的函数块名称。块名称必须用引号括起。

9.4 声明块接口

- 5. 在“ARRAY 限值”(ARRAY limits) 输入字段中，指定各维度的上限和下限值。
一维 ARRAY:[0..3] 的示例：
三维 ARRAY:[0..3, 0..15, 0..33] 的示例
- 6. 确认输入。

参见

更新块接口 (页 9660)
实例声明 (页 9646)

9.4.10 声明参数实例

简介

下表列出了块接口中的声明方式和使用的参数实例：

声明	块类型	应用
InOut	FB FC	之后，可调用在函数块中作为实参传送的实例。
Input	FC	用户读取作为实参传送的实例数据，但无法在块中调用该实例。
Output	FC	用户可写入作为实参传送的实例数据，但无法在块中调用该实例。

要求

块接口已打开。

操作步骤

- 要声明参数实例，请按以下步骤操作：
- 1. 打开块接口的指定区域（参见表格）。
 - 2. 在 “名称”(Name) 列中，输入实例待传送的 in/out 参数名称。
 - 3. 在 “数据类型”(Data type) 列中，输入应定义实例的函数块名称。块名称必须用引号括起。

说明

自动声明参数实例

在程序段中编程一个块调用时，程序编辑器会自动声明参数实例，然后在“调用方式”(Call options) 对话框中指定将该块作为参数实例调用。

参见

参数实例 (页 77)

实例声明 (页 9646)

9.4.11 在调用块过程中隐藏参数

简介

在 LAD 或 FBD 中调用该块时，可隐藏块参数。刚开始时，隐藏的参数不可见；但可通过框底部边缘处的小箭头进行显示。

可通过以下两种方式隐藏参数：

- 隐藏
该参数始终隐藏。
- 未分配参数时隐藏
只要参数未互连则隐藏如果指定一个实参，则该参数将显示在调用框中。

操作步骤

要定义在块调用中显示或隐藏块参数，请按以下步骤操作：

1. 打开块接口。
2. 在块接口中选择一个参数。
3. 在巡视窗口中，打开“属性”(Properties) 选项卡。

9.4 声明块接口

4. 在区域导航中选择“属性”(Attributes) 组。
5. 在“在 LAD / FBD 的块调用中显示”(Visibility at block calls in LAD / FBD) 中，选择以下一个选项：
 - “显示”
 - “隐藏”
 - “未分配参数时隐藏”

结果

根据设置，在 LAD 或 FEB 的块调用中显示或隐藏参数。

如果在块调用中发现语法错误，则参数无法隐藏。

参见

连接隐藏参数 (页 9817)

声明预定义的实参 (页 9626)

9.4.12 声明预定义的实参

简介

在块参数的声明过程中，可指定块调用时所用的实参。

如果正在使用程序块作为库元素并且希望存储要与库元素一起使用的实参的信息，这将非常有用。

此外，在 LAD 和 FBD 中还可隐藏块调用中具有有效预定义实参的参数。刚开始时，隐藏的参数不可见；但可通过框底部边缘处的小箭头进行显示。

可通过以下两种方式隐藏预定义的参数：

- 该参数始终隐藏。
- 只要未指定预定义的实参，则该参数将隐藏。但如果指定了不同的参数，则该参数将显示可见。

要求

位于“Input”、“Output”或“InOut”部分的参数。

操作步骤

要预定义实参，请按以下步骤操作：

1. 打开块接口。
2. 在块接口中选择一个参数。
3. 在巡视窗口中，打开“属性”(Properties) 选项卡。
4. 在区域导航中选择“属性”(Attributes) 组。
5. 在“预定义实参”(Predefined actual parameter) 输入框中输入所需的实参。
6. 在“在 LAD / FBD 的块调用中显示”(Visibility at block calls in LAD / FBD) 中，选择选“隐藏”(Hide)。
7. 在附加选项中，选择“如果块调用时指定的参数与预定义的实参不同则显示”(Show if parameter assigned on block call is not identical to the predefined actual parameter)。仅当事先激活了“隐藏”(Hide) 选项时，该选项才可用。

结果

- 实参被预定义。如果将程序块保存为库元素，它还会包含有关待使用实参的信息。
- 如果在程序中使用该库元素，则会检查预定义的实参是否可以在程序寻址。如果可能的话，它将会自动用作实参。
- 如果程序中找不到实参，则将发出一个语法错误，此时需手动进行参数分配。分配无效的参数无法隐藏。

参见

在调用块过程中隐藏参数 (页 9625)

9.4.13 编辑块接口的表格

9.4.13.1 插入表格行

步骤

请按以下步骤操作，在所选行上面插入一行：

1. 选择要在其上插入新行的行。
2. 在表格的工具栏上，单击“插入行”按钮。

9.4 声明块接口

结果

将在所选行上方插入一个新行。

9.4.13.2 插入表格行

步骤

要在所选行下面插入一行，请按以下步骤操作：

1. 选择要在其下插入新行的行。
2. 单击表工具栏上的“添加行”按钮。

结果

将在选定行的下方插入一个新的空行。

9.4.13.3 删除变量

操作步骤

要删除变量，请按以下步骤操作：

1. 选择待删除变量所在行。也可以在按住 <Ctrl> 键时依次单击行，或者按住 <Shift> 键单击第一行和最后一行来选择多行。
2. 在快捷菜单中，选择“删除”命令。

9.4.13.4 自动填充连续单元格

可以将一个或多个表格单元格的内容装载到下面的单元格，将自动连续填充单元格。

如果自动填充“名称”列中的单元格，将在每个名称后面附加一个连续的编号。例如，“Motor”将变为“Motor_1”。

可以将单个或多个单元格以及整行定义为源区域。

如果所打开表格中的行数少于要填充的行数，则必需首先插入所缺少的空行。

要求

- 表格已打开。
- 有足够的声明行。

步骤

要自动填充连续单元格，请按以下步骤操作：

1. 选择要装载的单元格。
2. 单击该单元格右下角的“填充”符号。
鼠标指针将变为十字形。
3. 按住鼠标按键并向下拖动鼠标指针扫过要自动填充的单元格。
4. 释放鼠标按键。
单元格自动填充。
5. 如果在待自动填充的单元格中已经存在有条目，则将显示一个对话框。在该对话框中，可以指出是否要覆盖现有的条目，还是为新变量插入新行。

9.4.13.5 显示和隐藏表列

可根据需要显示或隐藏表中的列。

步骤

要显示或隐藏表格的列，请按以下步骤操作：

1. 单击列标题。
2. 在快捷菜单中，选择“显示/隐藏”(Show/Hide) 命令。
显示可以选择的列。
3. 要显示列，选中该列的复选框。
4. 要隐藏列，清除选中该列的复选框。
5. 要隐藏或显示多个列，请单击“更多”(More) 并激活或禁用“显示/隐藏”(Show/Hide) 对话框中相应列的复选框。

9.4.13.6 使用外部编辑器编辑变量

要在外部表格编辑器中编辑单个变量，可通过复制和粘贴操作对这些变量进行导入/导出。

说明

显示和复制 ARRAY 元素时的限制条件

根据计算机上可用的工作存储器空间大小，在 Excel 表格中显示和复制块接口中的 ARRAY 时最大限值为：

- RAM < 8 GB：最多可显示和复制 10000 个元素。
 - RAM ≥ 8 GB：最多可显示和复制 40000 个元素。
-

9.4 声明块接口

要求

打开块接口和外部编辑器。

操作步骤

要将变量导出到外部编辑器并再次导入变量，请按以下步骤操作：

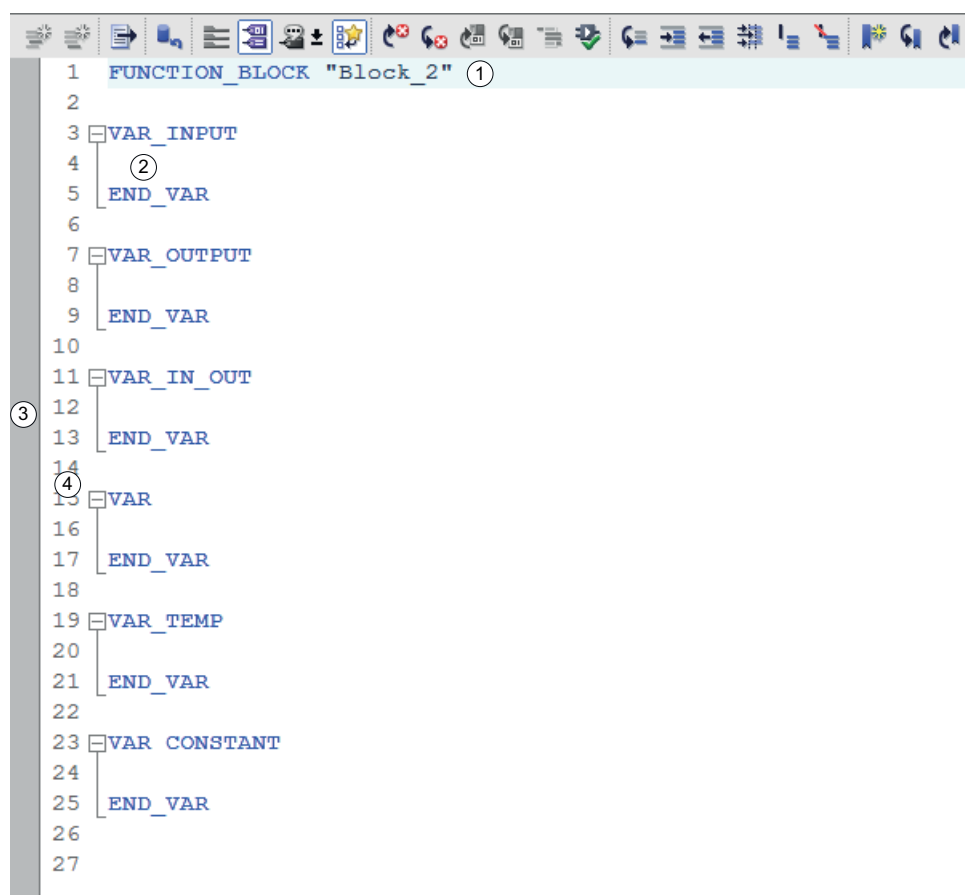
1. 单击“扩展模式”(Expanded mode) 按钮，显示结构化数据类型中的所有元素。
2. 选择一个或多个变量。
3. 在快捷菜单中，选择“复制”。
4. 切换到外部编辑器，并粘贴复制的变量。
5. 根据需要编辑变量。
6. 在外部编辑器中复制变量。
7. 在外部编辑器中选择变量。
8. 切换回块接口。
9. 在快捷菜单中，选择“粘贴”(Paste)。

9.5 在文本视图中声明块接口

9.5.1 文本块接口的结构

文本块接口的结构

下图显示了文本式块接口的结构，与创建新块后的显示相同。



文本块接口由以下几个区域组成：

9.5 在文本视图中声明块接口

区域	含义
① 块声明	在块声明中，定义块的名称与类型
② 声明部分	在声明部分，声明该块的参数和局部数据。声明部分已输入到空白块接口中。但用户可根据需要进行移动、复制或删除各部分。
③ 侧栏	在侧栏中，可设置书签。
④ 行号	行号将显示在声明部分的左侧。

参见

选择块接口的表格式或文本式表示方式 (页 9612)

9.5.2 块声明和返回值

块声明

文本块接口通常以块声明开始。用户无需自己输入块声明，块声明已包含在接口模板中。

返回值

在函数 (FC) 中，可在声明后指定该函数块的返回值。返回值为返回给调用块的值。如果指定“VOID”值，该函数不返回任何值。

语法

块声明和返回值使用的语法如下：

块类型	语法	示例
函数 (FC)	FUNCTION <名称> [:返回值数据类型] <声明>	FUNCTION "My_Function" : Int FUNCTION "My_Function" : Void
函数块 (FB)	FUNCTION_BLOCK <名称> <声明>	FUNCTION_BLOCK "My_FunctionBlock"
组织块 (OB)	ORGANIZATION_BLOCK <名称> <声明>	ORGANIZATION_BLOCK "My_OrganizationBlock"

参见

文本块接口示例 (页 9652)

9.5.3 声明部分

声明部分

文本块接口分为不同的声明部分，各声明部分的特征通过关键字对描述。允许使用不同部分，具体视块类型而定。

声明部分的顺序不重要。一个部分可在块接口中出现多次。

在下表中，列出了声明部分使用的语法结构：

声明部分	语法
Input 参数	VAR_INPUT [<ATTRIBUTE> <声明> END_VAR
Output 参数	VAR_OUTPUT [<ATTRIBUTE> <声明> END_VAR

9.5 在文本视图中声明块接口

声明部分	语法
In/out 参数	VAR_IN_OUT [<ATTRIBUTE>] <声明> END_VAR
临时局部数据	VAR_TEMP <声明> END_VAR
静态局部数据	VAR [<ATTRIBUTE>] <声明> END_VAR
常量	VAR CONSTANT <声明> END_VAR

有关各声明部分的信息，请参见：
块接口概述 (页 9603)

声明部分的属性

可选择为各部分分配属性。属性会应用到此部分中声明的所有变量。如果某一部分多次出现，每次出现时可使用其它属性。

以下属性可用：

属性	含义
RETAIN	此部分中的变量具有保持性，即电源故障后变量值仍然可用。
DB_SPECIFIC	在背景数据块中，可设置保持性。

如果未指定任何属性，则声明部分数据不具有保持性。

参见

设置局部变量的保持性 (页 9611)

文本块接口示例 (页 9652)

9.5.4 变量声明和初始化

变量或常量的声明和初始化

在各声明部分中声明块所用的局部变量和常量。

变量声明由符号名称和数据类型组成。可选择指定初始化变量的默认值。

除符号名称和数据类型外，常量声明始终包含一个数值。仅“VAR CONSTANT”部分才允许进行常量声明。

用 AT 覆盖变量

要访问声明变量内的数据部分，可以通过附加声明覆盖已声明的变量。这样，可以选择对具有不同数据类型的已声明变量进行寻址。例如，可以使用 BOOL 的 ARRAY 对 WORD 数据类型变量的各个位寻址。

虽然 AT 声明在表格视图中必需直接跟随在被引用变量之后，但在文本视图中，该声明可位于任何位置。

AT 声明不能初始化。请注意，AT 声明只能用于非优化块中。

语法

声明变量和常量所用的语法如下：

9.5 在文本视图中声明块接口

声明	语法	示例
变量	<名称> : <数据类型> [:= <值>];	myBit : BOOL; myBit : BOOL := true;
常量	<名称> : <数据类型> := <值>;	PI : REAL := 3.141592; myInt: INT := INT#16#7FFF; myString: STRING := 'hello';
AT 声明	<名称> AT <引用变量名称> : <数据类型>;	myReferenceToVar2 AT Var_2 : Int;

参见

- 有效数据类型概述 (页 255)
- 声明块接口的一般规则 (页 9605)
- 使用 AT 覆盖变量 (页 129)
- 声明更高级别的变量 (页 9622)
- 文本块接口示例 (页 9652)
- 常量 (页 113)

9.5.5 STRUCT 声明

STRUCT 声明

STRUCT 数据类型是指一种元素数量固定但数据类型不同的数据结构。

也可以基于 PLC 数据类型 (UDT) 创建结构。为此，需将 PLC 数据类型指定为一个结构化数据类型。小系统将通过工具提示，显示该 PLC 数据类型的各个元素。

语法

声明 STRUCT 所用的语法如下：

声明	语法	示例
STRUCT	<pre><名称> : Struct <元素名称> : <数据类型> [:= <值>]; <元素名称> : <数据类型> [:= <值>]; ... END_Struct := (<初始化列表>);</pre>	<pre>myStruct : Struct mem_1 : Int; mem_2 : Int; mem_3 : BOOL; END_Struct := (6,0, TRUE) ;</pre>
基于 PLC 数据类型 (UDT) 的 STRUCT	<pre><名称> : "<UDT 名称>" := (<初始化列表>);</pre>	<pre>myStruct : "myType" := (6,0, TRUE);</pre>

说明

S7-1500：一个数据块中最多有 252 个结构

S7-1500 系列 CPU 中的一个数据块中最多允许有 252 个结构。如果需要更多结构，则必须重新构造自己的程序。例如，可以在多个全局数据块中创建结构。

STRUCT 的初始化

STRUCT 初始化为可选操作。进行初始化时，可在变量声明后直接跟随一个值，也可使用初始化列表。

初始化列表位于关键字 **END_STRUCT** 之后，包含在一个括号内并通过 **:=** 进行赋值。如果该结构中还包含下级结构，则可通过括号嵌套进行初始化。如果这两种初始化方式均未采用，则可使用初始化列表中的值。

9.5 在文本视图中声明块接口

初始化列表使用的语法如下所示：

声明	语法	示例	注释
STRUCT 的初始化	<code><名称> : Struct</code> <code> <元素名称> : <数据类型> [:=</code> <code> <值>];</code> <code> ...</code> <code>End_Struct</code>	<pre>myStruct : Struct mem_1 : Int := 1; mem_2 : Int := 2; mem_3 : BOOL := FALSE End_Struct</pre>	该结构中的元素将进行如下初始化： // mem_1 := 1 // mem_2 := 2 // mem_3 := FALSE
STRUC 的初始化列表	<code>:= (<值>, <值>, <值>...)</code>	<pre>myStruct : Struct mem_1 : Int; mem_2 : Int; mem_3 : BOOL End_Struct := (2, 0, TRUE);</pre>	该结构中的元素将进行如下初始化： // mem_1 := 2 // mem_2 := 0 // mem_3 := TRUE
嵌套 STRUC 的初始化列表	<code>:= (<值>, (<值>, <值>...))</code>	<pre>myStruct : Struct mem_1 : Int; mem_2 : Int; mem_3 : Struct mem_4 : BOOL; mem_5 : BOOL; End_Struct; End_Struct := (2, 0, (TRUE, TRUE));</pre>	该结构中的元素将进行如下初始化： // mem_1 := 2 // mem_2 := 0 // mem_4 := TRUE // mem_5 := TRUE
STRUC 的初始化列表 (带元素命名规范)	<code>:= (<元素名称> := <值>), (<元素名称> := <值>)...</code>	<pre>myStruct : Struct mem_1 : Int; mem_2 : Int; End_Struct := (mem_2 := 55);</pre>	该结构中的元素将进行如下初始化： // mem_2 := 55
PLC 数据类型 (UDT) 的初始化列表	<code>:= (<值>, <值>, <值>...)</code>	<pre>myStruct : "myType" := (2, 0, TRUE);</pre>	该结构中的元素将进行如下初始化： // mem_1 := 2 // mem_2 := 0 // mem_3 := TRUE

声明	语法	示例	注释
嵌套 PLC 数据类型 (UDT) 的初始化列表	<code>:= (<值>,<值>,<值>...)</code>	<code>myStruct : "myType" := (2,0, (TRUE,TRUE);</code>	该结构中的元素将进行如下初始化: <code>// mem_1 := 2</code> <code>// mem_2 := 0</code> <code>// mem_3 := TRUE</code> <code>// mem_4 := TRUE</code>
PLC 数据类型 (UDT) 的初始化列表 (带元素命名规范)	<code>:= (<元素名称> := <值>), (<元素名称> := <值>)...</code>	<code>myStruct : "myType" := (mem_1:=22,mem_2:=55);</code>	该结构中的元素将进行如下初始化: <code>// mem_1 := 22</code> <code>// mem_2 := 55</code>

参见

声明 STRUCT 数据类型的变量 (页 9619)

STRUCT 数据结构 (匿名结构) (页 317)

声明块接口的规则 (页 9605)

文本块接口示例 (页 9652)

9.5.6 ARRAY 声明

ARRAY 声明

ARRAY 数据类型表示一个由数目固定且数据类型相同的元素组成的数据结构。

声明 ARRAY 所用的语法如下:

9.5 在文本视图中声明块接口

声明	语法	示例
ARRAY	<名称> : ARRAY [下限..上限] OF <数据类型> := [<初始化列表>];	MyARRAY_1 : ARRAY[0..7] OF BOOL; MyARRAY_1 : ARRAY[0..7] OF BOOL := [1,1,0,0,0,1,0,0];
限值可变的 ARRAY	<名称> : ARRAY [*] OF <数据类型>;	MyARRAY_1 : ARRAY[*] OF INT; MyARRAY_2 : ARRAY[* , * , *] OF INT;
局部常量作 限值的 ARRAY	<名称> : ARRAY [#<常量名称>..#<常量名称>] OF <数据类型> := [<初始化列表>];	MyARRAY_1 : ARRAY[#LocConst1..#LocConst2] OF INT; MyARRAY_2 : ARRAY[1..#LocConst] OF INT; MyARRAY_3 : ARRAY[1..#LocConst] OF INT := [1,1,0,0];
全局常量作 限值的 ARRAY	<名称> : ARRAY ["<常量名称>.."<常量名称>"] OF <数据类型> := [<初始化列表>];	MyARRAY_1 : ARRAY["GlobConst1.."GlobConst2"] OF INT; MyARRAY_2 : ARRAY[1.."GlobConst", 2..5,#1..#u] OF INT; MyARRAY_3 : ARRAY[1.."GlobConst"] OF INT:= [1,1,0,0];
STRUCT 数 据类型 ARRAY	<名称> : ARRAY[下限..上限] OF Struct <元素名称> : <数据类型>; <元素名称> : <数据类型>; ... END_Struct := [<初始化列表>];	MyARRAY_1 : Array[0..1] OF Struct mem_1 : Int; mem_2 : Int; END_STRUCT := [(2,4) , (22,44)];
UDT 数据类 型的 ARRAY	<名称> : ARRAY[下限..上限] OF "<UDT_名称>" := [<参数列表>];	MyARRAY_1 : Array[0..1] OF MyType := [(2,4) , (22,44)];

ARRAY 初始化

可选择初始化 ARRAY。对 ARRAY 进行初始化时要使用初始化列表。初始化列表位于声明后的方括号内，以 := 开头。列表包含的以逗号分隔的值将用于 ARRAY 的所有元素。最多可初始化 10000 个元素。

可选择使用括号中的重复因子，用相同值对多个连续元素进行初始化。重复因子必须是一个正整数。

对于 STRUCT 数据类型的 ARRAY，还可以选择逐个元素进行初始化。可集中为同一类型的所有结构化元素分配一个数值。

初始化列表使用的语法如下所示：

声明	语法	示例	注释
初始化列表	:= [<值>,<值>,<值>...]	<pre>myArray : ARRAY[0..2] OF BOOL := [1,1,0];</pre>	ARRAY 中的元素将按照以下方式进行初始化： // myArray[0] := 1 // myArray[1] := 1 // myArray[2] := 0
包含重复因子的初始化列表	:= [<值>,<重复因子>(值),<值>...]	<pre>myArray : ARRAY[1..2, 1..3] OF INT := [9,8,3(10),6];</pre>	ARRAY 中的元素将按照以下方式进行初始化： // myArray[1,1] := 9 // myArray[1,2] := 8 // myArray[1,3] := 10 // myArray[2,1] := 10 // myArray[2,2] := 10 // myArray[2,3] := 6

9.5 在文本视图中声明块接口

声明	语法	示例	注释
STRUCT 数据类型 ARRAY 的初始化列表	:= [(<值列表元素 1>), (<值列表元素 2>)...]	<pre> myArray : Array[0..1] OF Struct Element1 : Int; Element2 : Int; END_Struct := [(2,4), (22,44)]; </pre>	<p>ARRAY 中的元素将按照以下方式进行初始化:</p> <pre> // myArray[0].Element1 := 2 // myArray[0].Element2 := 4 // myArray[1].Element1 := 22 // myArray[1].Element2 := 44 </pre>
ARRAY of STRUCT 的 初始化列表 (带元素命名 规范)	:= [(<元素名称> := <值>), (<元素名称> := <值>)...]	<pre> myArray : Array[0..1] OF Struct Element1 : Int; Element2 : Int; END_Struct := [(Element1 := 2 , Element2:=4), (Element1 := 22, Element2:=44)]; </pre>	<p>ARRAY 中的元素将按照以下方式进行初始化:</p> <pre> // myArray[0].Element1 := 2 // myArray[0].Element2 := 4 // myArray[1].Element1 := 22 // myArray[1].Element2 := 44 </pre>

声明	语法	示例	注释
ARRAY of STRUCT 的 初始化列表 (带重复因 子)	:= [<重复因子>(值列表元素 1),< 重复因子>(值列表元素 2),...]	<pre> myArray : Array[0..1, 0..1] OF Struct Element_x : Int; Element_y : Int; End_Struct:= [2((1, 1)), 2((2, 2))]; </pre>	<p>ARRAY 中的元素将按照以下方式进行初始化:</p> <pre> // myArray[0,0].Element_x: = 1 // myArray[0,0].Element_y: = 11 // myArray[0,1].Element_x: = 1 // myArray[0,1].Element_y: = 11 // myArray[1,0].Element_x: = 2 // myArray[1,0].Element_y: = 22 // myArray[1,1].Element_x: = 2 // myArray[1,1].Element_y: = 22 </pre>

9.5 在文本视图中声明块接口

声明	语法	示例	注释
UDT 数据类型 ARRAY 的初始化列表	<code>:= [(值列表元素 1>), (<值列表元素 2>)...]</code>	<code>myArray : Array[0..1] OF MyType := [(2,4), (22,44)];</code>	ARRAY 中的元素将按照以下方式进行初始化: // myArray[0].UDT-Element1 := 2 // myArray[0].UDT-Element2 := 4 // myArray[1].UDT-Element1 := 22 // myArray[1].UDT-Element2 := 44
ARRAY of UDT 的初始化列表（带元素命名规范）	<code>:= [<元素名称> := <值>), (<元素名称> := <值>)...]</code>	<code>myArray : Array[0..1] OF MyType := [(UDT-Element1 := 2, UDT-Element2:=4), (UDT-Element1 := 22, UDT-Element2:=44)];</code>	ARRAY 中的元素将按照以下方式进行初始化: // myArray[0].UDT-Element1 := 2 // myArray[0].UDT-Element2 := 4 // myArray[1].UDT-Element1 := 22 // myArray[1].UDT-Element2 := 44
ARRAY of UDT 的初始化列表（带重复因子）	<code>:= [<重复因子>(值列表元素 1), <重复因子>(值列表元素 2),...]</code>	<code>myArray : Array[0..1] OF myType := [2(((),1))];</code>	ARRAY 中的元素将按照以下方式进行初始化: // myArray[0].UDT-Element1 := 0 // myArray[0].UDT-Element2 := 1 // myArray[1].UDT-Element1 := 0 // myArray[1].UDT-Element2 := 1

参见

ARRAY (页 322)

声明块接口的规则 (页 9605)

文本块接口示例 (页 9652)

9.5.7 STRING 和 WSTRING 声明

STRING 和 WSTRING 的声明

STRING 和 WSTRING 数据类型存储一个字符串中的多个字符。允许在字符串中使用任何 ASCII 码类型的字符。这些字符将使用一个单引号括起。

可选择指定初始化字符串的默认值。

操作数声明期间，可在关键字 STRING 或 WSTRING 后使用方括号（例如 WSTRING[4]）指定字符串的最大长度。要声明最大长度，则可输入一个绝对值或使用局部/全局常量。

如果未指定最大长度，则相应的操作数长度设置为标准的 254 个字符。

语法

声明 STRING 和 WSTRING 所用的语法如下：

声明	语法	示例
STRING	<Name> : STRING [:= <Value>] ;	myString: STRING; myString: STRING := 'hello';
WSTRING	<Name> : WSTRING [:= <Value>] ;	myWstring: WSTRING; myWstring_var: WSTRING := 'helloWorld';
长度为定义的最大值的 STRING	<Name> : STRING [[Constant]] ;	myString: STRING[10]; myString: STRING["globConst"]; myString: STRING[#locConst];

9.5 在文本视图中声明块接口

参见

STRING (页 294)

WSTRING (页 298)

声明 STRING 和 WSTRING 数据类型的变量 (页 9621)

声明块接口的规则 (页 9605)

文本块接口示例 (页 9652)

9.5.8 实例声明

多重实例声明

在文本块接口的“VAR ...END_VAR”部分声明多重实例。只需输入被调用块的名称作为数据类型。

多重实例也可创建一个 ARRAY。

无法在文本界面中初始化被调用块的参数。

另请参见：

多重背景 (页 74)

声明多重背景 (页 9623)

参数实例声明

建议在文本式块接口的“VAR_IN_OUT...END_VAR”中将参数实例声明为输入/输出参数。但也可将这些参数实例声明为函数 (FC) 中的输入或输出参数。只需输入被调用块的名称作为数据类型。

无法在文本界面中初始化被调用块的参数。

另请参见：

参数实例 (页 77)

声明参数实例 (页 9624)

实例	语法	示例	含义
多重实例	VAR <多重实例名称>:"<块名称>"; END_VAR	VAR instFB : "MyFB"; END_VAR	"instFB"是"MyFB"块的一个多重实例。
多重实例 ARRAY	VAR <多重实例名称>: ARRAY [n...m] of "<块名称>"; END_VAR	VAR instArray : ARRAY [0...4] of "MyFB"; END_VAR	"instArray"是"MyFB"块的一个多重实例数组。
参数实例	VAR <参数实例名称>:"<块名称>"; END_VAR	VAR_IN_OUT instParam:"MyFB"; END_VAR	"instParam"是"MyFB"块的一个参数实例。

参见

文本块接口示例 (页 9652)

9.5.9 变量属性

变量属性

可选择为各变量分配属性。如果没有为变量明确分配属性，变量会使用预分配的属性值。用户可在 PLC 编程的常规设置中更改某些属性的默认设置。

以下属性可用：

属性	含义	默认值
在 HMI 工程组态中可见	指示 HMI 选择列表中是否默认显示变量。	TRUE
从 HMI/OPC UA/Web API 可访问	指示在运行过程中 HMI/OPC UA/Web API 是否可访问该变量。	TRUE

9.5 在文本视图中声明块接口

属性	含义	默认值
从 HMI/OPC UA/Web API 可写	指示在运行过程中是否可从 HMI/OPC UA/Web API 写入变量。	FALSE
设定值	将变量标记为设定值。设定值是指调试过程中可能需要进行微调的值。该列仅适用于函数块的接口中。	FALSE

语法

为属性使用的语法如下：

属性	语法	示例
在 HMI 工程组态中可见	<名称> {EXTERNALVISIBLE:=<值>}: <数据类型>;	myInt {EXTERNALVISIBLE:= 'False'} : INT;
从 HMI/OPC UA/Web API 可访问	<名称> {EXTERNALACCESSIBLE:=<值>}: <数据类型>;	myInt {EXTERNALACCESSIBLE:= 'False'} : INT;
从 HMI/OPC UA/Web API 可写	<名称> {EXTERNALWRITABLE:=<值>}: <数据类型>;	myInt {EXTERNALWRITABLE:= 'False'} : INT;
设定值	<名称> {S7_SETPOINT:=<值>}: <数据类型>;	myInt {S7_SETPOINT:= 'False'} : INT;

参见

文本块接口示例 (页 9652)

PLC 编程的常规设置 (页 9487)

9.5.10 注释

注释

可通过以下几种方式为文本块接口中的变量添加注释：

- 行注释
行注释以“//”开头，直到行尾。
- 注释段
注释段以“/*”开始，以“*/”结束。该注释可跨多个行。

以下规则适用于注释部分：

- 注释无法嵌套。
- 文本式块接口中不支持多语言版注释。
- 注释不允许放置在块声明与第一个声明部分之间。
例外情况：此时，函数 (FC) 允许单行注释。

语法

为注释使用的语法如下：

属性	语法	示例
行注释	<声明>; // <注释>	MyInt : INT; // this is my comment
注释段	<声明>; (* <注释> *)	MyInt : INT; (* these are my comments: 1st comment 2nd comment *)

参见

编辑块接口中的多语言注释信息 (页 9659)

文本块接口示例 (页 9652)

9.5.11 编辑文本块接口

编辑文本块接口时，可使用的功能与编辑程序代码时相同。

9.5 在文本视图中声明块接口

使用自动完成

1. 在键盘上输入声明语法。
输入过程中，自动完成会打开。它提供当前位置允许使用的所有语法元素。
2. 在自动填充中，选择所需元素。

另请参见“手动输入 SCL 指令 (页 9988)”

自动设置行格式

1. 选择要设置格式的文本，或者将插入点放入相应的代码行。
2. 选择程序编辑器工具栏中的“自动设置所选文本的格式”(Format selected text automatically) 按钮。
只要所选行语法正确，便会设置其格式。

另请参见“设置 SCL 代码格式 (页 9980)”

缩进或伸出行

1. 单击要缩进或伸出的行。
2. 按下编程编辑器工具栏中的“缩进文本”(Indent text) 或“取消缩进文本”(Outdent text) 按钮。
可以在“选项 > 设置”(Options > Settings) 中设置缩进的长度。

另请参见“设置 SCL 代码格式 (页 9980)”

展开和折叠代码部分

1. 单击轮廓视图中的减号。
将关闭代码部分。
2. 单击轮廓视图中的加号。
将打开代码部分。

另请参见“展开和折叠代码部分 (页 9982)”

使用书签

1. 单击要设置书签的行。
2. 单击工具栏中的“设置/删除书签”(Set/delete bookmark) 按钮。

说明

接口与程序代码间的浏览视图

如果同时在块和程序代码中设置有书签，则无法通过命令“转至下一个/上一个书签”(Go to next/previous bookmark) 在两个窗口间进行浏览。

另请参见“使用书签 (页 9983)”

在文本式和表格式接口间复制和移动变量

在文本式和表格式接口间，可对变量进行复制和移动。

1. 在声明部分，选择一个或多个变量。
2. 在快捷菜单中，选择“复制”(Copy) 或“剪切”(Cut) 命令。
3. 将光标放置在变量待插入的位置处。
4. 在快捷菜单中，选择“粘贴”(Paste)。

说明

复制或移动带有初始化的结构化数据类型

将嵌套的结构化数据类型（如，ARRAY of STRUCT）从文本式接口复制到表格式接口时，下级结构的默认值不会随之复制或移动。

9.5.12 恢复组织块的默认接口

简介

组织块中定义了包含 OB 启动信息的接口参数，如果更改或删除文本块接口中的这些预定义参数，可选择通过优化的组织块恢复定义的标准接口。

操作步骤

要恢复组织块的默认接口，请按以下步骤操作：

1. 在快捷菜单中，选择命令“恢复默认接口”(Restore default interface)。

结果

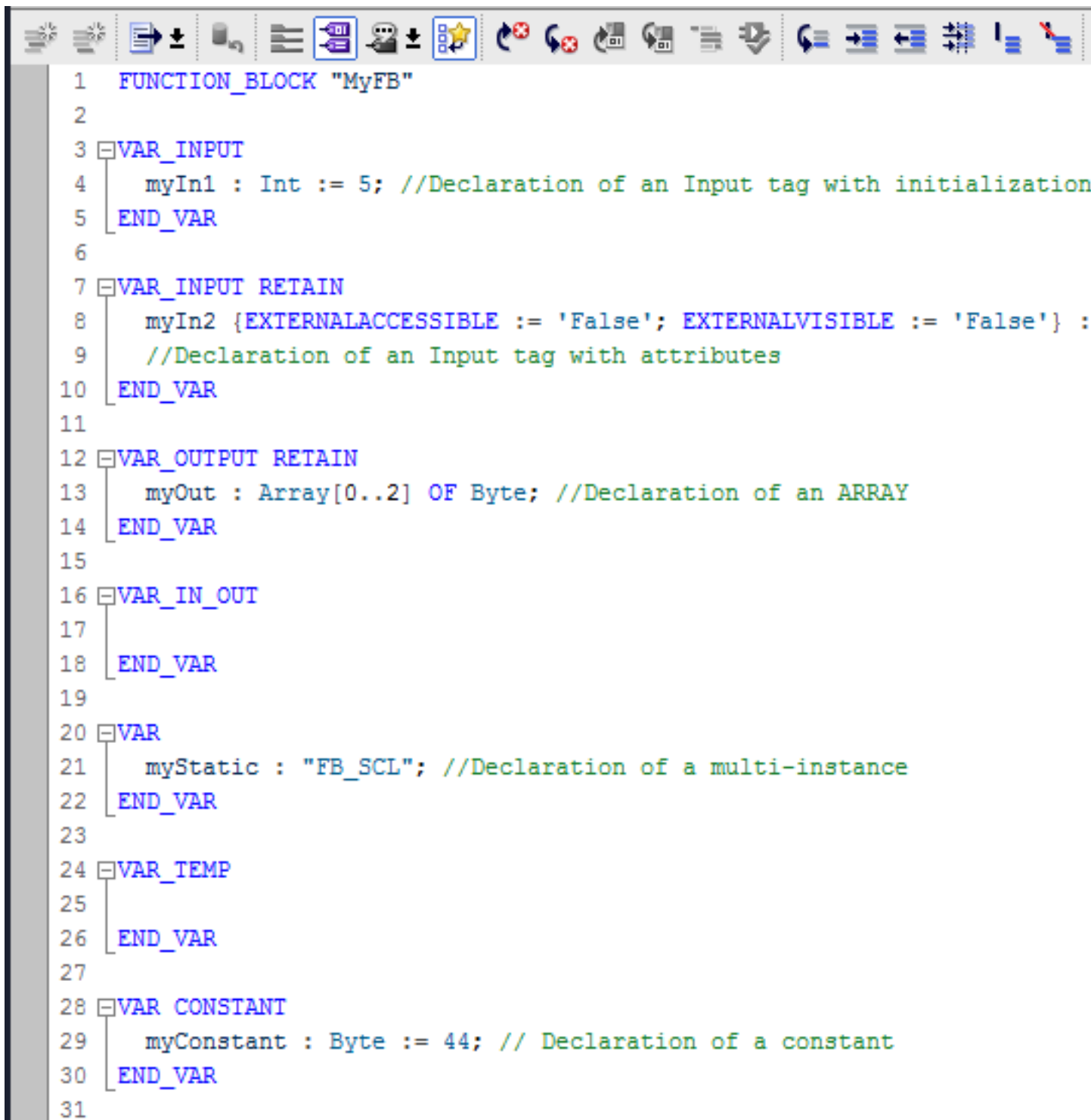
- 系统将恢复已删除或更改的 OB 参数。
- 用户手动添加的参数不会删除，而会保留下来。

9.5 在文本视图中声明块接口

9.5.13 文本块接口示例

示例

下图给出了文本块接口示例：



```
1  FUNCTION_BLOCK "MyFB"
2
3  VAR_INPUT
4      myIn1 : Int := 5; //Declaration of an Input tag with initialization
5  END_VAR
6
7  VAR_INPUT RETAIN
8      myIn2 {EXTERNALACCESSIBLE := 'False'; EXTERNALVISIBLE := 'False'} :
9      //Declaration of an Input tag with attributes
10 END_VAR
11
12 VAR_OUTPUT RETAIN
13     myOut : Array[0..2] OF Byte; //Declaration of an ARRAY
14 END_VAR
15
16 VAR_IN_OUT
17
18 END_VAR
19
20 VAR
21     myStatic : "FB_SCL"; //Declaration of a multi-instance
22 END_VAR
23
24 VAR_TEMP
25
26 END_VAR
27
28 VAR CONSTANT
29     myConstant : Byte := 44; // Declaration of a constant
30 END_VAR
31
```

9.6 编辑局部变量和局部常量的属性

9.6.1 局部变量和局部常量的属性

属性

下表概述了局部变量和局部常量的属性：

组	属性	说明
常规	名称	元素的名称
	数据类型	元素的数据类型
	默认值	代码块接口中可预分配给特定变量的值，或局部常量的值。 对于变量，可选择是否指定默认值。如果未指定任何值，则使用指定数据类型的预定义值。例如， BOOL 类型的预定义值为“false”。 在相应的实例中，将变量的默认值作为初始值。之后，即可使用实例特定的起始值替换所用的默认值。
	注释	元素的注释

9.6 编辑局部变量和局部常量的属性

组	属性	说明
属性	保持性	<p>将变量标记为具有保持性。</p> <p>即使在关断电源后，保持性变量的值也将保留不变。</p> <p>该属性只有在具有优化访问的函数块的接口中提供。</p>
	在 HMI 工程组态中可见	显示默认情况下，该变量在 HMI 选择列表中是否显示。
	从 HMI/OPC UA/Web API 可写	指示在运行过程中是否可从 HMI/OPC UA/Web API 写入变量。
	从 HMI/OPC UA/Web API 可访问	<p>指示在运行过程中 HMI/OPC UA/Web API 是否可访问该变量。</p> <p>但请注意，属性为“从 HMI/OPC UA/Web API 可访问”的变量不能进行常规访问保护。即使未启用此属性，也可从其它应用程序进行读/写访问。</p>
	LAD/FBD 中块调用的可见性	<p>显示：</p> <p>在 LAD 或 FEB 的块调用中，该参数始终显示。</p>
		<p>隐藏：</p> <p>在 LAD 或 FEB 的块调用中，该参数始终隐藏。</p> <p>此时，指令框的下边缘处将显示一个小箭头。单击该箭头，可显示这些参数并进行互连。</p>
		<p>未分配参数时隐藏：</p> <p>在 LAD 或 FEB 的块调用中，参数未互连时始终隐藏。如果指定一个实参，则该参数将显示在调用框中。</p>
	预定义实参	定义在块调用过程中用作实参的参数。
	块调用时指定的参数与预定义的实参不同时，显示	<p>只要未指定预定义的实参，则该参数将在 LAD 或 FEB 的块调用中隐藏。但如果指定了不同的参数，则该参数将显示可见。</p> <p>仅当选择了“隐藏”参数并定义了一个预定义实参时，该选项才激活。</p>

9.6 编辑局部变量和局部常量的属性

组	属性	说明
用户自定义属性	CFC_Configurable	可组态 指出参数是否可在 CFC 中组态。
	CFC_ForTest	用于测试 指出是否将参数注册为用于 CFC 测试模式。
	CFC_Visible	可见性 指出参数是否在 CFC 中显示。
	CFC_Interconnectable	互连性 指出参数在 CFC 中是否可互连。
	CFC_EnableTagReadback	启用变量回读 指示参数是否与 CFC 中的“回读表”功能相关。
	CFC_EnumerationTexts	枚举文本 该属性仅在内部使用。
	CFC_EngineeringUnit	工程组态单元 为 CFC 中的单元分配一个参数。
	CFC_LowLimit	下限 定义 CFC 中参数的下限。
	CFC_HighLimit	上限 定义 CFC 中参数的上限。

参见

- 设置局部变量的保持性 (页 9611)
- 更改局部变量和局部常量的属性 (页 9657)
- 关键字和保留标识符 (页 106)
- 在调用块过程中隐藏参数 (页 9625)
- 声明预定义的实参 (页 9626)

9.6.2 更改局部变量和局部常量的属性

在表格式块接口中编辑某个元素的属性

要在块接口编辑元素的属性，请按以下步骤操作：

1. 打开块接口。
2. 在表中选择所需元素。
3. 更改列中的条目。

在表格式块接口中编辑多个元素的属性

对于选择的多个元素，可同时对某些属性进行设置或复位。

要更改多个元素的其中一种属性，请按以下步骤操作：

1. 打开块接口。
2. 按住 <Ctrl> 键。
3. 在所需列中，选择要更改值的各个表单元格。
4. 在快捷菜单中，选择“设置 <属性>”(Set <property>) 或“复位 <属性>”(Reset <property>) 命令。

在属性窗口中编辑属性。

要编辑某个变量或常量的属性，请按以下步骤操作：

1. 选择该变量或常量。
元素属性将显示在“巡视”(Inspector) 窗口中。
2. 更改巡视窗口中的条目。

在程序编辑器中直接重命名变量

要重命名一个或多个元素，请按以下步骤操作：

1. 在程序选择一个或多个元素。
2. 在快捷菜单中，选择“重命名变量”命令。
将打开“重命名变量”对话框。显示所选元素的声明表。
3. 更改“名称”列中的条目。
4. 单击“更改”按钮确认输入。

9.6 编辑局部变量和局部常量的属性

在程序编辑器中编辑数据类型或注释。

要在程序编辑器中编辑数据类型或注释变量，请按以下步骤操作：

1. 选择变量的名称。
2. 在快捷菜单中，选择“重新连接变量”命令。
将打开“重新连接变量”对话框。在此对话框中，将显示一个变量声明表。
3. 更改“数据类型”或“注释”列中的条目。
4. 单击“更改”按钮以确认输入。

在程序中生效

对于更改变量或常量的名称、数据类型或地址的情况，在程序中变量的每个应用位置都会自动更新。

说明

如果更改了块接口，则程序可能变为不一致。如果可能，可对不一致程序进行自动更新。如果无法进行自动更新，则对不一致的调用以红色进行标记。然后，必须手动更新不一致。

另请参见

在 LAD 中更新块调用 (页 9801)

在 FBD 中更新块调用 (页 9863)

参见

块接口概述 (页 9603)

局部变量和局部常量的属性 (页 9654)

设置局部变量的保持性 (页 9611)

有关起始值的基本信息 (页 9677)

在程序中使用变量 (页 112)

关键字和保留标识符 (页 106)

更新块接口 (页 9660)

编辑块接口的表格 (页 9627)

声明块接口的规则 (页 9605)

9.6.3 编辑块接口中的多语言注释信息 (S7-1500)

用户可对块接口处的注释信息翻译为所有项目语言，这些项目语言在项目的语言设置中已激活。有关激活项目语言的更多信息，请参见“AUTOHOTSPOT”。

请注意，即使翻译后，文本的长度也不得超过 32767 个 Unicode 字符。

文本式块接口中不支持多语言版注释。

要求

- 已经激活多个项目语言。
- 块接口打开，且包含至少一条注释信息。
- 该块未设置专有技术保护。

操作步骤

要将某条注释信息编辑为所有项目语言，请按以下步骤操作：

1. 单击待编辑译文的注释信息。
如果要同时编辑多条注释信息的译文，则可选择所有所需的注释信息。
2. 在巡视窗口中，打开“属性”(Properties) 选项卡。
3. 打开“文本”(Texts) 选项卡。
在“文本”(Texts) 选项卡中，将以当前激活的项目语言显示所选注释信息。如果该语言可用，则显示该注释的翻译文本。
4. 用户可在“文本”(Texts) 选项卡的表格中直接编辑翻译文本。
5. 要使用外部编辑器编辑翻译文本，可使用“导出/导入项目文本”(Export/Import project texts) 按钮，将所显示的文本信息导出为 OOXML 格式。

说明

在全局“项目文本”(Project texts) 表中，编辑所有项目文本信息

也可在全局“项目文本”(Project texts) 表格中，编辑各项目语言的翻译文本。该表格位于项目树下的“语言和资源 > 项目文本”(Languages & Resources > Project texts) 中。其中包含整个项目中所有可翻译的文本信息。

有关文本翻译的更多信息，请参见“AUTOHOTSPOT”。

9.7 更新块接口

简介

如果更新或删除块接口中使用的 PLC 数据类型或多重背景，则接口会变得不一致。要解决不一致问题，必须更新接口。

有两种选择可以更新块接口：

- 显式更新块接口。
更新所用的 PLC 数据类型和多重背景。在此过程期间，属于块的背景数据块不会隐式更新。
- 编译期间隐式更新。
将更新所有的 PLC 数据类型和多重背景以及相关的背景数据块。

块接口的显式更新

要显式更新块接口，请按以下步骤操作：

1. 打开块接口。
2. 在快捷菜单中，选择“更新”命令。

编译期间的隐式更新

在编译期间，按以下步骤操作以隐式更新所有使用的 PLC 数据类型和多重背景以及背景数据块。

1. 打开项目树。
2. 选择“程序块”文件夹。
3. 选择快捷菜单中的“编译 > 软件（重建所有块）”命令。

参见

声明基于 PLC 数据类型的变量 (页 9621)

有关起始值的基本信息 (页 9677)

在程序中使用变量 (页 112)

关键字和保留标识符 (页 106)

局部变量和局部常量的属性 (页 9654)

设置局部变量的保持性 (页 9611)

在 LAD 中更新块调用 (页 9801)

声明多重背景 (页 9623)

编辑块接口的表格 (页 9627)

声明块接口的规则 (页 9605)

9.8 扩展块接口 (S7-1200, S7-1500)

说明

为了可以编辑已经过调试并在系统上正常运行的 PLC 程序，S7-1500 产品系列的 CPU 和 S7-1200 V4 及更高版本系列的大多数 CPU 都支持在运行时对函数块接口进行扩展。

无需将 CPU 设置为 STOP 模式，即可下载已修改的块，而不会影响所加载变量的值。

这是一种简单的更改程序的方式。这一加载过程（无需重新初始化的下载）不会对所控制的过程造成负面影响。

工作原理

每个函数块都被分配了一个默认存储器预留区域。在初期并未使用该存储器中所预留的区域。如果该块已编译且已加载，并确定稍后再加载接口更改，则可激活存储器预留。之后声明的所有变量都将保存到存储器预留区域中。后续下载不会影响任何已加载的变量，或对正在进行的操作造成不利影响。

如果决定稍后在工厂没有运转时检查程序，那么还可以一次性重新设置一个或多个块的存储器布局。通过该操作，可以将所有变量从预留区域移动到常规区域。存储器预留区域现已清除，可用于进行接口扩展。

要求

如果满足以下要求，则可使用“无需重新初始化的下载”功能：

- 项目为“TIA Portal V12”格式，或更高版本。
- 使用支持“无需重新初始化的下载”的 CPU。
- 块使用 LAD、FBD、STL 或 SCL 创建。
- 块由用户创建，即，这些块不包含在随包交付的块中。
- 这些块将被分配到优化访问属性。

9.8 扩展块接口 (S7-1200, S7-1500)

基本步骤

如果希望对某个函数块的接口进行扩展并随后加载而不重新进行初始化，则请按以下步骤操作：

1. 所有块都默认预留了 100 个字节的存储器。可以修改这一存储器预留区域以适应用户的需求。
2. 激活存储器预留区域。
3. 扩展块接口。
4. 编译块。
5. 照常将块下载到 CPU。

有关各种步骤的更多信息，请参见“加载块 (S7-1200/1500)”章节。

说明

“无需重新初始化的加载”的完整功能仅适用于 S7-1500 和 S7-1200 V4 产品系列的 CPU。但是，所有 CPU 系列都可受影响地扩展函数块接口并下载新声明的变量：

- 可以在“Temp”区域添加新变量并在不影响过程的情况下下载这些变量。
 - 可以在“InOut”区域中新建结构化数据类型的变量并在不影响过程的情况下下载这些变量。
-