中控·SUPCON

AdvanTrol-Pro 软件

图形编程

浙江中控技术股份有限公司

目　录

1 图形编程.....1-1
1.1 图形编程简述.....1-1
1.2 图形编程的性能特点.....1-2
1.3 编程指南.....1-3
1.3.1 综述.....1-3
1.3.2 工程管理.....1-3
1.3.3 功能块图（FBD）语言.....1-11
1.3.4 梯形图（LD）语言.....1-17
1.3.5 顺控图 SFC.....1-26
1.3.6 ST 语言概述.....1-35
1.4 图形编程使用指南.....1-58
1.4.1 图形编程的运行环境.....1-58
1.4.2 编程界面介绍.....1-58
1.4.3 菜单功能项介绍.....1-61
1.4.4 变量类型说明.....1-69
1.4.5 工程设计.....1-70
1.4.6 文件结构.....1-71
1.4.7 在线调试.....1-71
1.4.8 密码保护.....1-73
1.5 图形编程模块库.....1-74
1.5.1 IEC 模 块 库.....1-74
1.5.2 辅助模块库.....1-165
1.5.3 附加模块库.....1-253
2 资料版本说明.....2-1

1 图形编程

图形编程软件是 SUPCON 集散控制系统软件的重要组成部分之一，基于 Windows 操作系统设计，充分利用 Windows 操作系统的优点，具有良好的用户界面。

图形编程简述

图形编程软件用图形方式描述控制过程，使控制过程组态变的更简单，也使控制工程师可以专注于控制方案。

图形编程性能特点

图形编程集成了 LD 编辑器、FBD 编辑器、SFC 编辑器、ST 编辑器、数据类型编辑器、变量编辑器、DFB 编辑器。采用工程化的文档管理方法，提供了一个功能强大的实现程序重用和结构化的工具 。

编程指南

图形编程软件的编程包括 LD 语言编程、FBD 语言编程、SFC 语言编程和 ST 语言编程。FBD 编辑器、LD 编辑器、SFC 编辑器和 ST 编辑器作为其最重要的编辑器，与变量编辑器、数据类型编辑器、DFB 编辑器等共同构成了一个强大的编辑环境。

图形编程使用指南

使用指南介绍了软件的运行环境和工作界面，并介绍了用户如何利用菜单功能、图形化的功能模块及其他一些工具，顺利地进行 LD、FBD、SFC 及 ST 语言的编程。

图形编程模块库

模块即为图形编程软件的功能模块，图形编程软件提供了近 200 个基本模块供用户选用，分为 IEC 模块以及非 IEC 标准模块两大类。另外，用户还可以使用自行设计的自定义模块。

1.1 图形编程简述

图形编程软件,作为集成的图形编程工具，是针对集散控制系统所开发的全中文界面的 DCS 组态与控制工具，是 SUPCON 系列 DCS 的控制方案组态工具，依据 IEC61131-3 标准，为用户提供高效的组态环境，与系统组态软件联合完成对系统的组态，是 SUPCON 集散控制系统软件的重要组成部分之一。图形编程软件基于 Windows 操作系统设计，充分利用 Windows 系统的优点，具有良好的用户界面。

图形编程软件的组态通过图形用户接口进行，只要求用户有基本的 Windows 操作基础。

图形编程提供灵活的在线调试功能，用户可以观测程序的详细运行情况。

图形编程提供了详细的在线帮助，上下文关联的联机帮助使用简单的按鼠标或 F1 键为组态中的每种情况提供支持。

1.2图形编程的性能特点

图形编程集成了 LD 编辑器、FBD 编辑器、SFC 编辑器、ST 编辑器、数据类型编辑器、变量编辑器及 DFB 编辑器。

图形编程的所有编辑器使用通用的标准 File、Windows、Help 等菜单。灵活地自动切换不同编辑器的特殊菜单和工具条。

图形编程在图形方式下组态十分容易。在各编辑器中，目标（功能块、线圈、触点、步、转换等）之间的连接在连接过程中进行语法检查。不同数据类型间的链路在编辑时就被禁止。图形编程提供注释、目标对齐等功能改进图形程序的外观。

图形编程采用工程化的文档管理方法。通过导入导出功能，用户可以在工程间重用代码和数据。

图形编程 DFB 提供了一个功能强大的实现程序重用和结构化的工具。

图形编程的特点可简单归纳如下：

1. 使用 Windows 的友好图形界面，使用鼠标也可以使用键盘进行编辑操作，工具条上所有功能都有文字提示；
2. 编辑环境通过工程文件管理多个图形文件，用户容易操作；
3. 组态元素放置灵活，自动格线对齐，触点、线圈、功能块和变量等可用文本进行注释；
4. 图形绘制采用矢量方式，具备块剪切、拷贝、粘贴、删除等功能，达到事半功倍的效果；
5. 具备对前次操作步骤的撤消和恢复功能，提高了组态效率；
6. 智能连线处理，模块引脚接近时自动连接；
7. 连线时动态检查数据类型，数据类型不一致拒绝连接；
8. 强大的查找和替换功能，可在当前程序段也可在当前整个工程中查找变量、常数、位号及模块，并进行标记，用户只需用鼠标点击相应的信息就可以直接跳到所要查找的位置。替换功能亦然，可在当前程序段或当前整个工程中逐个替换或全部替换所选择的变量、常数、位号 及模块；
9. 提供缩放功能，使用户更清晰地查看页面或按照缩小的比例看到页面中更多的内容；
10. 系统为用户管理定义的位号和变量，用户不用关心具体物理内存；
11. 在每个编辑器中可以使用系统已定义的基本功能模块（EFB）和用户自己定义的功能模块(DFB)。每个编辑环境中内嵌自定义模块（DFB）编辑器。极大地提高了程序的重用性，减少编程工作量；
12. 用户可以用 EFB 和 DFB 再组成新的 DFB。具有无限的功能扩展性。方便用户做二次开发；
13. 用户可以使用工程的导入导出功能重用功能模块；
14. 用户可通过数据类型编辑器生成自定义的数据类型；
15. 功能块编辑器（FBD）、梯形图编辑器（LD）及顺控图编辑器(SFC)集成在一起，可相互嵌套调用，具有无限的功能扩展性；
16. 提供在线调试功能；

17. 强大的在线帮助功能。

1.3 编程指南

图形编程软件的编程包括 LD 语言编程、FBD 语言编程、SFC 语言编程和 ST 语言编程。编程流程包括工程的创建、段落的创建、区段的创建、程序段的编辑、工程的编译 及链接等几个过程。

在图形编程软件中，FBD 编辑器、LD 编辑器作为最重要的编辑器，与变量编辑器、数据类型编辑器、DFB 编辑器等共同构成了一个强大的编辑环境。

用户可进行在线调试，可以将外部现成的有用 LD、FBD 程序导入进工程中，也可以将本工程中比较实用、或能用于其他工程的各种文件通过导出操作，提供给其他工程利用，充分代码重用。

1.3.1 综述

图形编程的编程语言包括功能块图（FBD）、梯形图（LD）、顺控图（SFC）及 ST 语言。支持国际标准 IEC61131-3 数据类型子集。用户可以使用数据类型编辑器生成自己的数据类型。

图形编程的每一个工程对应一个控制站。工程可包含多个段落。每个段落只能选用一种编辑器。按 IEC61131-3 标准，FBD 编程语言的基本元素是功能块；LD 编程语言的基本元素除了功能块外还包括触点和线圈；SFC 编程语言的基本元素是转换、步和跳转；ST 编程语言除了可使用基本的 ST 语法外，还可调用系统函数。在工程中可以分别指定不同段落的执行周期和执行次序。

图形编程提供以下编辑器：

- FBD 编辑器
- LD 编辑器
- SFC 编辑器
- ST 编辑器

在生成段落时，用户可以指定生成的段落的类型。段落的类型指定了使用何种编辑器。

除与编程语言有关的编辑器外，还有：

- 数据类型编辑器
- 变量编辑器

1.3.2 工程管理

以下介绍：

- 工程
- 段落
- 区段
- 变量
- 注释文本
- 调试文本

1. 工程

图形编程用一个工程（Project）描述一个控制站的所有程序。工程包含一个或多个段落（Section）。每个工程唯一对应一个控制站，工程必须指定其对应的控制站地址。图形编程通过工程管理多个段落文件，在工程文件中保存配置信息。

2. 段落

段落是通常意义上的一个文档，是组成工程的基本单位。

新建段落时必须指定段落的编辑类型和程序类型。

按编辑类型可将段落分类为：

- FBD 段落
- LD 段落
- SFC 段落
- ST 段落

按程序类型分可将段落分类为：

- 程序段落
- 模块段落

选择编辑类型相当于选择何种编辑器进行编程。选择程序类型相当于选择是生成一个可执行的程序或是进入 DFB 编辑器生成 DFB 模块。

3. 段落管理

选择**工程**菜单中的**段落管理**进入段落管理对话框。

图 1-1 段落管理对话框

可以通过**新建**按钮新建一个段落，效果与**文件**菜单中的**新建程序段**命令一样。

图 1-2 新建程序段

选择一个段落，然后可以通过**打开**按钮打开段落。

选择**文件**菜单中的**打开程序段**命令。弹出打开段落对话框，选择需打开的段落按打开按钮也可以打开段落。

图 1-3 打开段落

直接在工程栏中双击相应的段落名，也可以打开段落。

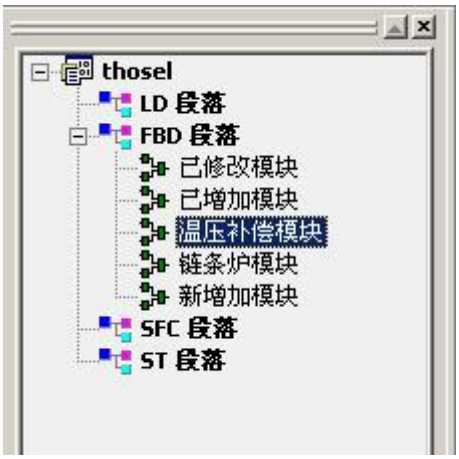


图 1-4 工程栏中打开段落

在段落管理对话框中选择想删除的段落，按删除按钮可以删除段落。

在工程栏中，选择想删除的段落，按鼠标右健，弹出浮动菜单，选择删除段落，也可以删除段落。

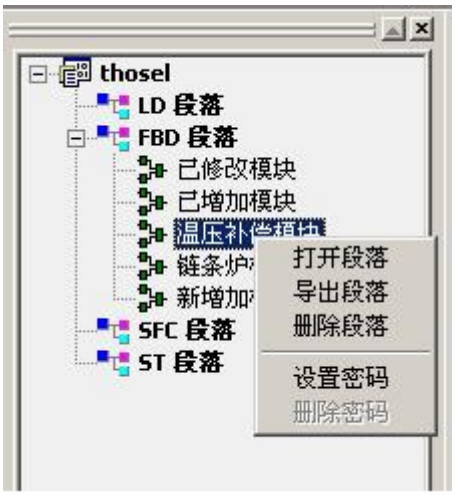


图 1-5 后键菜单

在段落管理对话框中选择想导出的段落，按**导出**按钮可以导出段落到文件。

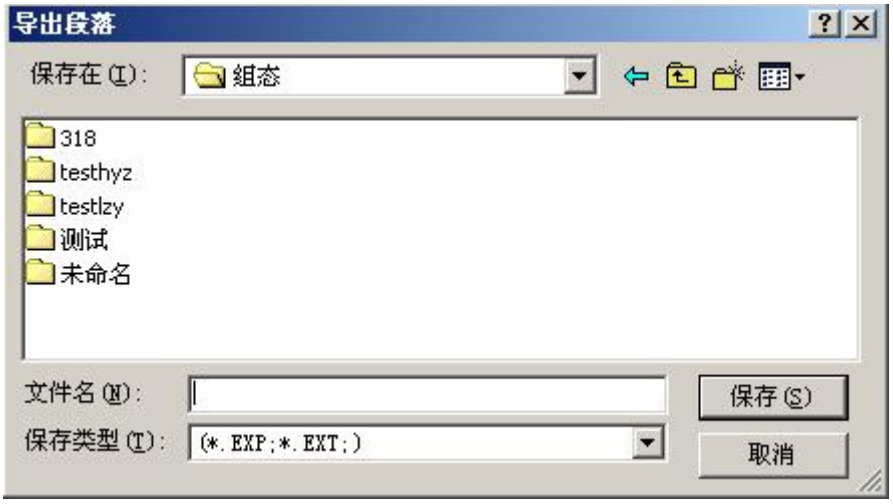


图 1-6 导出段落

导出文件名必须为*.exp。

在工程栏中，选择想导出的段落，按鼠标右健，弹出浮动菜单，选择导出段落，也可以导出段落。

在段落管理对话框中按**导入**按钮可以从文件中导入段落到工程。

当在 **段落管理** 对话框中选择一个或多个段落导出时，用户要指定导出段落存放的文件名。图形编程先检查所有的段落，如段落中包含未被选择的 DFB，则图形编程自动追加这些 DFB 段落。然后检查所有段落中包含的变量的数据类型，若发现其中的数据类型是由自定义数据类型派生而来，图形编程将自动追加这些自定义数据类型。导入时，选择已生成的导出文件，工程中将添加所包含的数据类型、段落。当导入时，发现段落名冲突，将提示用户是替换或保留或用新名导入。

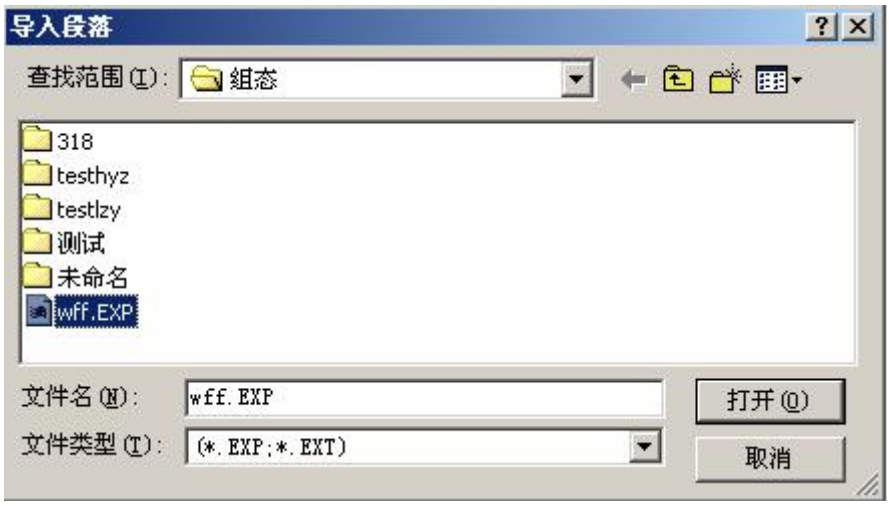


图 1-7 导入段落

在段落管理对话框中按**修改**按钮可以修改段落名。

图 1-8 修改段落名称

4.任务管理

当有多个程序时，程序段落的执行周期和执行次序会影响程序的运行结果。在工程菜单中选择任务管理弹出任务管理对话框用于设置执行周期和执行次序。

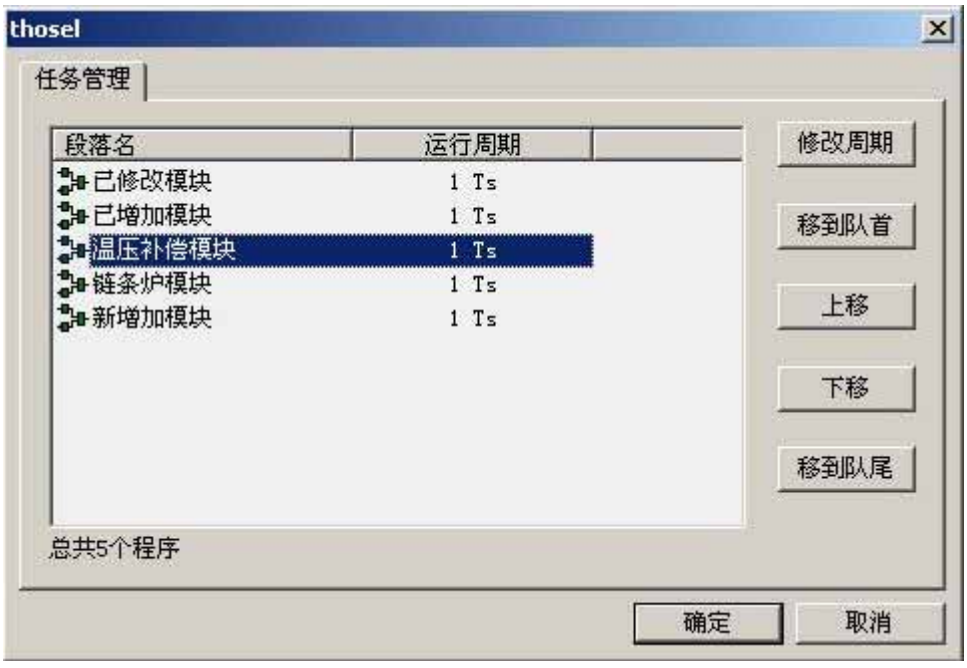


图 1-9 任务管理对话框

图形编程以系统组态软件中设置的控制周期为 1Ts，即：如果在系统组态软件的组态过

程中设置了控制周期为 0.1s，则 1Ts = 0.1s.

用户还可通过操作“移到队首”、“上移”、“下移”、“移到队尾”等操作设置同一运行周期各程序运行的优先级，即排在队列靠前的同一运行周期程序比排在队列靠后的程序优先执行。不同运行周期的程序之间的优先级无法比较。

5.区段

区段指在同一段落中有数据信号相连的元素的总和。一个段落可以包含一个或多个区段（SFC 段落只有一个区段）。

在区段内 EFB 或 DFB 的执行次序是由区段中间的数据流来决定的。在 FBD 区段内那些输入只连接变量或位号或常数的、在 LD 区段输入只连接变量或位号或常数或左汇流条的、SFC 区段中的起始步，被称为区段的**起始模块**。区段内有多个起始模块时，在图形区域中位置最上的模块称为**启动模块**。区段的执行就从启动模块开始，按数据流要求逐步进行。

同一段落内区段间的执行次序就依据区段的启动模块在图形区域中位置来决定。启动模块在上的先执行。

6. 数据类型

类 型	关键字	字节数	表 示 范 围
无符号字节型	BOOL	1	0 ~ 255
字	WORD	2	0 ~ 65535
双字	DWORD	4	0 ~ 4294967295
整型	INT	2	-32768 ~ +32767
无符号整型	UINT	2	0 ~ 65535
长整型	LONG	4	-2147483648 ~ 2147483647
无符号长整型	ULONG	4	0 ~ 4294967295
半浮点型	SFLOAT	2	-7.9997 ~ +7.9997
浮点型	FLOAT	4	±1.175490351E-38 - 3.402823466E+38
累积型	structAccum	8	

7. 累积类型

在程序中可以直接定义累积类型变量。累积类型是系统提供的一种结构类型，即 structAccum。该结构的定义如下：

```
struct structAccum
{
  sfloat remainder;     //小数部分
  long accum;           //整数部分
  int reserved;         //保留部分，禁止使用
}
```

由定义可以看出，累积类型变量含有三个成员：remainder、accum、reserved。其中保留部分禁止用户使用，remainder 表示累积的小数部分（小于 1），accum 表示累积的整数部分。当小数部分超过表示范围[0－1）时，自动向整数部分进位。

模拟量累积

模拟量累积在工程中大量运用，它由两部分构成：

sum1	sum0

其中，sum1 占 32 位，是长整形；sum0 占 16 位，是 sfloat 型，是无符号 12 位定点小数，整数部分占 4 位。当模拟量累积超过 sum0 所能表示的范围（0－15.999），自动向高位的 sum1 进位。

例如，accum=12.123443 时，accum.sum0=12.123443，accum.sum1=0。

accum=34.457638 时，accum.sum0=2.457638，accum.sum1=2，因为 sum1 从第 17 位开始，所以 sum1=2,实际指的是 2×16=32。

8. 数据类型存储方式

在计算机中，所有数据都由二进制表示：

BOOL：占 1 字节，零表示 FALSE,非零表示 TRUE；

WORD：两字节，占 16 位，无符号；

DWORD：四字节，占 32 位，无符号；

INT：两字节，占 16 位，最高位是符号位：0 表示正数，1 表示负数；

UINT：两字节，占 16 位，无符号；

LONG：四字节，占 32 位，最高位是符号位：0 表示正数，1 表示负数；

ULONG：四字节，占 32 位，无符号；

SFLOAT：两字节，占 16 位，用定点法表示。在定点表示法中，二进制小数点位置通常是固定不变的。小数点可以固定在数值位之前，也可以固定在数值位后面。前者称为定点小数表示法，后者叫做定点整数表示法。SFLOAT 定点数 N 的一般表示形式为：

符号位	整数位	尾数

其中，符号位占一位：0 为正数，1 为负数；整数位占三位；尾数占十二位。

FLOAT：四字节，占 32 位，用符点法表示。在采用符点表示的二进制数中，小数点位置是浮动的,不固定的。通常任何一个二进制都可以写成：

$$N=2^P\times S$$

式中，S 为二进制数 N 的尾数，代表了 N 的实际有效值；P 为 N 的阶码，可以决定小数点的具体位置。因此，任何一个符点数 N 都由阶码和尾数两部分组成。阶码部分包括阶符和阶码，尾数部分有数符和尾数组成。其形式为：

阶符	阶码	数符	尾数

其中，阶符占一位，阶符=0 表示阶码为正，阶符=1 表示阶码为负；阶码为七位；数符占一位，数符=0 表示该数为正数，数符=1 表示该数为负数；尾数为二十三位。

9. 变量

变量包括用于在段落中间、段落之间的指定名称的数据以及操作站和控制站进行数据交换的位号。

变量按组织形式分为：

- 基本变量
- 复合变量

基本变量的数据类型是基本数据类型。复合变量的数据类型为复合数据类型。复合数据类型通过数据类型编辑器生成，通过基本数据类型和已生成的复合数据类型组合而成。

变量按作用关系分为：

- 全局变量

全局变量指在段落之间共享的变量。在工程中声明全局变量后，在所有段落都可以访问。全局变量一经声明，就被分配一个固定的控制站地址，放在系统数据区中，能够将当前数据保持到下一个控制周期。

- 私有变量

在程序中可以声明私有变量。私有变量一经声明，就被分配一个固定的控制站地址，放在系统数据区中，能够将当前数据保持到下一个控制周期。私有变量与全局变量的不同在于私有变量只有声明的段落能够存取，其他段落对此变量不可见。变量的作用范围就被限制在当前段落中。变量封装有利于用户编程。

- 输入变量与输出变量

在 DFB 段落（创建时程序属性被指定为模块的段落）中，可以声明输入变量和输出变量。通过指定输入变量和输出变量，就设定了 DFB 的外部接口。对输入变量和输出变量，系统不分配控制站内存。

在 DFB 段落中也可以声明私有变量。但此段落中的私有变量只有在 DFB 被使用时才被分配控制站地址。DFB 每被使用一次，私有变量就被创建一次。DFB 相当于一个类，此私有变量相当于成员。类可以创建多个实例，而每个实例的成员地址都不同。DFB 段落内的私有变量与 DFB 外部接口无关，在 DFB 外部不可见，主要用来存放需要将当前状态保持到下一个控制周期的数据。

用包含私有变量的 FFB 构建的 DFB 时，新的 DFB 将继承包含的 FFB 的私有变量。在 DFB 段落中，私有变量在保证 DFB 重用性的同时还满足了与时序相关的控制的要求。

系统中还包含了一种对用户不可见的热备份变量。热备份变量用于当 DCS 系统配置为双主控卡热冗余时，在双主控卡间定时的同步数据。如积分模块，就包含了一个热备份变量，用来同步积分的当前值。这样，在 冗余切换时，保证了无扰动切换。通过这种热冗余方式，充分保证了系统的安全性。

跟时间相关具有累积作用的 EFB 模块都包含了热备份变量。

用包含热备份变量的 FFB 构建新的 DFB 时，新的 DFB 将继承包含的 DFB 的热备份变量。

10. 注释文本

注释文本在程序中增加标注信息，用于增加程序的可读性。文本目标的大小取决于文本的长度。根据文本字体大小，目标的大小可以通过在垂直以及在水平方向上更多的网格来进行扩展。注释文本不占控制站内存。注释文本的字体和颜色都可以设置。

11. 调试文本

调试文本是在联机状态下显示变量或位号在控制站中的实际值的文本信息。用户通过调试文本可以操纵控制站数据。方便调试程序和监视系统运行。调试文本不占控制站内存。

图 1-10 属性

12. 查找与替换

用于进行用户所需信息（如模块、文字、变量名等）的检索。用户可自行填写，也可按照不同类型、数据源通过浏览进行检索。

按 CTRL+F 或用编辑菜单的查找命令进入查找对话框。

图 1-11 查找对话框

使用查找功能可以查找变量和模块。选择当前段则在当前打开的段落中搜索目标，选择当前工程将搜索整个工程。按查找按钮，鼠标将跳到搜索到的下一个目标并将其选中。按标记按钮则在信息栏中显示所有搜索到的目标。用鼠标双击目标信息，则打开目标所在得段落，并跳到目标位置。

按 CTRL+H 或用编辑菜单的替换命令进入替换对话框。

图 1-12 替换对话框

按查找按钮，鼠标将跳到搜索到的下一个目标并将其选中。然后按替换，则搜索目标将被替换成替换目标。如果按全部替换按钮，则所有符合搜索条件的目标都被替换。

1.3.3 功能块图（FBD）语言

根据 IEC1131-3，FBD 编辑器将基本的功能/功能块（EFB）和信号（变量、位号）组成

功能块图（FBD）。EFB 和变量可以加注释。图形内可以自由放置基本元素和文本。部分 EFB 的输入可以扩展方便使用。

在 FBD 编辑器中，窗口的背景是逻辑坐标网格。当正在创建时，功能块在该网格的光栅中对准。如果功能块发生与别的功能块重叠的情况，将会出现错误信息并且功能块不创建。当实际参数在功能块输入/输出创建时，他们可能与别的目标重叠但不会破坏区段画面的界限。如果有一链路作为与别的功能块的连接，则该连接要进行检查。如果这是一个未经许可的连接，该链路将不生成。

除这些目标之外，还可以将注释文本和调试文本放入 FBD 段落中。该文本目标的大小取决于文本的长度。根据文本大小，目标的大小可以通过在垂直以及在水平方向上更多的网格来进行扩展。

1. FBD 编辑器

根据 IEC61131-3，FBD 编辑器将基本的功能/功能块（EFB）和信号（变量、位号）组成功能块图（FBD）。EFB 和变量可以加注释 ，图形内可以自由放置基本元素和文本，部分 EFB 的输入可以扩展方便使用。

图形编程提供了部分预定义的 EFB 模块库，包含了近 200 个基本模块，并且库中的模块被组织成不同的组。 模块库包括：

- IEC 模块库

包括在 IEC61131-3 中定义的功能块。包括：算术运算、数学运算、逻辑运算、转换、选择、触发器、计数器、定时器等类型。

- 辅助模块库

包括控制模块、通讯辅助、累积函数、输入处理及混合运算等。

在 FBD 编程中，用户可以使用基本功能块和自定义功能块。

用户用 DFB 编辑器生成的自定义模块被放在自定义模块库中。DFB 加入模块库中后，就可以被各种语言编辑器使用。

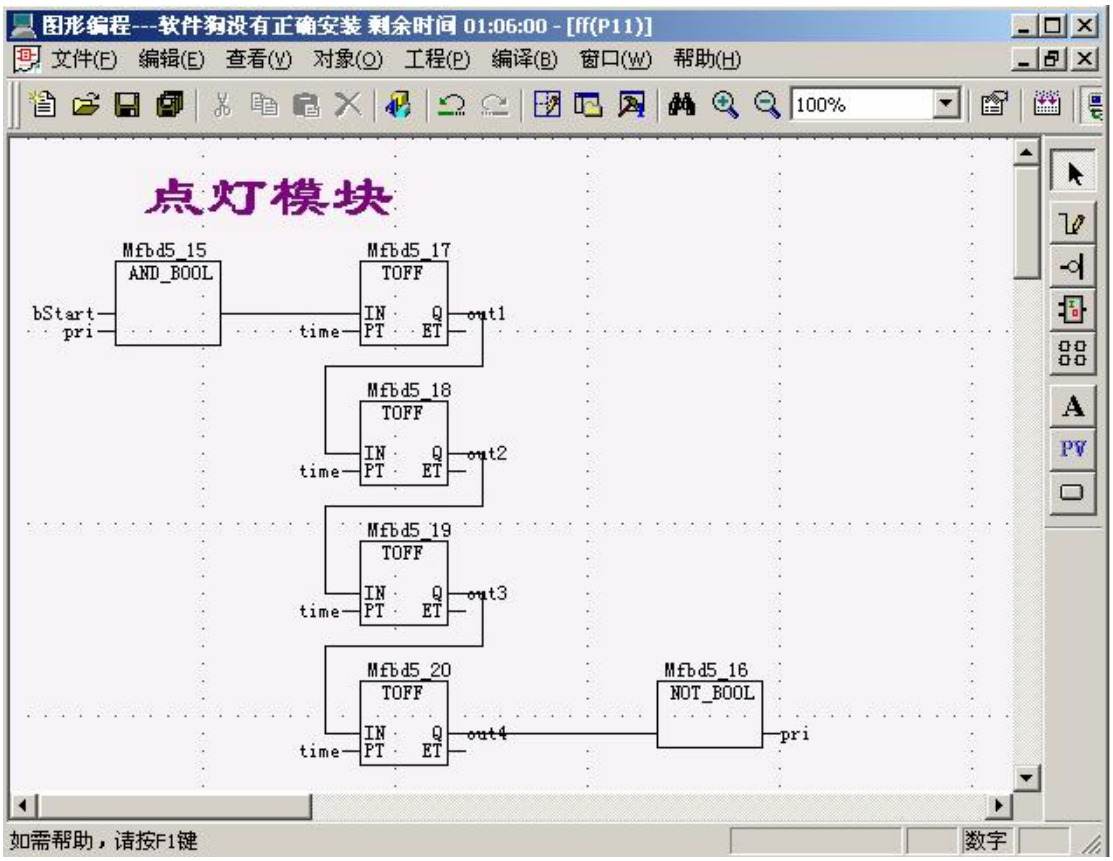


图 1-13 FBD 编辑器

2. FFB（功能和功能块）

FFB 是基本功能块（EFB）和自定义功能块（DFB）的统称。功能块用带有输入和输出的图形框来描绘。输入在图形框的左边，输出在图形框的右边。功能块的名称在图形框的中间显示。功能块的实例名在图形框上显示。在同一工程内，模块的实例名是唯一的。

所有功能块都可以用一个 EN 输入和一个 ENO 输出进行配置。

如果当调用功能块时 EN 值等于 0，则该功能块将不被执行，ENO 值自动设置成 0；如果 EN 值等于 1，则该功能块将被执行，执行完后，ENO 值自动设置成 1。

在不需要 EN 的时候，可以隐藏 EN 和 ENO 引脚。以下分别是显示有 EN、ENO 口与隐藏了 EN 和 ENO 口的模块示意图：

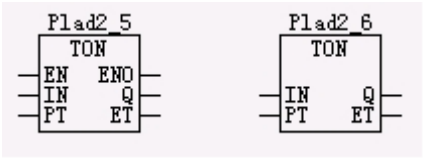


图 1-14 模块示意图

FFB 的输入/输出关系不受 EN 和 ENO 的影响。

EFB 和 DFB 都可以包含私有变量和热备份变量。在编辑 DFB 时，用户可以添加私有变量。在引用时，FFB 的私有变量和热备份变量对用户透明。

3. 实际参数

在程序运行中，取值过程的值和其他数据是通过实际参数向 FFB 传送的。实际参数包含变量、位号、常数。

实际参数的数据类型必须与相连接的引脚类型一致。

4. 基本功能块库

图形编程提供了一个基本功能块库，包含了以下几类模块：

- 算术运算，如 ADD、SUB、MUL、DIV
- 比较运算，如 GE、LT、NE
- 转换运算，如 INT_TO_LONG
- 数学函数，如 SIN、EXP、LOG
- 逻辑运算，如 AND、OR、XOR、
- 选择运算，如 MUX、LIM
- 计数器、定时器、触发器，如 TON
- 累积函数，如 ACCUM_TO_SUM
- 通讯辅助函数，如 GETBIT
- 控制模块，如 BSC、CSC

5. 链接

链接是功能块之间的连接。一个功能块输出可以连接多个功能块的输入，要连接的输入/输出必须要有相应的数据类型。链接允许与其他目标重叠，链接不能循环配置。循环必须通过实际参数来解决。

在 LD 段落中当触点靠近左汇流条时，自动生成链路。触点与触点、触点与线圈靠近时，也将自动生成链路。

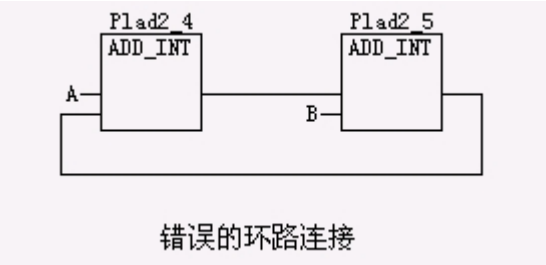


图 1-15 错误的环路连接

解决方法：

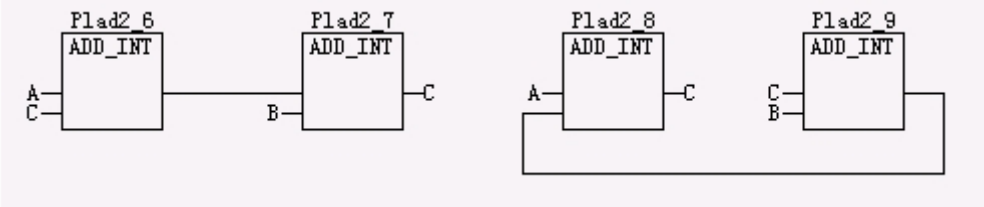


图 1-16 正确的环路连接

6. 执行次序

在 FBD 区段内那些输入只连接变量或位号或常数的模块，被称为区段的**起始模块**。

区段内有多个起始模块时，在图形区域中位置最上的模块称为**启动模块**。

区段的执行从启动模块开始。

FBD 区段内的执行次序由区段内的数据流决定。

FBD 段落中区段间的执行次序由区段的启动模块在段落图形中的位置决定。执行次序由上到下。

下图说明了功能块图的执行次序：

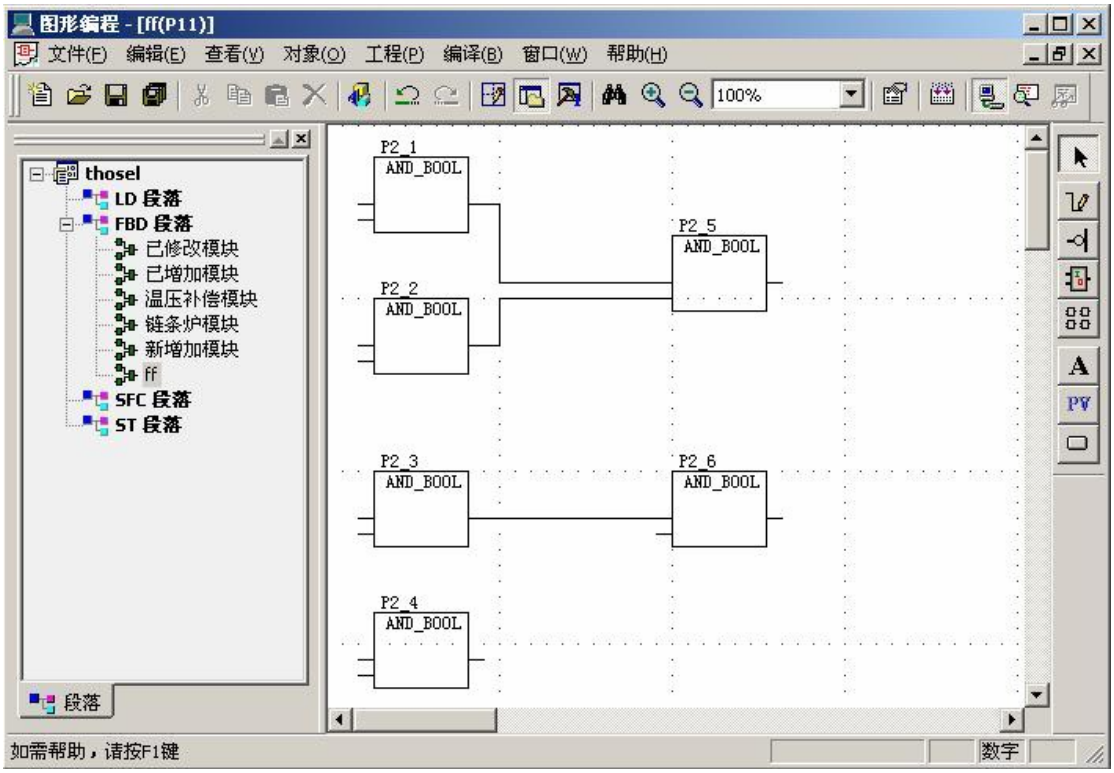


图 1-17 功能块图执行次序

7. FBD 语言编程

FBD 语言的编程分为创建新工程、创建新程序段、程序段的编程、工程的编译、连接等几个步骤：

- 创建一个新工程（project）

在图形编程软件编程环境中，点击“文件”菜单项，在弹出的菜单条中，选择“新建工程”，弹出一个“新建工程”对话框：

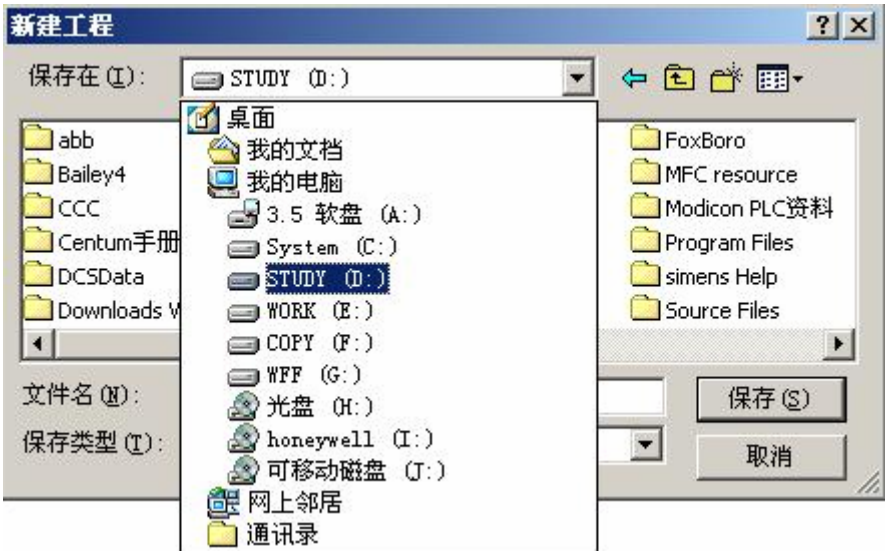


图 1-18 新建工程

用户可通过浏览，选择工程存放的位置，在“文件名”框内填上工程名（工程文件名以.prj

为后缀名），单击“保存”按钮，一个新工程就创建成功了。单击“取消”按钮，即放弃当前工程的创建工作。

在图形编程软件编辑环境中,可同时创建多个工程,这些工程将同时显示在工作空间(Workspace)中。

■ 创建一个新程序段

创建成功一个新工程后，还需要创建一个或多个程序段。

在图形编程编辑环境中，点击“文件”菜单项，在弹出的菜单条中，选择“新建程序段”，弹出一个“新建程序段”对话框：

图 1-19 新建程序段

在对话框的“程序类型”框中选择“功能块图”，即 FBD 语言类型；

接下来，用户可在程序和模块两种“段类型”之中选择其一：

- 程序——该类型的程序段可独立运行，程序段内可包括一个或多个模块；
- 模块——该类型的程序段相当于一般高级语言的子程序，需要别的程序调用方可发挥作用，不能独立运行。

用户在根据实际需要选择好段类型后，在“段名”框内填入程序段的名字，单击“确定”按钮，即成功创建一个新的程序段。单击“取消”按钮，即放弃当前程序段的创建工作。

如果用户已经创建或打开多个工程，则在创建一个新程序段时，要注意先将相应的工程激活（即确定先点击了该工程名），然后再创建，以免新建于其他的工程下。

■ 程序段的编程

FBD 编辑器用于将基本功能/功能块（EFB）和信号（变量）排列成功能块图。FBD 语言的编程类似于 LD 语言的编程，相比较而言，FBD 的编程不需要让逻辑行从母线以接点输入开始，同样可自由运用大量的功能模块等工具，遵循编程原则进行。

FBD 编程语言用来将程序段构造成一些基本功能和基本功能块、导出功能块和用户自定义功能块，可通过实际参数或链接进行连接。

- 基本元素

功能和功能块（联结时就变成了逻辑单元）

- 编程原则

a.变量必须先声明再使用；

b.输入输出类型必须一致；

c.功能块和变量可以有注释；

d. 不允许通过链路构成环路。

- 编程技巧

对于比较复杂或较大的 FBD 编程，宜先将程序分为简单的程序段，然后再逐段编程。

■ 编译、连接

用户编程后，将工程存盘，即可调用编译命令，对工程进行编译；反复调试，直至编译正确；然后，调用“设置相关控制站地址”命令，弹出对话框：

图 1-20 控制站地址设置

用户在对话框中填入下位机的主控卡地址，单击“确定”；随后，单击“连接”菜单项或按钮，即可完成与下位机的连接。

完成上述步骤后，调用“显示状态”功能，即可观察到程序实时运行的效果。

1.3.4 梯形图（LD）语言

根据 IEC1131-3，LD 编辑器将基本的功能/功能块（EFB）、线圈、触点和信号（变量、位号）组成梯形图（LD）。图形内可以自由放置基本元素和注释文本。部分 EFB 的输入可以扩展方便使用。

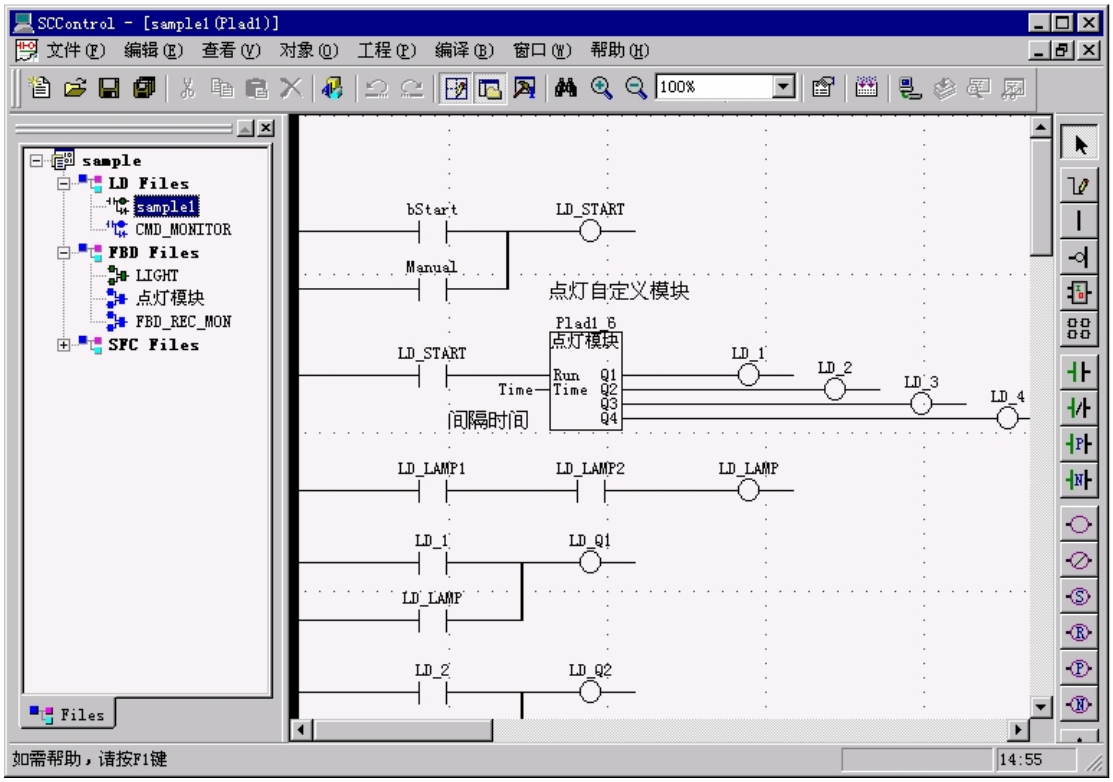


图 1-21 LD 编辑器

LD 段落的设计对应于继电器开关的梯级（rung）。图形的左边是汇流条，相应于梯级的相线（L）。只有直接或间接与相线有开关量相连的元素在编程期间被“扫描”。右汇流条缺省不画出。但可以认为所有的线圈和 FFB 开关量输出都连接到右汇流条上，从而建立电流回路。

在 LD 编程中，用户可以使用基本功能块和自定义功能块、线圈和触点。

用户用DFB 编辑器生成的自定义模块被放在自定义模块库中。DFB 加入模块库中后，就可以被各种语言编辑器使用。

在 LD 编辑器中，窗口的背景是逻辑坐标网格。当正在创建时，功能块或线圈触点在该网格的光栅中对准。如果发生与别的功能块重叠的情况，将会出现错误信息并且功能块不能创建。当实际参数在功能块输入/输出创建时，他们可能与别的目标重叠但不会破坏区段画面的界限。如果有一链路作为与别的功能块的连接，则该连接要进行检查。如果这是一个未经许可的连接，该链路将不生成。当触点靠近左汇流条时，自动生成链路。触点与触点、触点与线圈靠近时，也会自动生成链路。

除这些目标之外，也可以将注释文本和调试文本放入 LD 段落中。该文本目标的大小取决于文本的长度。根据文本大小，目标的大小可以通过在垂直以及在水平方向上更多的网格来进行扩展。文本不占用控制站内存。

下面将从以下几个方面分别作出介绍：

- LD 编辑器
- 触点
- 线圈
- FFB 模块
- 链接
- 执行次序
- 导入和导出
- 工程文件管理
- LD 编程

1. LD 编辑器

根据 IEC61131-3，LD 编辑器将基本的功能/功能块（EFB）、线圈、触点和信号（变量、位号）组成梯形图（LD）。图形内可以自由放置基本元素和注释文本。

LD 段落的设计对应于继电器开关的梯级（rung）。图形的左边是汇流条，相应于梯级的相线（L）。只有直接或间接与相线有开关量相连的元素在编程期间被“扫描”。右汇流条缺省不画出。但可以认为所有的线圈和 FFB 开关量输出都接到右汇流条上，从而建立电流回路。

在 LD 编程中，用户可以使用基本功能块和自定义功能块、线圈和触点。

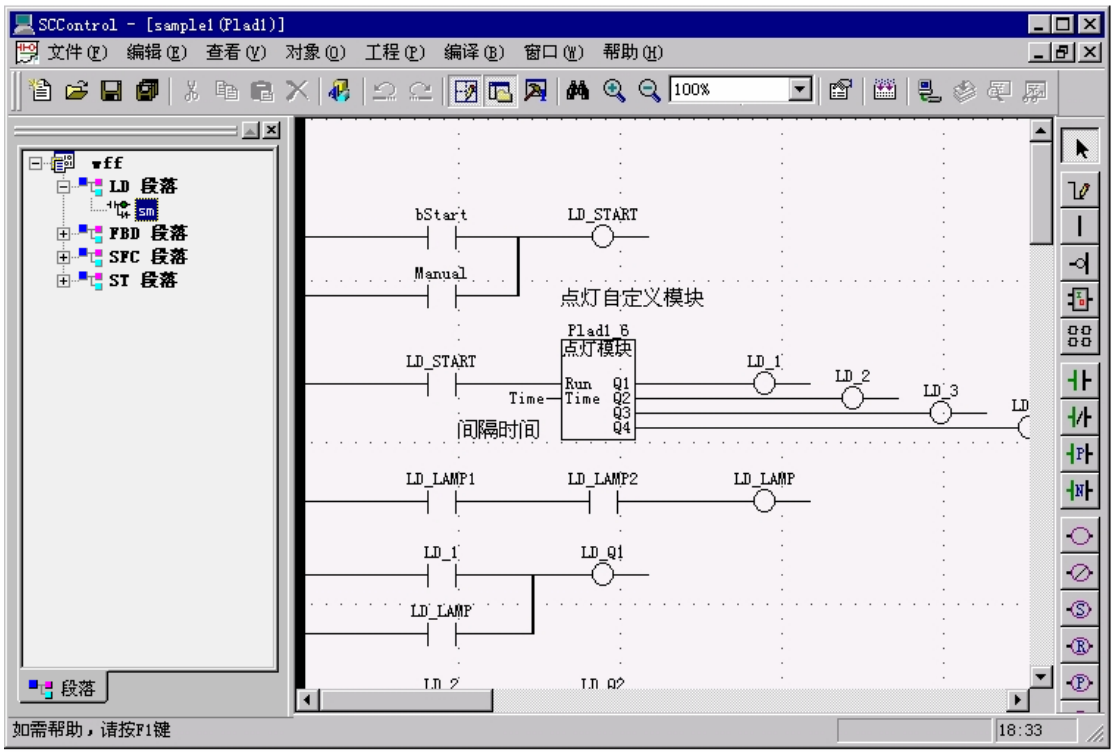


图 1-22 LD 编辑器

2. 触 点

触点是 LD 元素，它将状态传送至其右侧的水平链路。这一状态是在其左侧的水平链路中的状态与相关变量的状态进行布尔操作的结果。触点不改变相关变量的值。

- 常开触点
- 常闭触点
- 正跳变触点
- 负跳变触点

常开触点

在常开触点中，如果和触点相关的变量(IN1)为 ON 时，左链路的状态复制到右链路；否则右链路的状态为 OFF。

下图用梯形图和功能块图的方法描述了常开触点：

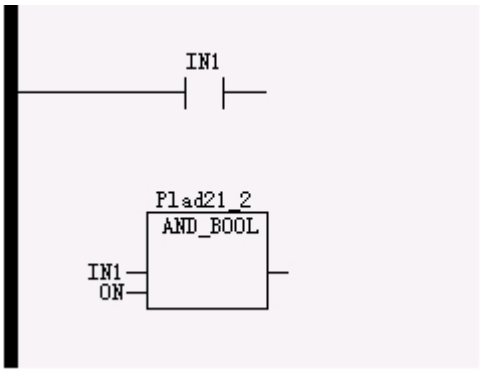


图 1-23 常开触点梯形图和功能块图

下图描述了常开触点在用垂直连接线连接后的原理示意：

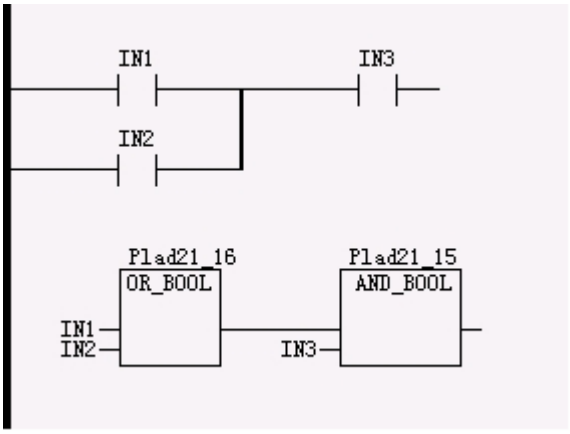


图 1-24 连接后的常开触点

常闭触点

在常闭触点中，如果和触点相关变量(IN1)的状态为 OFF 时，左链路的状态复制到右链路；否则右链路的状态为 OFF。

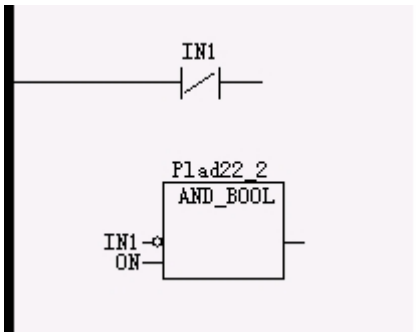


图 1-25 常闭触点

常闭触点对应于含两个输入的 AND_BOOL 功能，其中一个输入是反相的。

正跳变触点

在正跳变触点中，如果和触点相关 BOOL 变量(IN1)的状态从 OFF 跳变为 ON 时，同时左链路的状态为 ON 的话，则右链路在下一个程序周期为 ON；否则右链路的状态为 OFF。

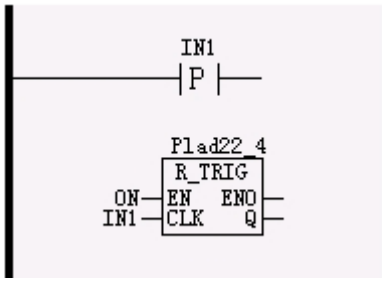


图 1-26 正跳变触点

负跳变触点

在负跳变触点中，如果和触点相关 BOOL 变量(IN1)的状态从 ON 跳变为 OFF 时，同时左链路的状态为 ON 的话，则右链路在下一个程序周期为 ON；否则右链路的状态为 OFF。

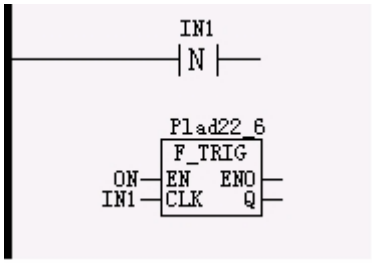


图 1-27 负跳变触点

3. 线圈

线圈是 LD 元素，它将其左侧的水平链路状态传送至其右侧的水平链路，相关变量的状态将保存。

- 常开线圈
- 常闭线圈
- 置位线圈
- 复位线圈
- 正跳变线圈
- 负跳变线圈

常开线圈

在线圈中，左链路的状态复制到相关的布尔变量和右链路。线圈通常跟在触点之后，但线圈之后也可以接触点单元。 常开线圈对应于 MOVE 功能。

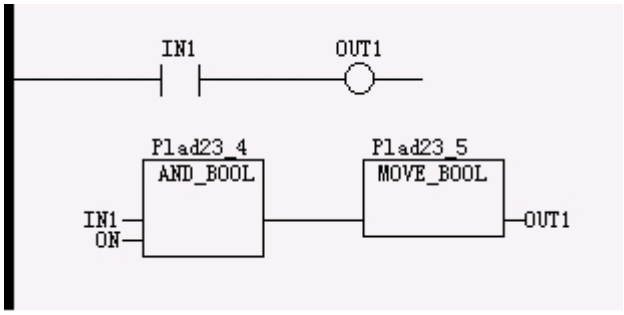


图 1-28 常开线圈

常闭线圈

在取反线圈中，左链路的状态复制到右链路。左链路的取反状态复制至相关的布尔变量(OUT1)。如果左链路为 OFF，则右链路将为 OFF，而相关变量将为 ON。

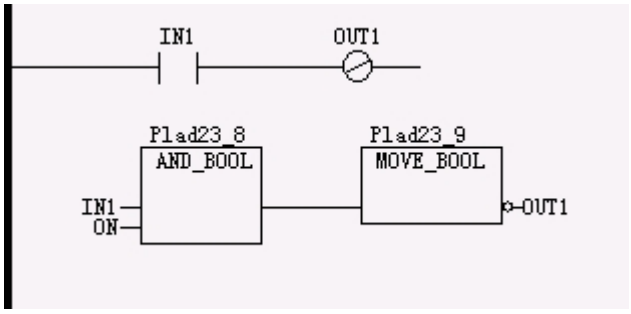


图 1-29 常闭线圈

取反线圈对应于带有反相输出的 MOVE 功能

置位线圈

在置位线圈中，左链路的状态复制至右链路。如果左链路为 ON，则相关的布尔变量(OUT1)置为 ON；否则相关的布尔变量保持不变直至程序或人工修改其值。相关布尔变量能够借助复位线圈复位。置位线圈对应于输入固定为 ON 的 MOVE 功能。

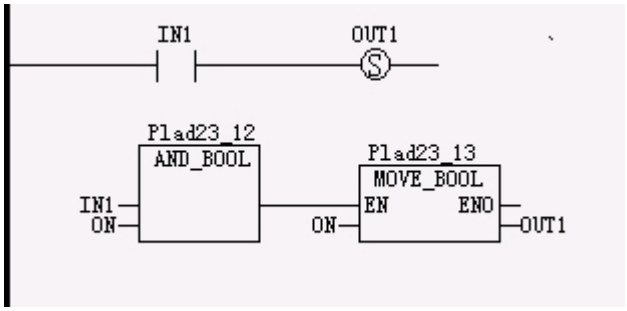


图 1-30 置位线圈

复位线圈

在复位线圈中，左链路的状态复制至右链路。如果左链路为 ON，则相关的布尔变量(OUT1)置为 OFF；否则保持不变直至程序或人工修改其值。相关布尔变量能够借助置位线圈置位。复位线圈对应于输入固定为 OFF 的 MOVE 功能。

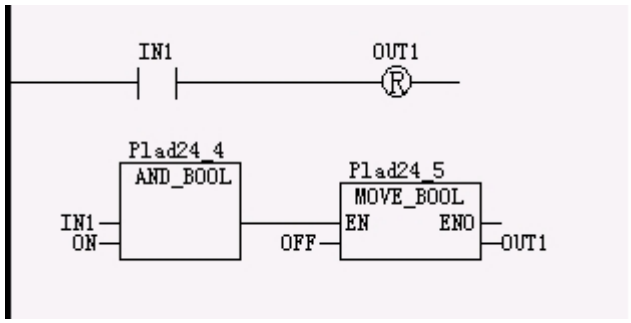


图 1-31 复位线圈

正跳变线圈

在正跳变线圈中，左链路的状态复制至右链路。如果左链路从 OFF 跳变为 ON，则相关的布尔变量(OUT1)将在下一个程序周期内为 ON。正跳变线圈对应于 R_TRIG 功能块。

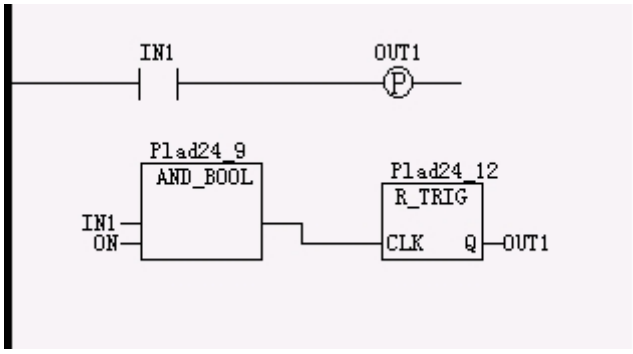


图 1-32 正跳变线圈

负跳变线圈

在负跳变线圈中，左链路的状态复制至右链路。如果左链路从 ON 跳变为 OFF，则相关的布尔变量(OUT1)将在下一个程序周期内为 ON。负跳变线圈对应于 F_TRIG 功能块。

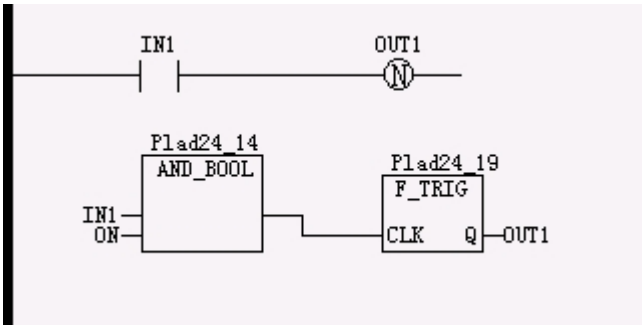


图 1-33 负跳变线圈

4.链接

链接是功能块之间的连接。一个功能块输出可以连接多个功能块的输入，要连接的输入/输出必须要有相应的数据类型。链接允许与其他目标重叠，链接不能循环配置。循环必须通过实际参数来解决。

在 LD 段落中当触点靠近左汇流条时，自动生成链路。触点与触点、触点与线圈靠近时，也将自动生成链路。

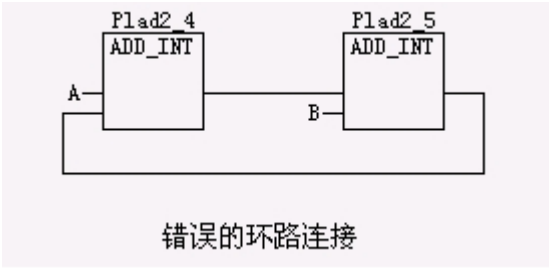


图 1-34 错误的环路连接

解决方法：

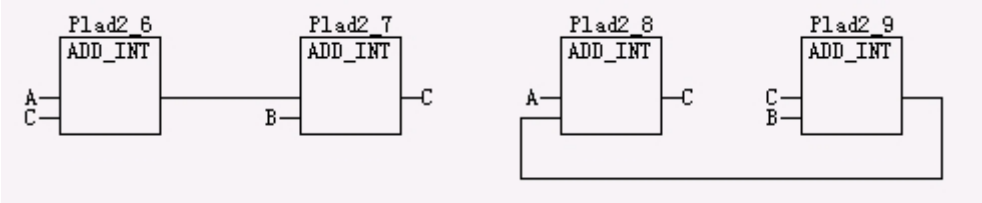


图 1-35 正确的连接

5. 执行次序

在 LD 区段输入只连接变量或位号或常数或左汇流条的被称为区段的**起始模块**。

区段内有多个起始模块时，在图形区域中位置最上的模块称为**启动模块**。

LD 区段从启动模块开始执行。

LD 区段内的执行次序由区段内的数据流决定。

LD 段落中区段间的执行次序由区段的启动模块在段落图形中的位置决定。执行次序由上到下。如下图所示：

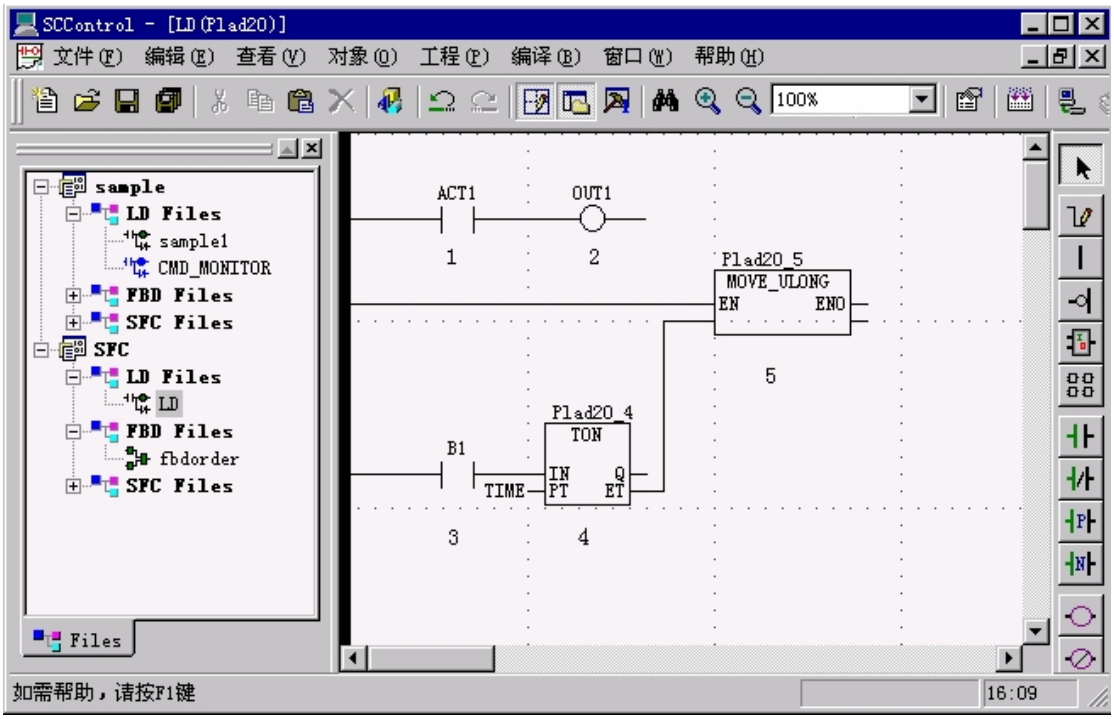


图 1-36 执行次序

6. LD 语言编程

LD 语言的编程分为创建新工程、创建新程序段、程序段的编程、工程的编译、连接等几个步骤：

■ 创建一个新工程（project）

在 SCControl 软件编程环境中，点击“文件”菜单项，在弹出的菜单条中，选择“新建工程”，弹出一个“新建工程”对话框：

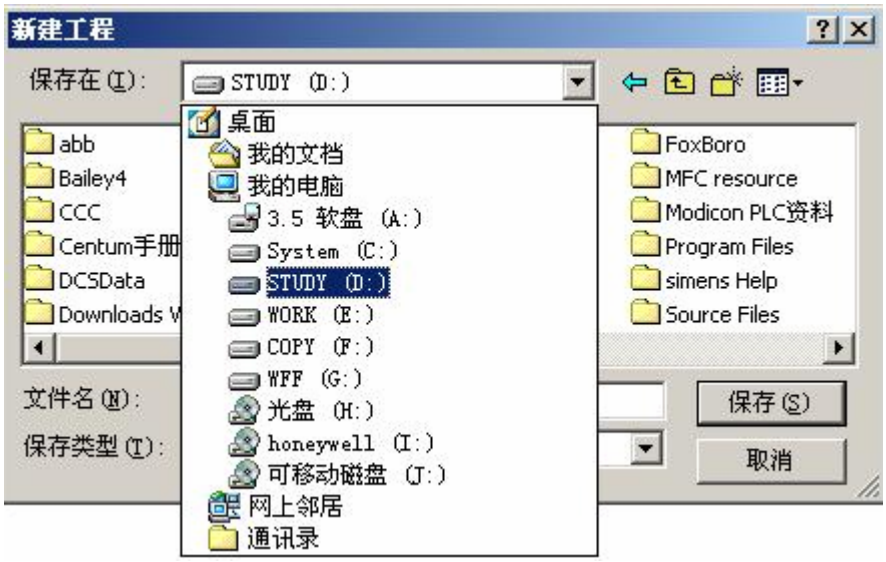


图 1-37 新建工程

用户可通过浏览，选择工程存放的位置，在“文件名”框内填上工程名（工程文件名以.prj 为后缀名），单击“保存”按钮，一个新工程就创建成功了。单击“取消”按钮，即放弃当前工程的创建工作。

在 SCControl 软件编辑环境中,可同时创建多个工程,这些工程将同时显示在工作空间

(Workspace)中。

■ 创建一个新程序段

创建成功一个新工程后，还需要创建一个或多个程序段。

在 SCControl 编辑环境中，点击“文件”菜单项，在弹出的菜单条中，选择“新建程序段”，弹出一个“新建程序段”对话框：

图 1-38 新建程序段

在对话框的“程序类型”框中选择“梯形图”，即 LD 语言类型；

接下来，用户可在程序和模块两种“段类型”之中选择其一：

- 程序——该类型的程序段可独立运行，程序段内可包括一个或多个模块；
- 模块——该类型的程序段相当于一般高级语言的子程序，需要别的程序调用方可发挥作用，不能独立运行。

用户在根据实际需要选择好段类型后，在“段名”框内填入程序段的名字，单击“确定”按钮，即成功创建一个新的程序段。单击“取消”按钮，即放弃当前程序段的创建工作。

如果用户已经创建或打开多个工程，则在创建一个新程序段时，要注意先将相应的工程激活（即确定先点击了该工程名），然后再创建，以免新建于其他的工程下。

■ 程序段的编程

LD 编辑器用于将基本功能/功能块（EFB）、触点、线圈和信号（变量）整理成梯形图。作为一种图形编程工具，SCControl 可被用户运用软件提供的大量功能模块、线圈、触点等，通过选取、拖动、放置、移动等操作，以及利用基本功能块创建自定义功能块等轻松地完成一系列编程工作。

- 基本元素

  功能和功能块、触点、线圈

- 编程原则

  a.变量必须先说明再使用；

  b.梯形图的每一逻辑行必须从左边母线以接点输入开始；

  c.接点的使用次数不受限制；

  d.EFB、触点、线圈和变量可以有注释；

  e.生成链路时，允许与其它链路和目标重叠、交叉。

- 编程技巧

  对于比较复杂或较大的 LD 编程，宜先将程序分为简单的程序段，然后再逐段编程。

■ 编译、连接

用户编程后，将工程存盘，即可调用编译命令，对工程进行编译；反复调试，直至编译正确；然后，调用“设置相关控制站地址”命令，弹出对话框：

图 1-39 设置控制站地址

用户在对话框中填入下位机的主控卡地址，单击“确定”；随后，单击“连接”菜单项或按钮，即可完成与下位机的连接。

完成上述步骤后，调用“显示状态”功能，即可观察到程序实时运行的效果。

1.3.5 顺控图 SFC

SFC 编辑器实现的元素和结构

步 Step

步是一段控制程序，用来执行规定的动作(Action)。动作是一组赋值语句，可以给常规控制模块的参数赋值。步有 3 种类型，起始步、常规步、结束步。每个 SFC 模块有一个起始步和一个结束步。常规步可以有多个。起始步在 SFC 模块启动时执行，结束步在 SFC 模块正常运行结束或放弃执行时执行，以保证 SFC 模块结束在固定状态。

转换 Transition

转换是一段条件判断程序，用来实现步间的运行转换。转换中包含条件。当条件判断满足，转换的输出值为 TRUE，转换之前的步执行结束，然后执行转换之后的步。

链路 Link

用来连接步和转换。

文本 Text

Text 是注释文本，用来使 SFC 模块增强可读性。文本可以放在绘图区中任何地方。

顺序结构

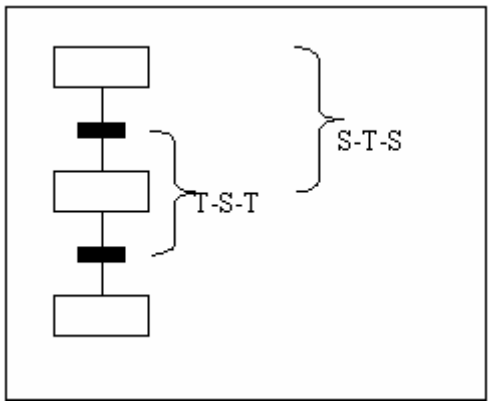


图 1-40 顺序结构

顺序结构用来实现**步**的顺序执行。顺序结构按结构划分可归结为 S-T-S 型和 T-S-T 型。S-T-S 型结构两头都是**步**，单独一个**步**也是 S-T-S 型结构。T-S-T 型结构两头都是**转换**，单独

一个转换也是 T-S-T 型结构。

在顺序结构中，某步激活时，它下面的转换开始判断内部的条件是否全满足，如果没有全满足，不断执行此步，如果内部的条件全满足则转到下一步继续执行。

并联结构

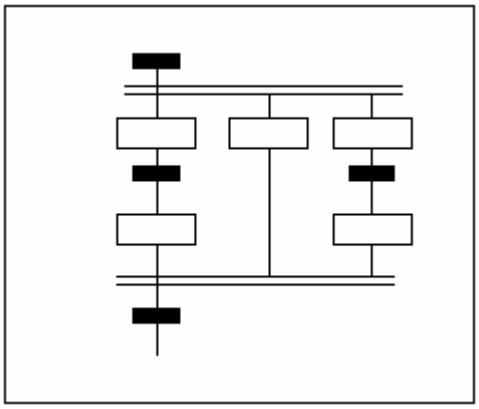


图 1-41 并联结构

并联结构用来实现几个顺序支路的并联执行。适用于几个操作需要同时进行的场合。并联结构以图形化表示的并联分支（双横线）开始，结束于以双横线表示的并联接合。在并联结构中，每个顺序分支必须是 S-T-S 结构。并联结构上面只有一个公共转换是允许的。当此公共转换为真时，各顺序支路并行启动，此后，各顺序支路彼此独立处理。当所有的顺序支路都处理完，才判断并联接合后的转换。

选择结构

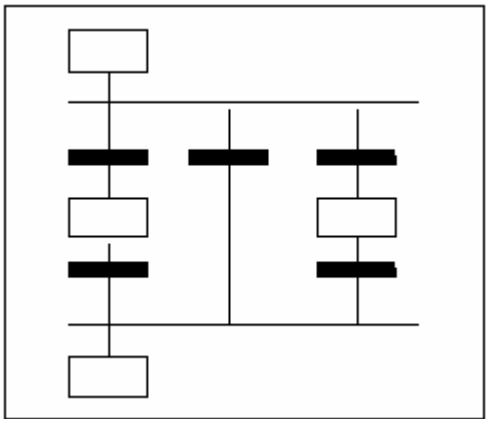


图 1-42 选择结构

选择结构实现在几个顺序支路中任意选择一个执行的功能。使用在当某个操作完成后，可根据当前工业过程的状态来选择一个操作的场合。选择结构以图形化表示的单横线选择分支开始，结束于以单横线表示的选择接合。在选择结构中，每个分支必须是 T-S-T 结构。如果同时有几个分支的条件都满足，则系统按从左到右的优先级选择一路执行。从同一个选择分支引出的顺序支路必须交汇到同一个选择接合上，或通过跳转跳出。

跳转

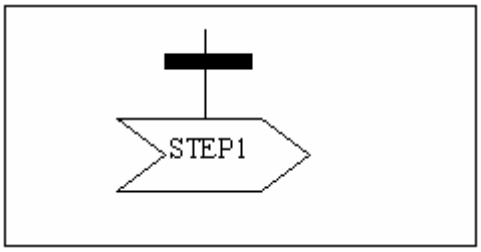


图 1-43 跳转

跳转允许程序从不同的区域继续运行。跳转有两类结构，顺序跳转和循环。

顺序跳转实现在选择分支中在条件满足时，跳过几步继续运行。如下图所示：执行完 S1 后，可以在条件满足的情况下，跳过 S2、S3，直接执行到 S4。

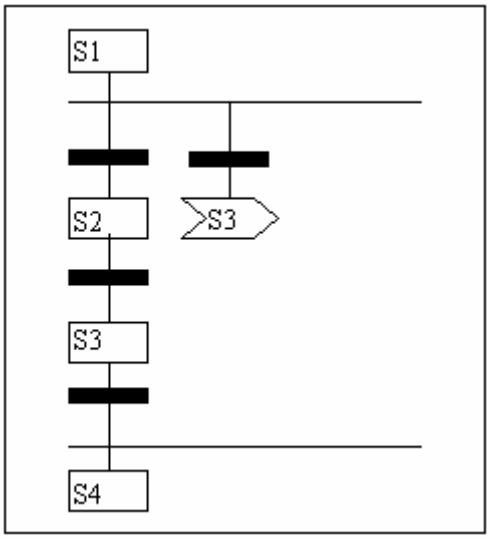


图 1-44 顺序跳转

循环实现重复操作。

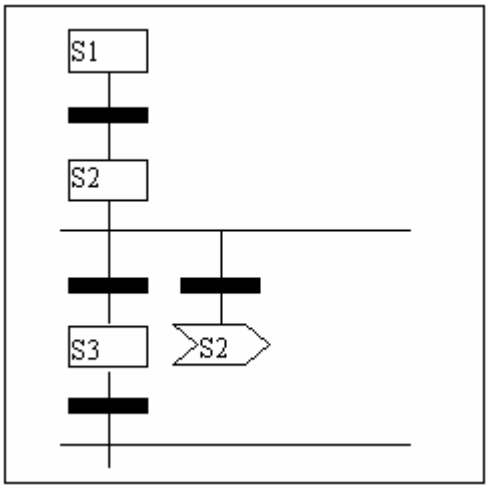


图 1-45 循环

如上图所示：执行完 S1 后执行 S2，执行完 S2 后，只要条件满足，就可以不执行接下去的 S3，而再一次执行了 S2。如此可重复执行多次 S2，直到重复的条件不再满足，方开始执行下面的 S3。

跳转的图形是一个箭头，内有跳转到的步名称。

1. 概述

SFC 编辑器实现的元素和结构包括了：

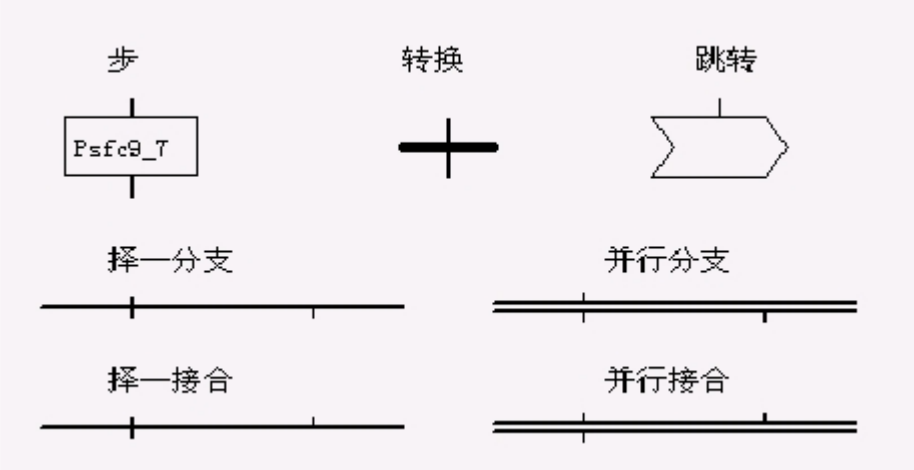


图 1-46 SFC 编辑器元件和结构

2. 步（step）

步是控制流程中相对独立的一组操作的集合,在步中可以定义任意数量的各种类型的操作（Action），以此来实现对流程的控制。

步在激活时才执行相应的操作,它只有在紧接它上面的转换条件满足时才被激活，步中的操作全都执行完毕之后如果紧接在它后面的转换条件满足才退出激活状态。

步的上面只能接转换、并行分支或择一接合；步的下面只能接转换、并行接合或择一分支。

步有三种类型：起始步、普通步、终止步，如下图所示

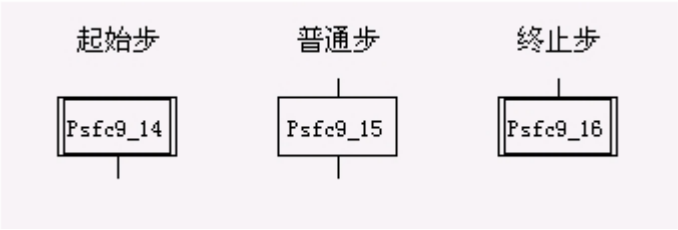


图 1-47 步的三种类型

在一幅 SFC 图中，起始步和终止步必须有且各只有一个。SFC 的执行从起始步开始到终止步结束。

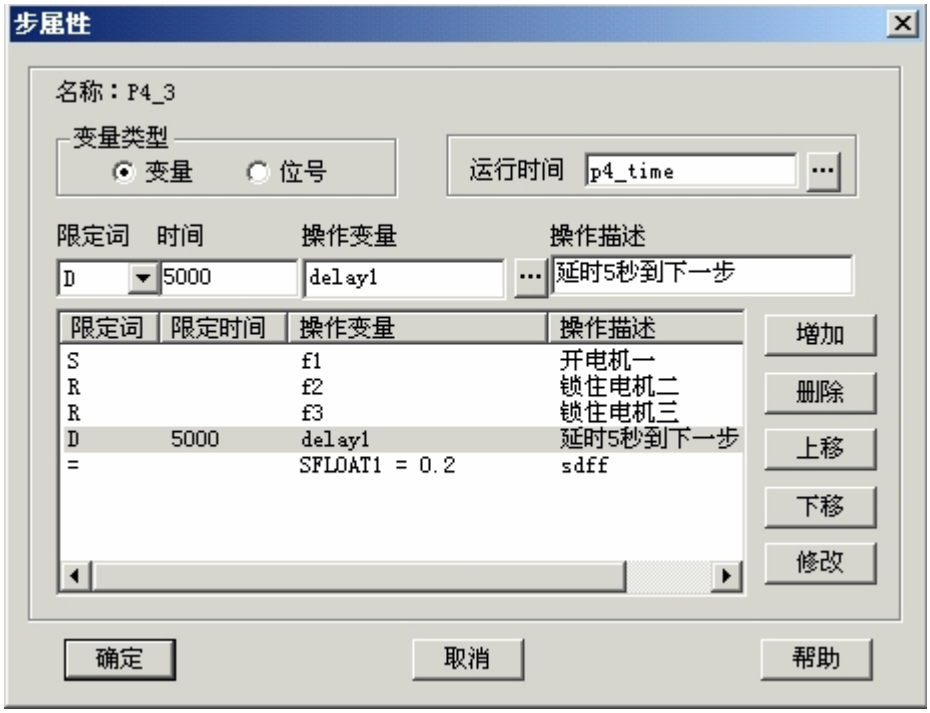


图 1-48 步属性

运行时间

指定将步的激活时间赋给 ULONG 类型的变量用于显示。

变量类型

指定选择变量时使用系统位号浏览还是使用变量浏览。

限定词

指定当前操作的类型

时间

指定当前操作的限定时间

操作变量

用于指定操作

操作描述

用于对当前操作添加注释。

通过增加删除按钮在当前步中加入或删除操作。

选择操作后可以通过上移和下移更改操作的执行次序。

通过修改按钮可以修改当前操作。

3. 转换（Transition）

转换用来指明将控制从一个步转移到其它步的条件。

当转换条件满足时，紧接在前的步从激活态变成不激活态。然后紧接在后的步将从不激活态转变成激活态。

只有当所有紧接在前的步都在激活状态时，转换的条件才被测试。

转换条件由一个布尔变量或布尔表达式定义。

转换的上面只能接步、择一分支、并行接合。转换的下面只能接步、择一接合、并行分支、跳转。

4. 跳转（Jump）

跳转允许程序从不同的步继续执行。根据跳转对象的不同，可以构成顺序跳转和顺序环路，不能在不同的并行区域间跳转。

下图为顺序跳转。顺序跳转主要用于故障处理。

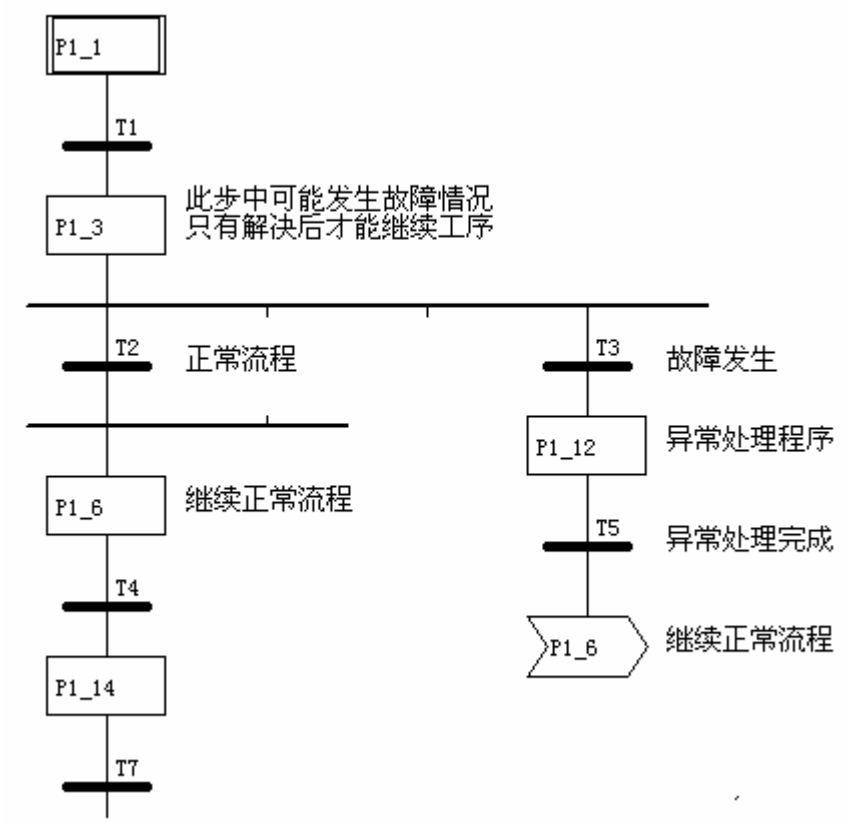


图 1-49 顺序跳转

下图为顺序环路：

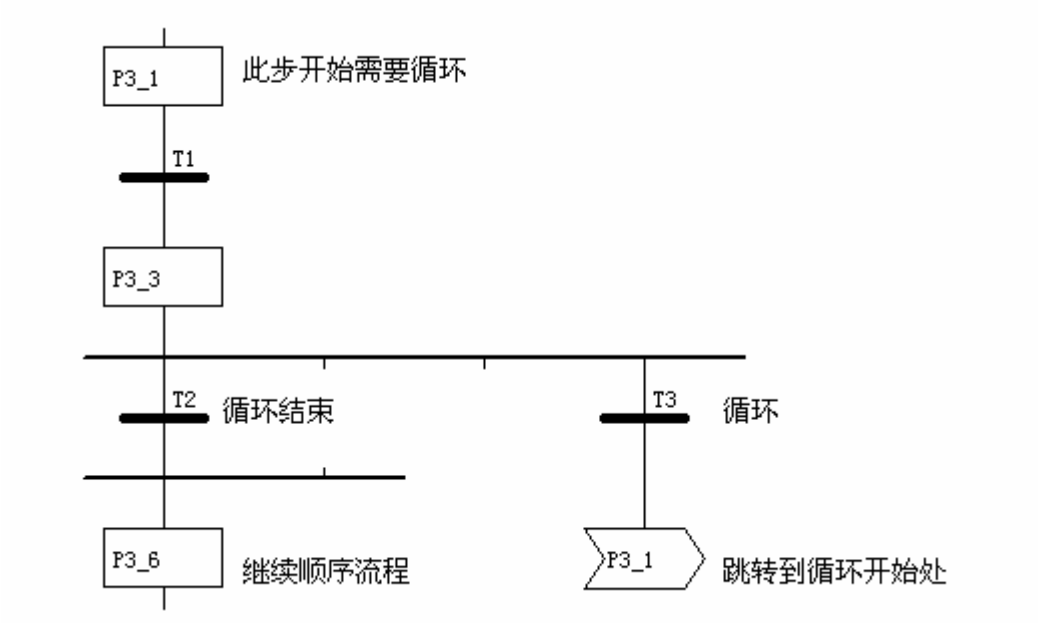


图 1-50 顺序环路

5. 择一分支（Alternative Branch）

择一分支提供了在 SFC 程序中实现条件控制的控制流程选择执行的方法。

在择一分支结构内只能有一个分支被激活。

分支跳转的优先级从左到右（如果几个分支的判断变量同时满足，那么执行最左边分支）。

择一分支和择一接合必须一一对应。

分支必须结束于同一择一接合或者结束于跳转。

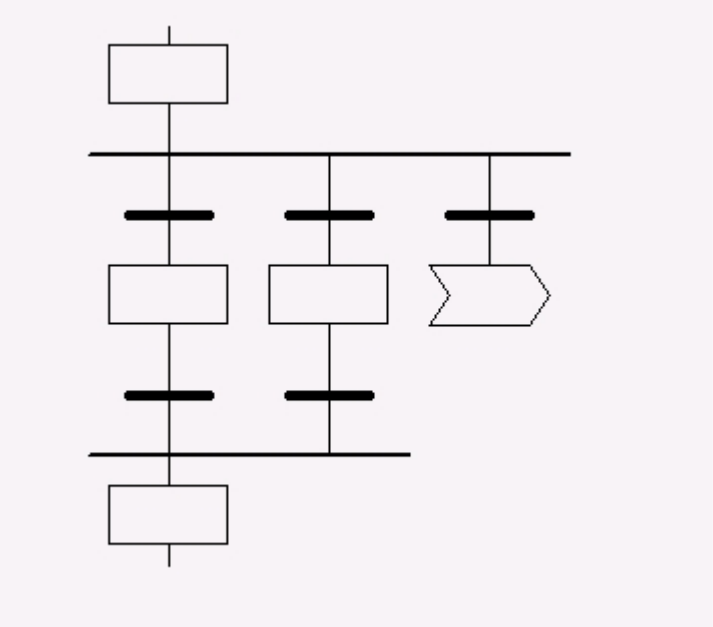


图 1-51 择一分支

6. 并行分支

并行分支使流程中几个子流程同时进行。各分支的执行同时进行，不相互影响。只有当所有的分支的最后一步都激活时，才测试并行接合紧接的转换的条件是否满足。

并行分支和并行接合必须一一对应。

在并行结构内部的跳转不能跳到并行结构的外部。

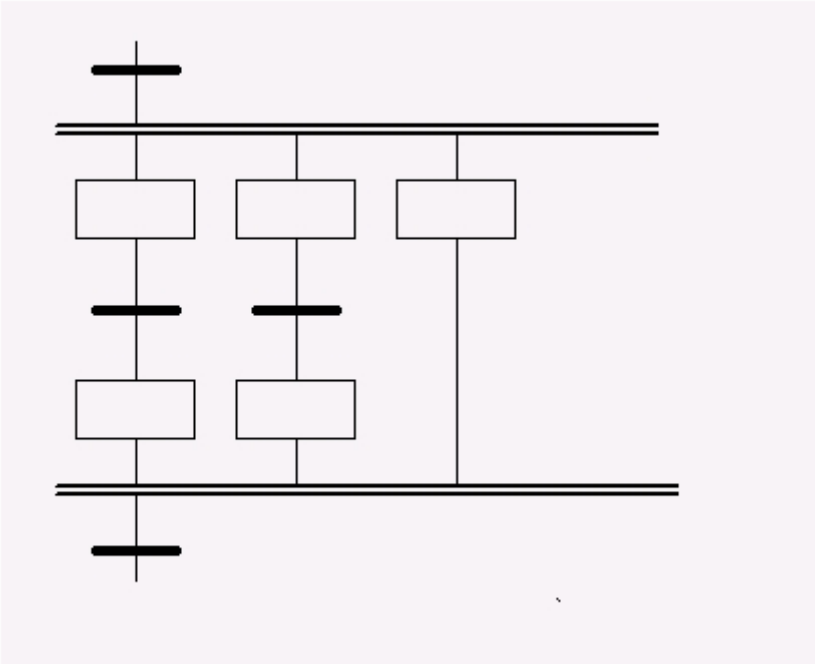


图 1-52 并行分支

7. 操作

操作是对系统信号（变量、位号）进行操纵的描述。

一个步中可以有 0 个或多个操作。操作有多种类型，操作类型由操作限定词来描述。操作可以是一个布尔变量（操作变量），也可以是一个赋值表达式。

在步属性窗口内可以编辑操作。

图形编程软件 SFC 编辑器提供以下符合 IEC1131-3 标准的操作限定词：

1) N 操作在步的整个激活期间激活，随着步退出激活状态恢复成不激活状态
2) S 操作在步激活后将一直保持激活
3) R 操作在步激活后将一直保持在不激活状态
4) L 操作在步激活后在限定的时间内保持激活，超出时间恢复成不激活状态
5) D 操作在步激活后经过限定的时间后，变为激活状态，随着步变成不激活状态，操作恢复成不激活
6) P 操作在步激活后只激活一个程序扫描时间，然后恢复成不激活状态
7) DS 操作在步激活后经过限定的时间后，变为激活状态，并一直维持激活状态

在以上类型的操作中，操作变量只能定义为布尔量，其中使用 L 、D、DS 操作限定词时必须指定限定时间。**限定时间的单位是 ms**。

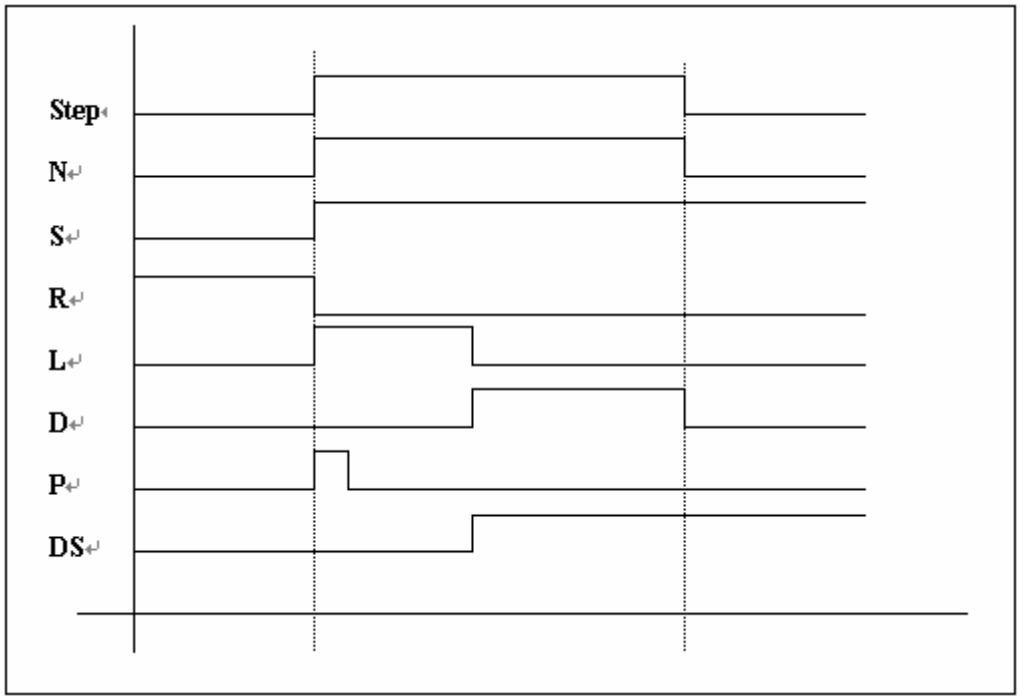


图 1-53 操作

图形编程软件 SFC 编辑器提供了扩展的操作限定词：

= 赋值操作限定词。表示在步的整个激活期间赋值操作一直进行直到步退出激活状态恢复成不激活状态。

在使用赋值操作限定词时，必须指定赋值表达式才可以增加操作，如需复杂表达式可以双击该操作编辑 ST 语言程序。

SFC 控制变量

可以设置以下控制变量来控制 SFC 程序的运行：

- 运行控制
- 复位
- 禁止转换
- 强制步进
- 操作使能

运行控制变量为 ON 时，SFC 程序正常执行。为 OFF 时，所有其它控制变量都无效，SFC 程序停止运行。程序必须指定一个运行控制变量。

复位变量状态为 ON 时，SFC 程序起始步被设置为激活步，其它步都强制变为不激活状态，顺控程序从头开始重新执行。当运行变量为 OFF 时，复位变量无效。

禁止转换变量为 ON 时，当前激活步将一直保持执行而不管紧接的转换条件是否满足，转换条件测试将不进行。禁止转换变量受运行变量和复位变量的影响。

强制步进变量为 ON 时，当前激活步不管转换条件是否满足，都变为不激活状态，按顺序的下一步变为激活状态。强制步进变量受以上所有变量的影响。

操作使能变量操作：操作使能变量为 ON 时，步中的操作才被执行。

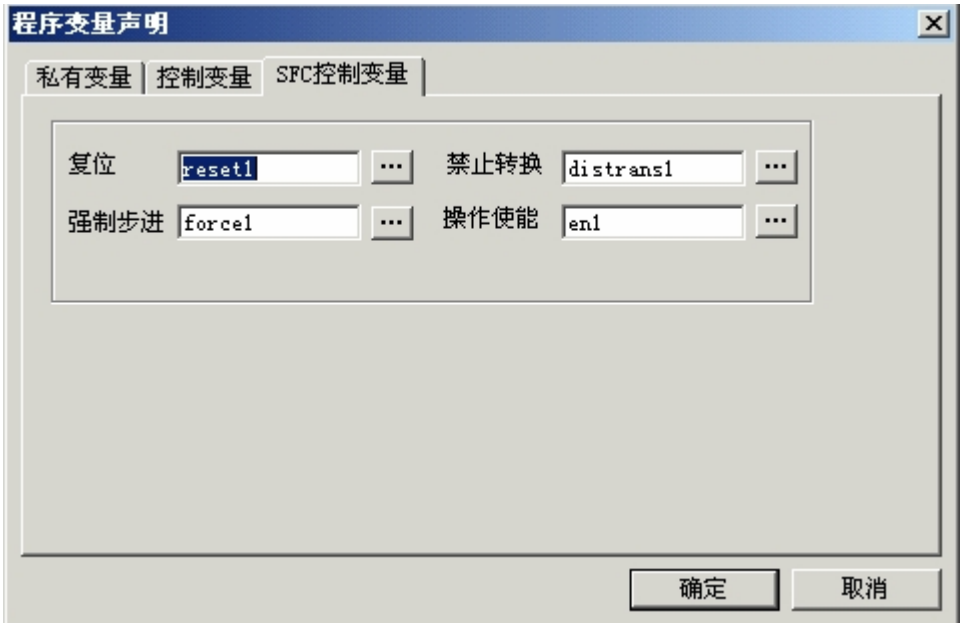


图 1-54 程序变量声明

1.3.6 ST 语言概述

ST 语言在图形编程中和其它图形编程语言组合使用。实现了 IEC1131-3 标准的一个子集。

使用方法

在工程中加入 ST 语言段落。

可以在梯形图和功能块图中插入文本代码模块。在模块中用 ST 语言编程。

可以在顺控图中的步的操作中使用 = 操作限定词，然后可以用 ST 语言编程。

在顺控图的转换条件中可以使用 ST 语言的逻辑表达式来指定条件。

在 SFC 编辑器中，当指定转换条件时可以使用 ST 语言逻辑表达式。

在操作中用 = 操作符可以使用 ST 的语句，双击该条操作即可弹出编辑界面。

注意：使用循环语句会明显增加编译时间，为获得更快的编译速度建议尽可能不使用 WHILE 语句并且少使 REPEAT/FOR 语句。

1.ST 语言语法

1) ST 语言语法

ST 语言在图形编程软件中和其它图形编程语言组合使用。实现了 IEC61131-3 标准的一个子集。支持多种数据类型，支持函数、结构和数组，可以操作各种系统变量。

表达式

表达式为变量、操作符、常量、函数的组合，求值结果为单个值。

表达式的求值按运算符的优先级进行，优先级高的运算符先被处理。相同优先级的运算符按从左到右的顺序执行。

以下为几个合法的表达式举例：

A + B * (C-3) + FUNC1(2,D)

B1

FUNC1()

使用方法

变量、函数等标识符的命名必须满足下列条件：

✓ 以英文字母开头；

续以英文字母、数字或下划线；

✓ 字符长度最多为 24 个字符。

标识符包括变量、函数、功能块、常数。

关键字

关键字	描述
CASE…OF…ELSE…END_CASE	CASE 语句
BOOL WORD DWORD INT LONG UINT ULONG SFLOAT FLOAT	数据类型
EXIT	终止循环
FALSE	逻辑假
FOR…TO…BY…DO…END_FOR	FOR 语句
FUNCTION…END_FUNCTION	函数定义
FUNCTION_BLOCK END_FUNCTION_BLOCK	功能块定义
IF…THEN…ELSEIF…ELSE…END_IF	IF 语句
ON	逻辑真
OFF	逻辑假
REPEAT…UNTIL…END_REPEAT	REPEAT 语句
RETURN	函数返回
TRUE	逻辑真
VAR…END_VAR VAR_INPUT…END_VAR VAR_OUTPUT…END_VAR	变量定义
WHILE…DO…END_WHILE	WHILE 语句

运算符

按运算优先级从高到低有：

运算符	描述	类型	优先级
()	表达式运算		9
.	取结构成员		8
[]	取数组成员		8
-	单目负		7
NOT	取反	逻辑运算	7
* (MUL)	乘	算术运算	6
/ (DIV)	除	算术运算	6
MOD	取余	算术运算	6
+ (ADD)	加	算术运算	5
- (SUB)	减	算术运算	5
>	大于	比较运算	4
>=	大于等于	比较运算	4
<=	小于等于	比较运算	4
<	小于	比较运算	4

=	等于	比较运算	4
<>	不等于	比较运算	4
AND	与	逻辑运算	3
XOR	异或	逻辑运算	2
OR	或	逻辑运算	1

语句

以下为允许的语句列表。

No.	语句	例子
1	赋值语句	A = B; A = B + 1;
2	函数调用、功能块调用	A = FUNC(P1,P2); FB1(IN1,OUT1,OUT2);
3	RETURN	A = FUNC(P1,P2); RETURN A;

4	IF	IF A > 0 THEN
		B = 1;
		ELSEIF A > -5 THEN
		B = 2;
		ELSE
		B = 3;
		END_IF;

5	CASE	TW = FUNC1(); CASE TW OF 1 : I = 1; 2 : I = 2; ELSE I = 3; END_CASE;

6	FOR	J = 10;
		FOR I = 1 TO 100 BY 2 DO
		IF B1 THEN
		J = 1;
		EXIT;
		END_IF;
		END_FOR;

7	WHILE	J = 1; WHILE J <= 100 AND B1 DO J = J + 2; END_WHILE;
8	REPEAT	J = 1; REPEAT J = J + 2; UNTIL J = 101 OR B1 END_REPEAT;

9	EXIT	J = 1;
		WHILE J <= 100 AND B1 DO
		J = J + 2;
		IF J>= 50 THEN
		EXIT;
		END_IF;

		END_WHILE;
10	EMPTY	FOR I = 1 TO 100 BY 2 DO ; END_FOR;

赋值语句

赋值语句将“=”右边表达式的值赋给左边的变量。

函数调用语句

函数和功能块的调用包括函数名或功能块名加小括号对，括号内为参数，参数间由逗号隔开。

函数的调用规则：

ret = Func(in1,in2)；（*作为表达式返回值*）

Func(in1,in2)；（*作为子程序处理*）

功能块的调用规则：

调用功能块时要严格按照输入输出顺序，先输入输入参数，再输入输出参数，参数顺序按照定义时的顺序。

输出参数必须是变量： FuncBlock(in1,in2,out1,out2)。

2) ST 语言函数和功能块

函数定义

只有一个输出（变量类型可以自行确定），根据输入可以唯一确定输出。

功能块定义

有多个输出，或输出不但和当前输入有关还和上次内部状态有关。

除系统内部的函数和功能块外，用户可以自定义函数和功能块。自定义的函数和功能块可以在工程内 LD/FBD 段落中调用。

函数的调用规则

ret = Func(in1,in2); （*作为表达式返回值*）

Func(in1,in2); （*作为子程序处理*）

功能块的调用规则

调用功能块时要严格按照输入输出顺序，先输入输入参数，再输入输出参数，参数顺序按照定义时的顺序。

输出参数必须是变量。

FuncBlock(in1,in2,out1,out2);

使用限制

ST 的 FUNCTION 模块

可以调用其它 ST FUNCTION,允许嵌套

可以调用标准函数

ST 的 FUNCTION_BLOCK 模块

可以调用 ST FUNCTION

可以调用其它 ST FUNCTION_BLOCK,但是所调用的 FUNCTION_BLOCK 不允许嵌套

可以调用标准函数

TEXTCODE 模块

可以调用 ST FUNCTION

可以调用 ST FUNCTION_BLOCK

可以调用标准函数

可以调用各种编程语言生成的功能块

编译速度

由于当前 SCControl 所使用的 ARM 编译器（SDT251）的缘故，使用循环语句，尤其是其中的 WHILE 语句，会明显增加编译时间，为获得更快的编译速度建议尽可能不使用 WHILE 语句以及少使用 REPEAT/FOR 语句。

3) FUNCTION

```
FUNCTION FUNC3 : BOOL
VAR_INPUT
        IN1:BOOL;
END_VAR
VAR
        TEMP1 : BOOL;
END_VAR
    FUNC3 := DoSomething();
END_FUNCTION
```

说明：

VAR_INPUT/END_VAR 用于说明功能块的输入变量；

VAR/END_VAR 用于说明功能块内部的临时变量（临时变量存储在系统堆栈中，不能维持状态到下一周期）；

各类变量声明的次序不能颠倒；

调用时要严格按照声明的先后次序。

4) FUNCTION_BLOCK

```
FUNCTION_BLOCK FB3
VAR_INPUT
        IN1:BOOL;
END_VAR

VAR_OUTPUT
        OUT1 : BOOL;
END_VAR

VAR
        TEMP1 : BOOL;
END_VAR

DoSomething();

END_FUNCTION_BLOCK
```

说明：

VAR_INPUT/END_VAR 用于说明功能块的输入变量；

VAR_OUTPUT/END_VAR 用于说明功能块的输出变量；

VAR/END_VAR 用于说明功能块内部的临时变量（临时变量存储在系统堆栈中，不能

维持状态到下一周期）；

各类变量声明的次序不能颠倒；

不能对输出变量进行自操作；

功能块调用时要严格按照声明的先后次序。

5) ST 语言程序实例

赋值语句

赋值语句将“=”右边表达式的值赋给左边的变量。

```
A = B ;
A = B + C;
A = B * C + D;
A = B AND C AND D OR E;
A = AND_DWORD ( B,C);
```

IF 语句

```
IF （A AND (B > C) OR (E > F + 1) ） THEN
        AA = BB;
ELSEIF ( B > G) THEN
            AA = CC;
ELSE
    AA = DD;
END_IF;
```

IF 语句规定了一组语句在规定的逻辑表达式为 TRUE 时执行。当逻辑表达式为 FALSE 时，这些语句不被执行，或在 ELSE(ELSEIF)中规定的另一组语句被执行。

CASE 语句

```
CASE  A  OF
1:
    AA = BB;
2:
    AA = CC;
3:
        AA = DD;
ELSE
            AA = EE;
END_CASE;
```

CASE 语句规定了整数类型的选择项，以及选择项在不同的值时的几组语句组。当选择项等于某个规定的值时，相应的语句组被执行，当没有规定的值符合时在 ELSE 中的语句组将被执行（在 CASE 语句中定义了 ELSE 分支）。

FOR 语句

```
FOR I = 1 TO 100 BY 2 DO
        DOSOMETHING();
END_FOR;
```

在以上 FOR 语句中，I 为控制变量，1 为初始值，100 为终止值，2 为步进值。在 FOR 语句中控制变量的初始值、终止值、步进值必须是相同的整型。步进值缺省为 1。终止条件的判断一开始就进行，当初始值大于终止值时，规定的语句组一次都不会执行。

WHILE 语句

WHILE 条件 DO

语句组

END_WHILE;

条件的判断一开始就进行，如条件一开始就变 FALSE 时，规定的语句组一次都不会执行。

REPEAT 语句

REPEAT

语句组

UNTIL 终止条件

END_REPEAT;

终止条件的判断在语句组执行一次后才进行，所以规定的语句组至少会执行一次。当终止条件成 TRUE 时，循环被终止。

2. ST 可调用函数列表

• 算术运算
• 比较函数
• 转换函数
• 逻辑运算
• 数学函数
• 选择函数
• 通讯辅助函数
• 系统时间函数
• 特殊函数
• 其他函数
• 输入处理函数

1) 算术运算

- 加法函数

这些函数的功能是将输入值相加，并将结果赋给输出值。函数类型如下：

FLOAT ADD_FLOAT(FLOAT a, FLOAT b)

INT ADD_INT(INT a, INT b)
LONG ADD_LONG(LONG a, LONG b)
UINT ADD_UINT(UINT a, UINT b)
ULONG ADD_ULONG(ULONG a, ULONG b)
SFLOAT ADD_SFLAOAT(SFLOAT a,SFLOAT b)

- 平均函数

这些函数的功能是求输入值的平均值，并将结果赋给输出值。函数类型如下：

FLOAT AVE_FLOAT(FLOAT a, FLOAT b)
INT AVE_INT(INT a, INT b)
LONG AVE_LONG(LONG a, LONG b)
UINT AVE_UINT(UINT a, UINT b)
ULONG AVE_ULONG(ULONG a, ULONG b)
SFLOAT AVE_SFLOAT(SFLOAT a,SFLOAT b)

对于 AVE_FLOAT 函数，两个输入之和不能超出浮点的量程。

- 除法函数

这组函数的功能是将输入值相除，并将结果赋给输出值。函数类型如下：

FLOAT DIV_FLOAT(FLOAT a, FLOAT b)
INT DIV_INT(INT a, INT b)
LONG DIV_LONG(LONG a, LONG b)
UINT DIV_UINT(UINT a, UINT b)
ULONG DIV_ULONG(ULONG a, ULONG b)
SFLOAT DIV_SFLOAT(SFLOAT a,SFLOAT b)

- 求模函数

该组函数的功能是将输入值相除，并将余数赋给输出值。函数类型如下：

INT MOD_INT(INT a, INT b);
LONG MOD_LONG(LONG a, LONG b);
UINT MOD_UINT(UINT a, UINT b);
ULONG MOD_ULONG(ULONG a, ULONG b);

- 赋值函数

该组函数的功能是将输入赋给输出值。函数类型如下所示：

BOOL MOVE_BOOL(BOOL a);
BYTE MOVE_BYTE(BYTE A);
DWORD MOVE_DWORD(DWORD a);
FLOAT MOVE_FLOAT(FLOAT a);
INT MOVE_INT(INT a);
LONG MOVE_LONG(LONG a);
UINT MOVE_UINT(UINT a);
ULONG MOVE_ULONG(ULONG a);
WORD MOVE_WORD(WORD a);
SFLOAT MOVE_SFLOAT(SFLOAT a);

- 乘法函数

该组函数的功能是将输入值相乘，并将结果赋给输出值。函数的类型如下所示：

FLOAT MUL_FLOAT(FLOAT a, FLOAT b);
INT MUL_INT(INT a, INT b);
LONG MUL_LONG(LONG a, LONG b);

```
UINT        MUL_UINT(UINT a, UINT b);
ULONG       MUL_ULONG(ULONG a, ULONG b);
SFLOAT      MUL_SFLOAT(SFLOAT a,SFLOAT b);
```

- 减法函数

该组函数的功能是将输入值相减，并将结果赋给输出值。函数类型如下所示：

```
FLOAT       SUB_FLOAT(FLOAT a, FLOAT b);
INT         SUB_INT(INT a, INT b);
LONG        SUB_LONG(LONG a, LONG b);
UINT        SUB_UINT(UINT a, UINT b);
ULONG       SUB_ULONG(ULONG a, ULONG b);
SFLOAT      SUB_SFLOAT(SFLOAT a,SFLOAT b)';
```

2) 比较函数

- 等于比较

该组函数功能是检查第一个输入值是否等于第二个输入值，若是，则输出值为 ON，否则为 OFF。函数类型如下所示：

```
BOOL        EQ_BOOL(BOOL a, BOOL b);
BOOL        EQ_BYTE(BYTE A,BYTE B);
BOOL        EQ_DWORD(DWORD a, DWORD b);
BOOL        EQ_FLOAT(FLOAT a, FLOAT b);
BOOL        EQ_INT(INT a, INT b);
BOOL        EQ_LONG(LONG a, LONG b);
BOOL        EQ_UINT(UINT a, UINT b);
BOOL        EQ_ULONG(ULONG a, ULONG b);
BOOL        EQ_WORD(WORD a, WORD b);
BOOL        EQ_SFLOAT(SFLOAT a,SFLOAT b);
```

- 大于等于比较

该组函数功能是检查第一个输入值是否大于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。函数类型如下所示：

```
BOOL        GE_FLOAT(FLOAT a, FLOAT b);
BOOL        GE_INT(INT a, INT b);
BOOL        GE_LONG(LONG a, LONG b);
BOOL        GE_UINT(UINT a, UINT b);
BOOL        GE_ULONG(ULONG a, ULONG b);
BOOL        GE_SFLOAT(SFLOAT a,SFLOAT b);
```

- 大于比较

该组函数功能是检查第一个输入值是否大于第二个输入值，若是，则输出值为 ON，否则为 OFF。函数类型如下所示：

```
BOOL        GT_FLOAT(FLOAT a, FLOAT b);
BOOL        GT_INT(INT a, INT b);
BOOL        GT_LONG(LONG a, LONG b);
BOOL        GT_UINT(UINT a, UINT b);
BOOL        GT_ULONG(ULONG a, ULONG b);
BOOL        GT_SFLOAT(SFLOAT a,SFLOAT b);
```

- 小于等于比较

该组函数功能是检查第一个输入值是否小于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。函数类型如下所示：

```
BOOL        LE_FLOAT(FLOAT a, FLOAT b);
BOOL        LE_INT(INT a, INT b);
BOOL        LE_LONG(LONG a, LONG b);
BOOL        LE_UINT(UINT a, UINT b);
BOOL        LE_ULONG(ULONG a, ULONG b);
BOOL       LE_SFLOAT(SFLOAT a,SFLOAT b);
```

- 小于比较

该组函数功能是检查第一个输入值是否小于第二个输入值，若是，则输出值为 ON，否则为 OFF。函数类型如下所示：

```
BOOL        LT_FLOAT(FLOAT a, FLOAT b);
BOOL        LT_INT(INT a, INT b);
BOOL        LT_LONG(LONG a, LONG b);
BOOL        LT_UINT(UINT a, UINT b);
BOOL        LT_ULONG(ULONG a, ULONG b);
BOOL       LT_SFLOAT(SFLOAT a,SFLOAT b);
```

- 不等比较

该函数的功能是对两个输入值进行比较，若输入值不等，则输出值为 ON，否则为 OFF。N 和 ENO 能作为附加参数加以设置。 函数类型如下所示：

```
BOOL        NE_BOOL(BOOL a, BOOL b);
BOOL       NE_BYTE(BYTE A,BYTE B);
BOOL        NE_DWORD(DWORD a, DWORD b);
BOOL        NE_FLOAT(FLOAT a, FLOAT b);
BOOL        NE_INT(INT a, INT b);
BOOL        NE_LONG(LONG a, LONG b);
BOOL        NE_UINT(UINT a, UINT b);
BOOL        NE_ULONG(ULONG a, ULONG b);
BOOL        NE_WORD(WORD a, WORD b);
BOOL       NE_SFLOAT(SFLOAT a,SFLOAT b);
```

3) 转换函数

- LONG DWORD_TO_LONG(DWORD a);
  该函数功能是将 DWORD 型的输入值转化为 LONG 型数据类型。
- INT FLOAT_TO_INT(FLOAT a);
  该函数功能是将 FLOAT 型的输入值转化为 INT 型数据类型。
- FLOAT INT_TO_FLOAT(INT a);
  该函数功能是将 INT 型的输入值转化为 FLOAT 型数据类型。
- LONG INT_TO_LONG(INT a);
  该函数功能是将 INT 型的输入值转化为 LONG 型数据类型。
- UINT INT_TO_UINT(INT a);
  该函数功能是将 INT 型的输入值转化为 UINT 型数据类型。
- WORD INT_TO_WORD(INT a);
  该函数功能是将 INT 型的输入值转化为 WORD 型数据类型。
- DWORD LONG_TO_DWORD(LONG a);

该函数功能是将 LONG 型的输入值转化为 DWORD 型数据类型。

- FLOAT LONG_TO_FLOAT(LONG a);
  该函数功能是将 LONG 型的输入值转化为 FLOAT 型数据类型。
- INT LONG_TO_INT(LONG a);
  该函数功能是将 LONG 型的输入值转化为 INT 型数据类型。
- ULONG LONG_TO_ULONG(LONG a);
  该函数功能是将 LONG 型的输入值转化为 ULONG 型数据类型。
- INT UINT_TO_INT(UINT a);
  该函数功能是将 UINT 型的输入值转化为 INT 型数据类型。
- ULONG UINT_TO_ULONG(UINT a);
  该函数功能是将 UINT 型的输入值转化为 ULONG 型数据类型。
- WORD UINT_TO_WORD(UINT a);
  该函数功能是将 UINT 型的输入值转化为 WORD 型数据类型。
- DWORD ULONG_TO_DWORD(ULONG a);
  该函数功能是将 ULONG 型的输入值转化为 DWORD 型数据类型。
- LONG ULONG_TO_LONG(ULONG a);
  该函数功能是将 ULONG 型的输入值转化为 LONG 型数据类型。
- UINT ULONG_TO_UINT(ULONG a);
  该函数功能是将 ULONG 型的输入值转化为 UINT 型数据类型。
- BYTE WORD_TO_BYTE(WORD A);
  该函数功能是将 WORD 型的输入值转化为 BYTE 型数据类型。
- WORD BYTE_TO_WORD(BYTE A);
  该函数功能是将 BYTE 型的输入值转化为 WORD 型数据类型。
- INT WORD_TO_INT(WORD a);
  该函数功能是将 WORD 型的输入值转化为 INT 型数据类型。
- UINT WORD_TO_UINT(WORD a);
  函数功能是将 WORD 型的输入值转化为 UINT 型数据类型。
- FLOAT SFLOAT_TO_FLOAT(SFLOAT A);
  函数功能是将 SFLOAT 型的输入值转化为 FLOAT 型数据类型。
- SFLOAT FLOAT_TO_SFLOAT(FLOAT A);
  函数功能是将 FLOAT 型的输入值转化为 SFLOAT 型数据类型。
- INT SFLOAT_TO_INT(SFLOAT A);
  函数功能是将 SFLOAT 型的输入值转化为 INT 型数据类型。
- SFLOAT INT_TO_SFLOAT(INT A);
  函数功能是将 INT 型的输入值转化为 SFLOAT 型数据类型。
- INT DEC_TO_BCD(BYTE dec);
  函数功能：同十进制转换为 BCD 码模块，具体见 FBD 说明书。

4) 逻辑运算

- **逻辑与函数**

该组函数的功能是将输入值进行该逻辑与操作，并将结果赋给输出值。函数类型如下所示：

```
BOOL    AND_BOOL(BOOL a, BOOL a);
BYTE    AND_BYTE(BYTE A,BYTE B);
```

DWORD AND_DWORD(DWORD a, DWORD b);
WORD AND_WORD(WORD a, WORD b);

- **逻辑取反函数**

该组函数的功能是将输入值进行逻辑取反操作，并将结果赋给输出值。函数类型如下所示：

BOOL NOT_BOOL(BOOL a);
BYTE NOT_BYTE(BYTE A);
DWORD NOT_DWORD(DWORD a);
WORD NOT_WORD(WORD a);

- **逻辑或函数**

该函数的功能是将输入值进行逻辑或操作，并将结果赋给输出值。函数类型如下所示：

BOOL OR_BOOL(BOOL a, BOOL b);
BYTE OR_BYTE(BYTE A,BYTE B);
DWORD OR_DWORD(DWORD a, DWORD b);
WORD OR_WORD(WORD a, WORD b);

- **循环左移函数**

该组函数功能是将输入值 IN 进行循环左移，并将结果赋给输出值。 函数类型如下所示：

DWORD ROL_DWORD(DWORD a, UINT b);
WORD ROL_WORD(WORD a, UINT b);

- **循环右移函数**

该组函数功能是将输入值 IN 进行循环右移，并将结果赋给输出值。函数类型如下所示：

DWORD ROR_DWORD(DWORD a, UINT b);
WORD ROR_WORD(WORD a, UINT b);

- **逻辑左移函数**

该组函数功能是将输入值 IN 进行左移（从右边填零），并将结果赋给输出值 OUT。函数类型如下所示：

DWORD SHL_DWORD(DWORD a, UINT b);
WORD SHL_WORD(WORD a, UINT b);

- **逻辑右移函数**

该组函数功能是将输入值 IN 进行右移（从左边填零），并将结果赋给输出值 OUT。函数类型如下所示：

DWORD SHR_DWORD(DWORD a, UINT b);
WORD SHR_WORD(WORD a, UINT b);

- **逻辑异或函数**

该组函数的功能是将输入值进行逻辑异或操作，并将结果赋给输出值。函数类型如下所示：

BOOL XOR_BOOL(BOOL a, BOOL b);
BYTE XOR_BYTE(BYTE A,BYTE B);
DWORD XOR_DWORD(DWORD a, DWORD b);
WORD XOR_WORD(WORD a, WORD b);

5) 数学函数

- FLOAT ABS_FLOAT(FLOAT a);
  INT ABS_INT(INT a);
  LONG ABS_LONG(LONG a);
  这三个函数的功能是计算输入值的绝对值并将结果赋给输出值。

- FLOAT ACOS(FLOAT a);
  该函数功能是计算输入值的反余弦值，并将结果以弧度的形式赋给输出值。

- FLOAT ASIN(FLOAT a);
  该函数功能是计算输入值的反正弦值，并将结果以弧度的形式赋给输出值。

- FLOAT ATAN(FLOAT a);
  该函数的功能是计算输入值的反正切值，并将结果以弧度的形式赋给输出值。

- FLOAT ATAN2(FLOAT a, FLOAT b);
  该函数的功能是计算坐标 (x,y)对应的反正切值，并将结果以弧度的形式赋给输出值。

- FLOAT COS(FLOAT a);
  该函数功能是计算输入值的余弦值，并将结果赋给输出值。

- FLOAT COSH(FLOAT a);
  该函数功能是计算输入值的工程余弦值，并将结果赋给输出值。输入值必须是弧度形式。

- FLOAT EXP(FLOAT a);
  该函数功能是计算以 e 为底，输入值 IN 为指数的幂级数，并将结果赋给输出值。

- FLOAT LN(FLOAT a);
  该函数功能是计算输入值自然对数，并将结果赋给输出值。

- FLOAT LOG(FLOAT a);
  该函数功能是计算以 10 为底的对数，并将结果赋给输出值。

- FLOAT POW(FLOAT a, FLOAT b);
  该函数功能是计算 y 为指数，x 为底的幂级数，并将结果赋给输出值 OUT。

- FLOAT SIN(FLOAT a);
  该函数功能是计算输入值 IN 的正弦值，并将结果赋给输出值 OUT。

- FLOAT SINH(FLOAT a);
  该函数功能是计算输入值 IN 的工程正弦值,并将结果赋给输出值。

- FLOAT SQRT_FLOAT(FLOAT a);
  SFLOAT SQRT_SFLOAT(SFLOAT a);
  这两个函数功能是计算输入值 IN 的平方根，并将结果赋给输出值 OUT。 对于

SQRT_SFLOLAT 函数，其输入值在 0 到 1 之间。

- FLOAT TAN(FLOAT a);
  该函数功能是计算输入值 IN 的正切值，并将结果赋给输出值 OUT。

- FLOAT TANH(FLOAT a);
  函数功能是计算输入值 IN 的工程正切值，并将结果赋给输出值 OUT。

6) 选择函数

- 限幅函数

该组函数的功能是限幅，即当输入大于上限值时输出上限值，当输入小于下限值时输出下限值，否则输出输入值。函数类型如下所示：

```
FLOAT     LIM_FLOAT(FLOAT max, FLOAT a, FLOAT min);
INT       LIM_INT(INT max, INT a, INT min);
LONG      LIM_LONG(LONG max, LONG a, LONG min);
UINT      LIM_UINT(UINT max, UINT a, UINT min);
ULONG     LIM_ULONG(ULONG max, ULONG a, ULONG min);
SFLOAT    LIM_SFLOAT(SFLOAT max,SFLOAT a,SFLOAT min);
```

- 最大值函数

该组函数的功能是将输入值中的最大值赋给输出值。函数类型如下所示：

```
FLOAT     MAX_FLOAT(FLOAT a, FLOAT b);
INT       MAX_INT(INT a, INT b);
LONG      MAX_LONG(LONG a, LONG b);
UINT      MAX_UINT(UINT a, UINT b);
ULONG     MAX_ULONG(ULONG a, ULONG b);
SFLOAT    MAX_SFLOAT(SFLOAT a,SFLOAT b);
```

- 最小值函数

该组函数的功能是将输入值中的最小值赋给输出值。函数类型如下所示：

```
INT       MIN_INT(INT a, INT b);
LONG      MIN_LONG(LONG a, LONG b);
UINT      MIN_UINT(UINT a, UINT b);
ULONG     MIN_ULONG(ULONG a, ULONG b);
SFLOAT    MIN_SFLOAT(SFLOAT a,SFLOAT b);
```

- 选择函数

该组函数的功能是当 SW=OFF 时，将输入值 IN1 赋给输出值；当 SW=ON 时，将输入值 IN2 赋给输出值。函数类型如下所示：

```
BOOL      SEL_BOOL(BOOL sw, BOOL a, BOOL b);
DWORD     SEL_DWORD(BOOL sw, DWORD a, DWORD b);
FLOAT     SEL_FLOAT(BOOL sw, FLOAT a, FLOAT b);
INT       SEL_INT(BOOL sw, INT a, INT b);
LONG      SEL_LONG(BOOL sw, LONG a, LONG b);
UINT      SEL_UINT(BOOL sw, UINT a, UINT b);
ULONG     SEL_ULONG(BOOL sw, ULONG a, ULONG b);
WORD      SEL_WORD(BOOL sw, WORD a, WORD b);
SFLOAT    SEL_SFLOAT(BOOL sw,SFLOAT a,SFLOAT b);
```

- BOOL SEL_1IN3(BOOL pv1,BOOL pv2,BOOL pv3);
  函数功能：同三选一开关信号选择模块，具体见 FBD 说明书。

- BOOL SEL_1IN5(BOOL pv1,BOOL pv2,BOOL pv3,BOOL pv4,BOOL pv5);
  函数功能：同五选一开关信号选择模块，具体见 FBD 说明书。

7) 通讯辅助函数

- BOOL GETBIT(DWORD num, UINT serial);

函数功能：从输入的 DWORD 型变量 num 中取出指定位的标号，如果该标号为 1，则输出为 ON，如果为 0，则输出为 OFF。

serial 取值范围是 0～31，如果 serial 大于 31，那么输出 OFF。

- FLOAT GETFLOAT(DWORD num);
  函数功能：将输入 DWORD 型变量用 FLOAT 型来解释输出。

- INT GETINT(DWORD num, UINT serial);

函数功能：从 32 位的 DWORD 型输入变量 num 的指定位置取出 16 位的 INT 型变量，当 SERIAL＝0 时，取低 16 位；当 SERIAL≠0 时，取高 16 位。

- DWORD GETMSG(UINT nStation, UINT serial);

函数功能：该模块根据输入的控制站序号 nStation 和消息序号 serial，直接从接收缓冲区中读出指定控制站指定位置的信息。

- UINT GETUINT(DWORD num, UINT serial);

函数功能：从 32 位的 DWORD 型输入变量 num 的指定位置取出 16 位的 UINT 型变量，当 serial＝0 时，取低 16 位；当 serial≠0 时，取高 16 位。

- WORD GETWORD(DWORD num, UINT serial);

函数功能：从 32 位的 DWORD 型输入变量 num 的指定位置取出 16 位的 WORD 型变量，当 serial＝0 时，取低 16 位；当 serial≠0 时，取高 16 位。

- SFLOAT GETSFLOAT(DWORD num,UINT serial);

函数功能 :该模块的功能是从输入的 32 位 DWORD 型值的指定位置取 16 位的 SFLOAT 型值， 当 serial=0，取低 16 位；当 serial≠0，取高 16 位。

- void SENDMSG(UINT size);

函数功能：该模块根据输入的消息个数，使能控制站共享信息广播，广播所指定的消息个数到 SCnetII 网络，其他控制站通过 SCnetII 网络收到该控制发送的信息。具体功能见 SENGMSG 功能块。

- DWORD SETBIT(DWORD num, BOOL val, UINT serial);
  函数功能：设置输入 num 的 serial 位的值，当 val 为 OFF 时，该位为 0，反之则为 1。
  serial 取值范围是 0～31，当 serial>31 时，返回 0。

- DWORD SETFLOAT(FLOAT val);
  函数功能：将输入 val 安 DWORD 型来解释输出。

- DWORD SETINT(DWORD num, INT val, UINT serial);
  函数功能：当 serial＝0，将输入 num 的低 16 位设置为 val，当 serial≠0，将输入 num 的高 16 位设置为 val。

- DWORD SETUINT(DWORD num, UINT val, UINT serial);
  函数功能：当 serial＝0，将输入 num 的低 16 位设置为 val，当 serial≠0，将输入 num 的高 16 位设置为 val。

- DWORD SETWORD(DWORD num, WORD val, UINT serial);
  函数功能：当 serial＝0，将输入 num 的低 16 位设置为 val，当 serial＝1，将输入 num 的高 16 位设置为 val。

- DWORD SETSFLOAT(DWORD num,SFLOAT val,UINT serial);
  函数功能：当 serial＝0，将输入 num 的低 16 位设置为 val，当 serial＝1，将输入 num 的高 16 位设置为 val。

8) 系统时间函数

- INT CENTURY();
  函数功能：得到当前年份的前两个数字，与 YEAR()函数得到的值组成当前年份。

- INT YEAR();
  函数功能：得到当前年份后两个数字，与 CENTURY()函数得到的值组成当前年份。

- INT MONTH();
  函数功能：得到当前月份。

- INT DAY();
  函数功能：得到当前日期。

- INT HOUR();
  函数功能：得到当前时钟值。

- INT MINUTE();
  函数功能：得到当前的分钟值。

- INT SECOND();
  函数功能：得到当前的秒数。

9) 特殊函数

- BOOL LIMITOR(INT IN1,INT IN2,INT IN3,INT IN4,INT IN5,INT IN6,INT IN7,INT IN8,INT S9,INT S10);
  函数功能：用来监视 8 个以下的数字量输入，并根据 S9 和 S10 设置的状态产生一个 bool

量输出。具体见 FBD 说明书。

- SFLOAT NEWSQRT(SFLOAT IN,SFLOAT KP,SFLOAT DIS);
  函数功能：用来计算半浮点输入 IN 的平方根，输出= KP×SQRT（IN）＋DIS 。

11) 其他函数

- BOOL ISSTANDBY();
  函数功能：取得主控卡冗余状态，返回为 ON 表示处于备用状态，OFF 则为工作状态。

11) 输入处理函数

- WORD GETPATFLAG(INT N);
  函数功能：得到第 N 通道 PAT 卡工作标志，具体见 FBD 说明书。

- WORD GETPATSTATE(INT N);
  函数功能：取得第 N 通道 PAT 卡状态，具体见 FBD 说明书。

- SFLOAT GETPATPV(INT N);
  函数功能：取第 N 通道的 PAT 卡 PV 值。

- FLOAT SATENTHA(FLOAT P);
  函数功能：计算压力范围是 0.1MPa-16.0MPa 的饱和蒸汽的焓值。输入的压力单位是 KPa。具体见 FBD 说明书。

- SFLOAT SATSTEAM(float press,float press0,SFLOAT Flow0);
  函数功能：饱和蒸汽的温压补偿，具体可见 FBD 说明书。

- SFLOAT SATSTEAM_DP(float PRESS,float DENSITY0,SFLOAT FLOW0);
  函数功能：饱和蒸汽的温压补偿，具体可见 FBD 说明书。

- SFLOAT SATSTEAM_EX(BOOL SIGNALSEL,float PRESS,float DENSITY0,SFLOAT SIGNAL);
  函数功能：饱和蒸汽的温压补偿模块，具体见 FBD 说明书。

- SFLOAT COMPENSATE(SFLOAT SteamFlow0,FLOAT SteamTemperture,FLOAT SteamPress,FLOAT DesignV);
  函数功能：过热蒸汽的温压补偿模块，具体可参见 FBD 说明书。

- SFLOAT FXY(SFLOAT x,INT n);
  函数功能：折线表处理函数，x 是输入值，n 是指采用第几个折线表进行插值。

- SFLOAT GET_FXY_X(INT num,INT n);
  函数功能：得到第 n 个折线表第 num 个 x 的值。

- SFLOAT GET_FXY_Y(INT num,INT n);
  函数功能：得到第 n 个折线表第 num 个 y 的值。

- void SET_FXY_X(SFLOAT x,INT num,INT n);

函数功能：设置第 n 个折线表第 num 个 x 的值。

- void SET_FXY_Y(SFLOAT x,INT num,INT n);
  函数功能：设置第 n 个折线表第 num 个 y 的值。

- SFLOAT LINECPS(SFLOAT IN,SFLOAT KP,SFLOAT DIS);
  函数功能：输出= IN×KP＋DIS。

- void SETPATCON(WORD CON,INT N);
  函数功能：设置第 N 个 PAT 通道的特殊标志字节。

7.导入和导出

图形编程中的自定义数据类型、全局变量、段落都可以导入和导出。通过导入和导出，用户可以方便的重用先前的信息，在工程间传递组态。

图形编程的智能导入导出给予用户极大的方便。

导入导出时先使用 **工程** 菜单中的 **工程管理** 命令弹出工程管理对话框。

当在 **数据类型管理** 对话框选择一个或多个自定义数据类型导出时，用户要指定导出数据存放的文件名。图形编程检查所有的数据类型，若发现所选中的数据类型是由未选中的自定义数据类型派生而来，图形编程将自动追加这些更底层的自定义数据类型。导入时，选择已生成的导出文件，工程中将添加所包含的数据类型。

当在 **全局变量** 对话框选择一个或多个变量导出时，用户要指定导出数据存放的文件名。导出的变量以 ASCII 文本格式存放，易于用户操作。导入时，选择已生成的导出文件，工程中将添加所包含的变量。

当在 **段落管理** 对话框中选择一个或多个段落导出时，用户要指定导出段落存放的文件名。图形编程先检查所有的段落，如段落中包含未被选择的 DFB，则图形编程自动追加这些 DFB 段落。然后检查所有段落中包含的变量的数据类型，若发现其中的数据类型是由自定义数据类型派生而来，图形编程将自动追加这些自定义数据类型。导入时，选择已生成的导出文件，工程中将添加所包含的数据类型、段落。当导入时，发现段落名冲突，将提示用户是替换或保留或用新名导入。

8. 数据类型编辑器

图形编程内置数据类型编辑器，用户通过数据类型编辑器生成自定义的数据类型。定义新数据类型时，可以使用基本数据类型和已在工程中生成的自定义数据类型。

图形编程内置数据类型编辑器，用户可以用数据类型编辑器生成自己的数据类型，并可以在任何编辑变量类型的地方使用。

用**工程**菜单中的**工程管理**命令打开工程管理对话框。再选择**数据类型管理**进入数据类型编辑器。

图 1-55 数据类型编辑器

按添加则弹出添加复合类型对话框，用户可以选择数组或是结构：

图 1-56 添加复合类型

选择一个数组类型按编辑按钮弹出数组声明对话框。

图 1-57 数组声明

选择一个结构类型按编辑按钮弹出结构声明对话框。

图 1-58 结构声明

通过导入/导出功能，用户可以将数据类型保存到一个文件中，其他功能从此文件中导入就可以使用这些定义的数据类型。

系统内已预定义了部分数据类型，这些数据类型只读，用户无法修改删除。

9. 变量编辑器

用户通过变量编辑器定义变量。

定义全局变量、私有变量、输入变量、输出变量的界面统一。可以定义变量名、变量数据类型和注释。

10. DFB 编辑器

DFB 编辑器用于建立自定义功能块（DFB）。从编程角度看，DFB 可以理解为子程序。

DFB 可以用 FBD 编辑器、LD 编辑器生成，也可以被 FBD、LD、SFC 调用。

图形编程 DFB 用于建立 DFB（Derived Function Blocks 自定义功能块）用 FBD、LD 都可以设计 DFB。

从编程角度上看，一个 DFB 就相当于一个子程序。

一个 DFB 代表由用户定义的输入/输出变量组成的框架。内部包含用户定义的程序逻辑。用户可以多次重用 DFB。

图 1-59 选择模块

在使用方法上看，DFB 与 EFB 没有不同。

在 DFB 内部可以引用一个或多个 EFB 和其他 DFB。DFB 不能自身循环嵌套。

在图形编程中，当新建或打开模块段落时，DFB 编辑器自动启动。

DFB 变量

DFB 内有 4 类变量：

- 私有变量
- 输入变量
- 输出变量
- 热备份变量

输入变量用作向 DFB 传递数值，输出变量用作从 DFB 传出数值。这两类变量被称为形式参数。这些变量在 DFB 被使用时表现为外部的输入输出引脚。

形式参数的名字、数据类型、引脚序号在变量编辑器中定义。引脚序号即为在变量编辑器中的自然顺序。

在图形编程 DFB 中，输入/输出变量最大可各定义 32 个。

在 DFB 被使用时，实际参数替代形式参数参与 DFB 内部的算法逻辑。

热备份变量对用户不可见，当 DFB 引用了包含热备份变量的 FFB 时，DFB 自动继承了 FFB 的热备份变量。

图形编程 DFB 提供了私有变量。在 DFB 中的私有变量与程序段落中的私有变量不同。程序中的每一个私有变量唯一对应控制站的一个内存地址。而 DFB 中的私有变量在 DFB 不被其他程序引用时，没有对应实际的控制站地址。当其他程序引用 DFB 时，每次引用时，DFB 中的私有变量都被分配一个不同的控制站地址。换句话说，DFB 中的私有变量封装了对控制站内存的申请。每一次 DFB 实例的创建，DFB 都按照私有变量的要求申明了相同数目的控制站内存，这些内存只被 DFB 的实例所引用，对外部程序不可见。图形编程 DFB 对私有变量的支持，极大地提高了 DFB 的灵活性和重用性。

用包含私有变量的 DFB 构建新的 DFB 时，新的 DFB 将继承包含的 DFB 的私有变量。

11.系统资源

1) 系统资源

- 定时器

100 毫秒定时器 UINT timerms[256]

秒定时器　UINT timers[256]

分定时器　UINT timerm[256]

应用举例

按 START 按钮 20 秒后做某事

```
IF  NOT  START  THEN
  timers[20] = 0;
END_IF;
IF timers[20] >= 20 THEN
  dosomething();
END_IF;
```

- 自定义控制模块

单回路控制模块数据块

g_bsc[128]

串级控制模块数据块

g_csc[128]

注意：g_bsc 和 g_csc 共用系统内存，不能同时使用同一个下标。

- 控制站间共享数据区

DWORD g_msg[128]

控制站共享此数据区用于站间交换数据

注意：各控制站不能共用同一个数组下标。

应用举例

从 2 号控制站（地址为 2，3） 发送数据到其它控制站,使用 6 号数组下标

```
g_msg[6] = 0;
g_msg[6] = setbit ( g_msg[6] , BOOL1 , 0); (*  用第 0 位发送开关量 1 *)
g_msg[6] = setbit ( g_msg[6] , BOOL2 , 1); (*  用第 1 位发送开关量 2 *)
g_msg[6] = setsfloat ( g_msg[6] , SFLOAT1 , 1);   (*  用高 16 位发送模拟量 1 *)
sendmsg ( 7);   (*  使用的最大下标为 6 *)
```

从其它控制站读取 2 号控制站发送的数据(使用了 6 号数组下标)

```
DWORD dw;
dw = getmsg ( 2 , 6);   (* 读取从 2 号控制站来的数据，使用 6 号数组下标  *)
BOOL1 = getbit ( dw , 0); (*  从第 0 位读开关量 1 *)
BOOL2 = getbit ( dw, 1); (*  从第 1 位读开关量 2 *)
SFLOAT1 = getsfloat ( dw,1); (*  从高 16 位读模拟量 1 *)
```

- 冷热启动及下载组态标志

BOOL g_bHotStartup

热启动标志，热启动时为 ON，需要编程清 0；

BOOL g_bColdStartup

冷启动标志，冷启动时为 ON，需要编程清 0；

g_bDownUsrPrgFlag

下载用户程序标志，下载了用户程序之后为 ON，需要编程清 0；

g_bDownCfgFlag

下载组态标志，下载了组态（包括硬件组态、用户程序组态）后为 ON，需要编程清 0。

- 全局热冗余数组

  LONG g_sfcbaks[32]

  全局热冗余数组。

2) 质量说明

质量码的使用方法

由于 SCX 语言并没有位操作指令，而质量码（Flag)采用位定义，所以可以通过质量码与某个成员的相关数值进行“与”操作的方法来获得该信号是否处于报警状态，如上限报警的相关数值为１。

模入位号数据成员 Flag（位号质量码）定义

质量码的位		说明	位格式	相关数值
D2=0	D0	上限报警	1——报警	1
			0——不报警	
	D1	下限报警	1——报警	2
			0——不报警	
D2=1	D0	上上限报警	1——报警	5
			0——不报警	
	D1	下下限报警	1——报警	6
			0——不报警	
D3				
D4				
D5				
D6				
D7				
D8		信号可疑	1——信号可疑	256
			0——信号不可疑	
D9				512
D10		手动输入	1——手动输入	1024
			0——自动输入	
D11		信号故障	1——信号故障	2048
			0——信号不故障	
D12		速度报警	1——速度报警	4096
			0——速度不报警	
D13				
D14				
D15				

★:若测量点与变送器之间的线路短路或断路，导致信号超量程，这又可称为开路报警。

如输入为III型电流信号，量程为 4～20mA,若输入线路短路，则为 0mA,那通过质量码的 D11 位就可以判断出来。

例如，如果你需要判断某个模入信号是否处于上上限报警，首先将这个模入信号的成员 Flag 与成员 HH 的相关数值 5（详见上表）进行“与”操作，即屏蔽无关位，只留下 FLAG 中的相关位（D2D0)；然后将“与”操作完成后的数值与该相关数值 5 比较判断是否相等。如果相等, 则表明此信号处于上上限报警状态；否则此信号没有处于上上限报警状态。原则上上限报警，上限一定报警；同理下下限报警，下限也一定报警。

1.4 图形编程使用指南

图形编程软件的使用是十分方便的。以下将分别介绍软件的运行环境、编辑界面、菜单项功能、程序类型说明、变量类型说明、工程设计、文件结构以及在线调试等方面的使用。

- 运行环境
- 编辑界面
- 菜单项功能
- 程序类型说明
- 变量类型说明
- 工程设计
- 文件结构
- 在线调试

1.4.1 图形编程的运行环境

图形编程要求运行在 PⅡ300 、64M 内存以上的 PC 机上。要求 Win9x 或 WinNT 或 Windows2000 上。如果需要在线调试，则需与 SUPCON DCS 的主控卡相连。

1.4.2编程界面介绍

图形编程软件的编程界面友好，在一个图形化的界面上提供了大量模块化的功能块，用户通过对模块、触点、线圈等的选取、放置、移动、连接等操作，即可轻松完成编程工作，较一般的高级语言编程要更简单、更方便、更可视化。

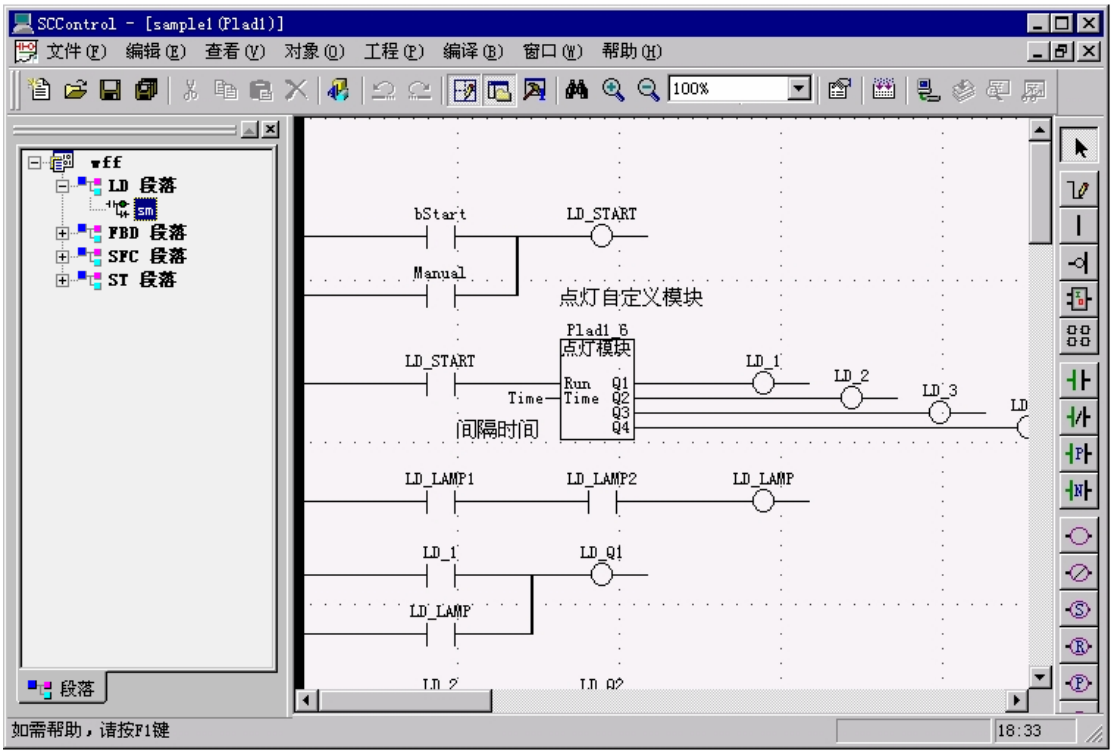


图 1-60 编辑界面

上图是软件的一个编程界面，包括了菜单栏、工具栏、状态栏、工程栏、信息栏以及程序的编辑区。

- 菜单栏

图 1-61 菜单栏

菜单栏集成了软件的大多数功能，有效利用菜单栏可完成一般程序的编辑工作。

- 工具栏

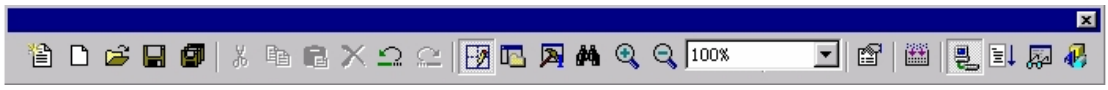


图 1-62 工具栏

工具栏将菜单栏中许多重要的功能图标化，简化了操作。工具栏中的图标和菜单项相连。它们的出现依赖于菜单项的状态：如果菜单项为灰色，对应的工具栏图标也为灰色；如果菜单项被激活，则对应的工具栏图标也处于激活状态。

➢ 梯形图(LD)组态元素

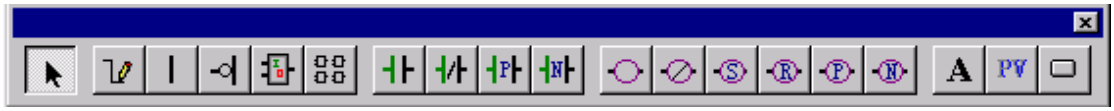


图 1-63 梯形图组态元素

➢ 顺控图(SFC)组态元素

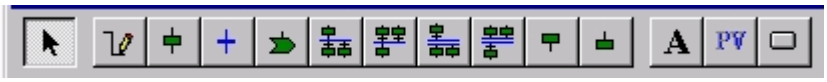


图 1-64 SFC 组态元素

➢ 功能块图(FBD)组态元素

图 1-65 FBD 组态元素

- 工程栏

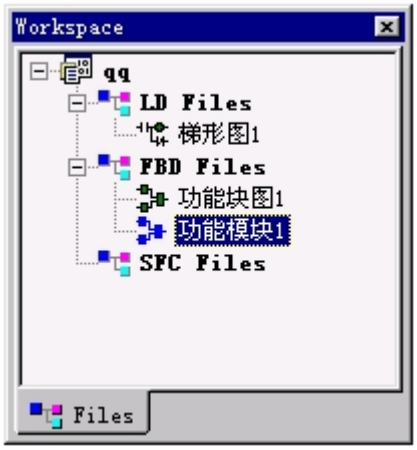


图 1-66 即为一个工程栏。

在 Workspace 中列出了工程中的所有程序段，图中的 LD Files 指的是梯形图类型的程序段；FBD files 指的是功能块图类型的程序段；SFC Files 指的是顺控图类型的程序段。

图 1-66 工程栏

- 编程区

下图所示的程序编辑区是用来进行程序与模块设计、编辑的。窗口的背景是逻辑坐标网格。

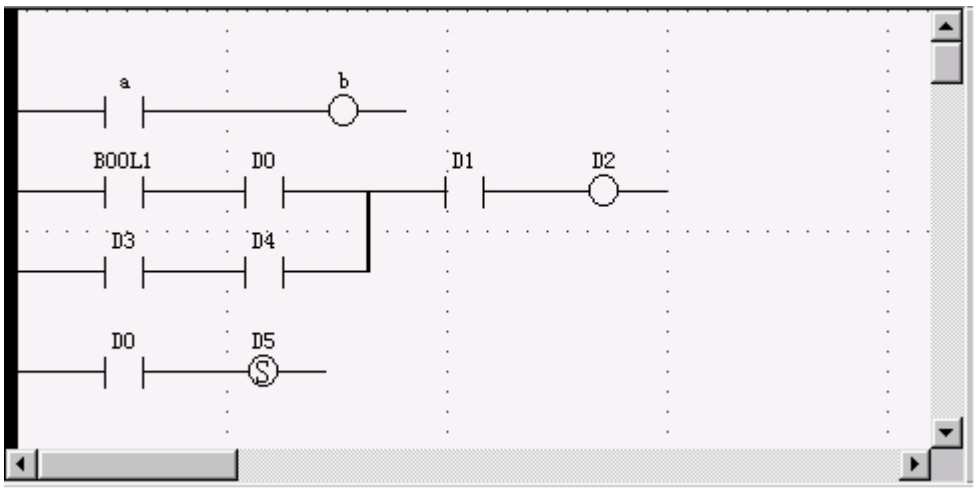


图 1-67 编程区

- 信息栏

信息栏用于编译及工程打开成功与否的信息。

成功打开一个工程或创建一个新工程后，编译信息输出栏输出“ 打开工程成功 ”信息（如下图）；否则输出“ 打开文件失败 ”信息；

图 1-68 信息栏

成功编译好一个程序段或工程后，编译信息输出栏输出“ 编译成功 ”信息（如下图）；

图 1-69 编译成功信息

如果程序段或工程没有完成或存在错误，则编译不成功，编译信息输出栏输出编译不成功的原因（如下图）。用户必须修改程序，反复调试，直到编译信息输出栏输出“编译成功”信息方能进行下一步的操作。

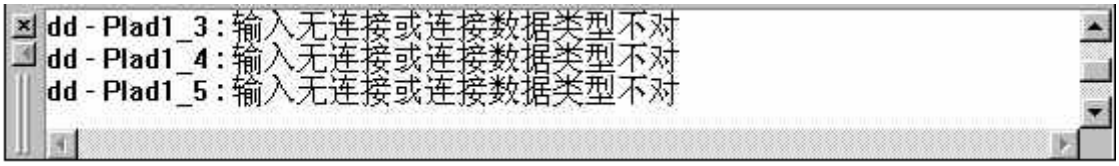


图 1-70 编译错误信息

此外，信息栏还显示编译过程中的错误信息，鼠标双击错误信息则自动跳转到相应的出错的作图元素上，方便改错。

- 状态栏

图 1-71 状态栏

状态栏位于界面的最下方，显示编辑状态，如提示用户在遇到疑难时，可按 F1 键，获取帮助信息等。

1.4.3 菜单功能项介绍

菜单项功能简介

菜单项集成了几乎所有图形编程软件中提供的功能，运用好菜单项，就可以完成整个工程的编辑。图形编程软件的菜单项分为“ 文件 ”、“ 编辑 ”等数个类别，以下将分别加以介绍：

文件	编辑
查看	对象
工程	编译
窗口	帮助

1. 文件菜单功能介绍

在文件菜单条上，设置了很多功能。利用、或者说有效地运用这些功能，将大大有助于图形编程软件的编辑。

新建工程

用于建立新的图形编程软件的工程文件。相关内容可参见 FBD 语言编程中的“ 创建一个新工程 ”，或 LD 语言编程中的“创建一个新工程”。在图形编程软件中，用户可在同一编辑环境中同时新建多个工程。

打开工程

用于打开以往所建立、保存的图形编程工程文件。当对已存在的工程进行编辑操作时，选取文件/**打开工程**命令或者直接用鼠标左键单击工具栏中的打开按钮，屏幕上就会出现**打开工程**对话框。在图形编程软件中，用户可在同一编辑环境中同时打开多个工程。

关闭工程

用于关闭一个正在编辑，或已编辑好的图形编程工程文件。

保存工程

当完成整个工程（包括程序段等）的编辑后，可单击工具栏上的保存图标或选取**文件**菜单中的**保存工程**菜单项，如果所要保存的文件是一个尚未赋予文件名的新文件，屏幕上就会出现保存为对话框，让用户键入文件名。

如果是对一个已存在的旧工程文件进行编辑后保存该文件，则不会出现对话框，而是自动直接以原工程文件名保存，编辑后的工程内容将覆盖原有工程文件内容。

工程另存为

用于将原工程文件保存为其他文件类型的文件，或保存为同一文件类型、其他文件名的文件。

新建程序段

用于建立一个新的程序段。相关内容可参见 LD 语言编程中的“创建一个新程序段”，或 FBD 语言编程中的“创建一个新程序段”的相关内容。

打开程序段

用于打开一个已经建立并存在的程序段。

保存程序段

用于将一个编辑好或正在编辑的程序段进行存盘操作。

打印

在保存好工程或程序段文件后，要打印所建立的文件内容时，可单击工具栏中的“ 打印 ”图标按钮或选择文件/打印命令，执行打印时，会出现如图“ 打印 ”对话框：

图 1-72 打印

- 打印机

画面所显示的打印机名称为当初安装 Windows 95 时所指定的机型，可在下拉式列表框中改变或添加。

- 打印范围

用于选择设定要打印的部分。可选择：

全部： 打印整份文件。

页数： 可在文字框内指定要打印的页次。

选定范围： 打印文件中用户所选定的部分。

- 打印到文件

此复选框是将要打印的文件内容存入一个文件中，而不是由打印机输出。当要把该文件拿到另外一台只有 DOS 系统的计算机上打印时，用户可在 DOS 提示符下键入命令：

TYPE 文件名>PRN

将其直接打印出来，不需进入 Windows 环境。

该项功能与硬件有关，也就是说，用于打印的文件内部不仅仅包含了打印的内容，而且还包含有打印机的配置信息，例如打印机型号、纸张类型等等。所以在使用该功能时，一定要保证相应的两台计算机的打印机、打印设置等等都完全一致。

- 份数

可选定要打印的份数。

- 属性

单击“属性”按钮时，会出现如下图所示对话框：

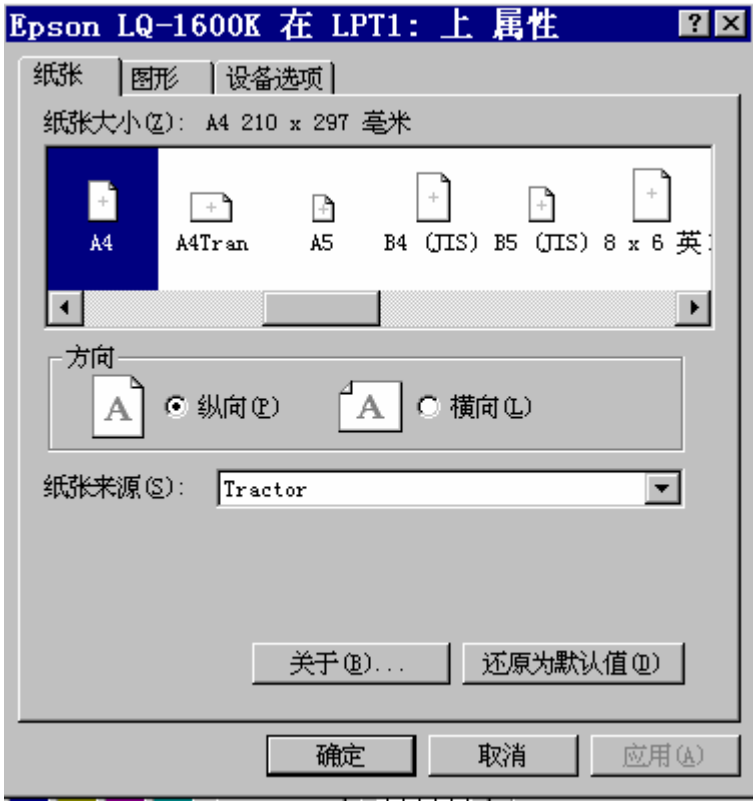


图 1-73 属性

其中包括纸张、图形及设备选项三部分，用户可以根据需要分别设定。

在正式打印文件之前，SUPCON 流程图制作软件提供了预先浏览文件打印状况的功能：**文件/打印预览**，可避免因打印状况不当，导致不必要的浪费。

打印预览

用于在正式打印之前，预先观察实际打印后的效果。详细情况可参考打印功能。

打印设置

用于在打印之前，根据用户自己的实际需要，在“打印设置”对话框中，选取所需的打印设置条件，以获取尽可能好的打印效果。详细情况可参考打印功能。

退出

退出图形编程软件的工程编辑状态，此时将关闭所有正在编辑的程序或模块文件。

2. 编辑菜单功能介绍

撤消

该功能用于取消上一次的操作。图形编程软件提供了多次撤消功能。

恢复

用于执行了撤消操作后，使编辑状态恢复到撤消操作前的状态。

剪切

用于将程序编辑区中用户指定区域的内容复制到**剪贴板**内，同时删除该区域里的内容。

所谓**剪贴板**，是指操作系统内部的一块内存。使用剪贴板，可以在几个模块文件之间传递功能块、线圈、触点、文字或它们之间的任意组合，还可以在几个程序之间进行同样的传递。

该功能的操作过程如下：

1） 选取要剪切的内容

用鼠标选出需要剪切的内容。

2） 剪切

鼠标左键单击**剪切**按钮（或者单击**编辑/剪切**命令），所选定区域中的图像就会复制到剪贴板上去，同时删掉了该区域中的内容。

此时，就可以根据需要，将“剪贴板”中的内容复制到其他的图形编程文件中去。

复制

用于将程序编辑区中选定区域复制到剪贴板内，除了不删除选定区域的内容外，其余功能和用法与“剪切”功能相同，具体使用请参见“剪切功能”的详细说明。

粘贴

用于将剪贴板中的最新内容（即最近一次剪切或复制的内容）复制到指定编辑区内。用鼠标左键单击工具栏的**粘贴**图标，剪贴板中的最新内容将被粘贴在画面的左上角。可以看到粘贴图标有被选中标志，此时，可以按住鼠标左键将其拖到需要的位置（选取工具必须在粘贴内容区域内），此为推荐操作；如果在移动粘贴内容之前进行了其它操作（诸如选取），则粘贴图标的被选中标志消失，若再想移动粘贴内容，必须重新选中，有可能将其下面的内容一起选中，带来不必要的麻烦。

该功能必须在执行**剪切**或**复制**之后使用。在未执行**剪切**或**拷贝**之前，由于剪贴板中无内容，所以图标呈“灰色”，表示该功能无效。

该功能可连续执行多次，在两次操作之间，用户可以使用左右滚动条和上下滚动条调整程序编辑区在实际程序编辑区中的位置。

删除

用于删除一个选定的对象。

查找

用于进行用户所需信息（如模块、文字、变量名等）的检索。用户可自行填写，也可按照不同类型、数据源通过浏览进行检索。可参见“查找与替换”。

替换

用于将当前工程或段落中的内容用用户需要的其他内容替代。参见“查找与替换”。

全选

用于将编辑区中的所有功能块、触点、连接线、线圈、文字等选中。

3. 对象功能介绍

图形编程软件提供了丰富的对象功能，可方便用户从菜单上选取功能块、各种触点、线圈、文本等对象，以下只是对象功能的一部分，其他内容可参看“跳转功能”等的相关介绍。

选择

对象操作时的缺省操作，用于选取编辑区内的模块，连接线，线圈等等对象。

连接线

用于模块与模块、模块与线圈、模块与母线、模块与触点以及其他对象之间连接的水平线条。

垂直连接线

用于连接来自几个输入与一个输出的垂直线条。输入可以来自模块、母线、线圈等对象，输入可以为多个；输出可连向模块、线圈等对象，只能为一个。垂直连接线的功能相当于一个“或”模块。

最近选择的功能块

用于在编辑区内放置一个最近选取过的功能模块。

取反

用于模块的输入管脚对输入信号取反，及用于模块的输出管脚将模块的输出值取反输出。

注释文本

在程序编辑区中起注释作用的文字说明。

功能块选择

用于选取各种不同类型（例如算术运算、选择运算模块等）、不同作用（例如“加”模块、“逻辑右移”模块等）以及不同数据类型（例如 BOOL、WORD 型“或”模块，FLOAT、INT、LONG、SFLOAT、UINT、ULONG 型的“EQ”模块等）的功能模块。用户可在如下图所示的对话框中任意选取：

图 1-74 功能块选择

变量定义

用于声明变量，变量必须在声明之后使用。

4. 工程功能介绍

控制站地址

用于指定工程对应的控制站地址。

数据类型编辑器

用于编辑自定义数据类型。

变量编辑器

用于定义全局变量。

段落管理

用于管理工程中的程序段，可以打开、删除、导入、导出程序段。

任务管理

该功能可用于设置工程中各程序的运行周期。在一个工程中往往有多个程序，它们的重要性或在工程中运行时需调用的周期大小也不同，为在工程的分时处理中，使各个程序得到合理的执行时间，避免各程序执行时的冲突，优化处理过程，用户可运用该功能，根据实际

需要自行安排各个程序的运行周期。

图 1-75 任务管理

上图为设置运行周期的对话框，用户可点击对话框右侧的“修改周期”按钮，自行修改运行周期。

图形编程以系统组态软件中设置的控制周期为 1Ts，即：如果在系统组态软件的组态过程中设置了控制周期为 0.1s，则 1Ts = 0.1s。

用户还可通过操作“移到队首”、“上移”、“下移”、“移到队尾”等操作设置同一运行周期各程序运行的优先级，即排在队列靠前的同一运行周期程序比排在队列靠后的程序优先执行。不同运行周期的程序之间的优先级无法比较。

5. 窗口

层叠

将所有已打开的窗口以层叠的方式排列，原本打开的窗口同时出现在当前编辑区中。如下图所示：

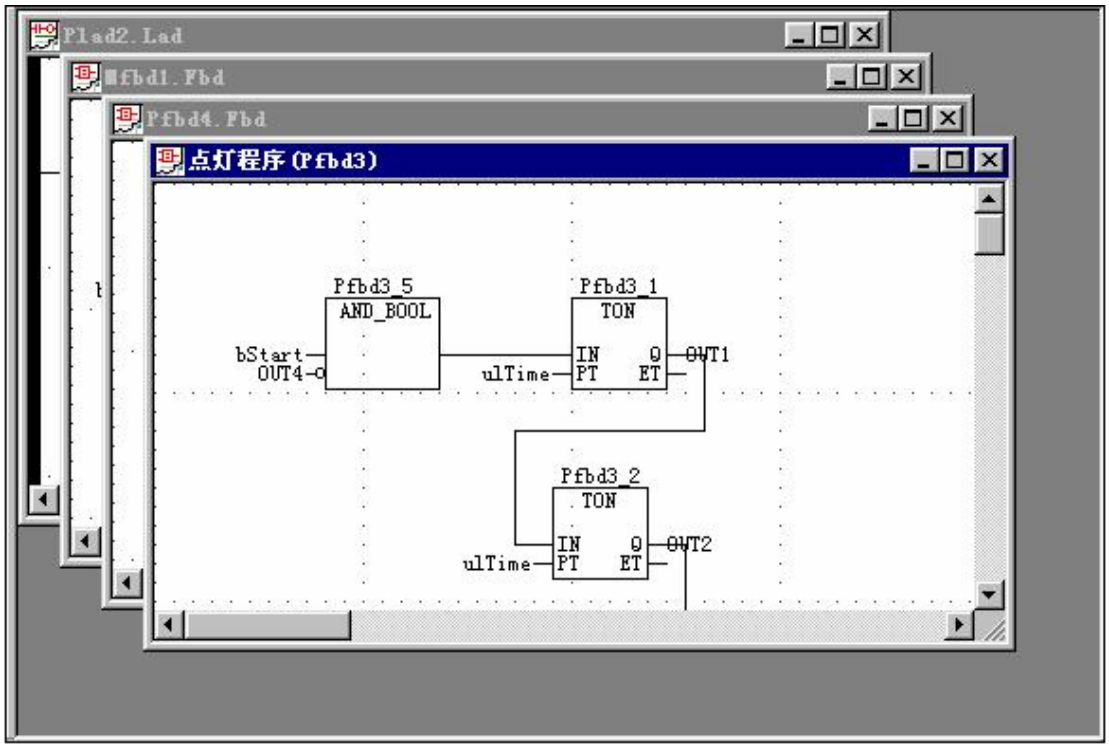


图 1-76 层叠

平铺

将所有已打开的窗口以平铺的方式排列显示于整个编辑区中。如下图所示：

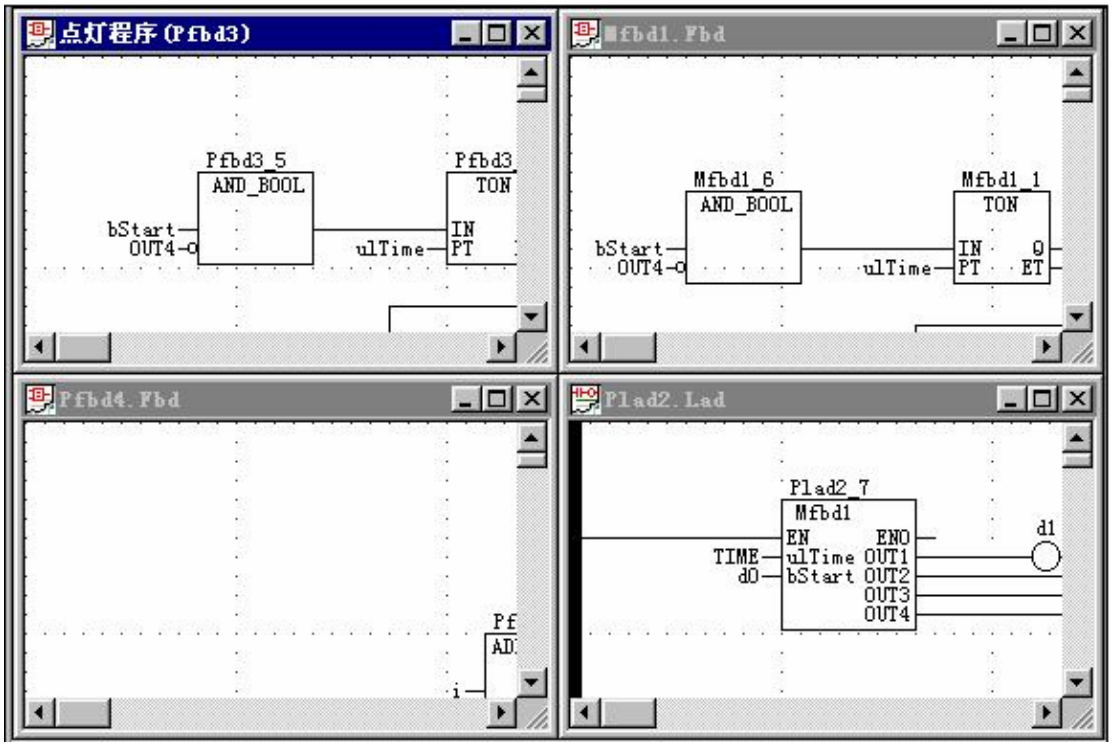


图 1-77 平铺

关闭所有窗口

将所有已经打开的窗口全部关闭。

6. 帮助菜单功能介绍

帮助主题

提供在线帮助。在使用图形编程软件进行 LAD、FBD 语言编程或其它操作的过程中产生疑问或遇到困难，需要帮助时，可以单击该菜单，将得到软件大量详实的相关帮助信息。

关于图形编程

提供软件的版权、版本号等一些相关信息。

7. 编译功能介绍

编译工程

用于将由程序、模块等组成的工程或单独的程序进行编译，以产生机器可识别的目标代码。

优化编译

在编译中重新分配控制站内存。

联机

用于工程与下位机的连接。

调试

用于观察工程执行时的效果。

变量调试窗口

用于在工程运行时，通过手工调节程序变量的取值（见下图），来调试工程或程序的正确性。

图 1-78 变量调试窗口

1.4.4 变量类型说明

图形编程软件支持多种变量类型，编程软件采用的是强类型检查的编译方式，不同类型的相互之间不存在隐式类型转换。系统支持的类型如表 1-1 所示。

表 1-1 图形编程软件支持的系统类型

类型	描述
BOOL	1 字节类型
BYTE	1 字节类型
INT	2 字节类型

UINT	2 字节类型
WORD	2 字节类型
SFLOAT	2 字节类型
LONG	4 字节类型
ULONG	4 字节类型
DWORD	4 字节类型
FLOAT	4 字节类型
structAccum	8 字节类型
structAI	AI 类型
structDI	DI 类型
structPAT	PAT 类型
structCSC	CSC 类型
structBSC	BSC 类型

另外图形编程软件还支持自定义数组和结构类型。其中自定义结构支持任意深度的嵌套。

1.4.5 工程设计

设计的主要步骤：

- **步骤 1** 启动图形编程
- 从 Windows 启动组态软件 Sckey，打开目标文件后，从自定义图形组态启动图形编程。
- 或直接从 Windows 启动图形编程。

- **步骤 2** 新建或打开工程
- 从 **文件** 菜单选择 **新建工程** 。在新建对话框中键入工程名。
- 从 **文件** 菜单选择 **打开工程** 。在打开对话框中选择需编辑的工程文件。
- 从 **文件** 菜单选择 **最近打开文件**。

- **步骤 3** 编程
- 从 **文件** 菜单选择 **新建段落** 。在对话框中键入段落名，选择段落属性。
- 使用工具条或 **对象** 菜单命令生成程序。
- 在 **工程** 菜单中选择 **设置任务** 命令设定程序的执行周期和执行次序。
- 在 **工程** 菜单中选择 **设置相关控制站地址** 设定相关联的控制站地址。

- **步骤 4** 保存编译
- 从 **文件** 菜单选择 **保存工程** 或 **工程另存为**。
- 从 **编译** 菜单选择 **编译工程** 或 **全部重新编译**。

- **步骤 5** 下载测试
- 从 **编译** 菜单选择 **连接** 再选择 **下载。**
- 从 **编译** 菜单选择 **调试**。

- **步骤 6** 优化和断开
- 在 程序修改和下载多次以后，应该进行内存分配优化，选择 **编译** 菜单的 **全部重新编译** 将重新分配内存资源优化编译。
- 断开连接只需再次选择 **连接** 命令。

- **步骤 7** 文档工作
- 在工程调试完毕，应该生成一套完整的文件。

1.4.6 文件结构

编译通过之后观察生成的文件目录结构如下：

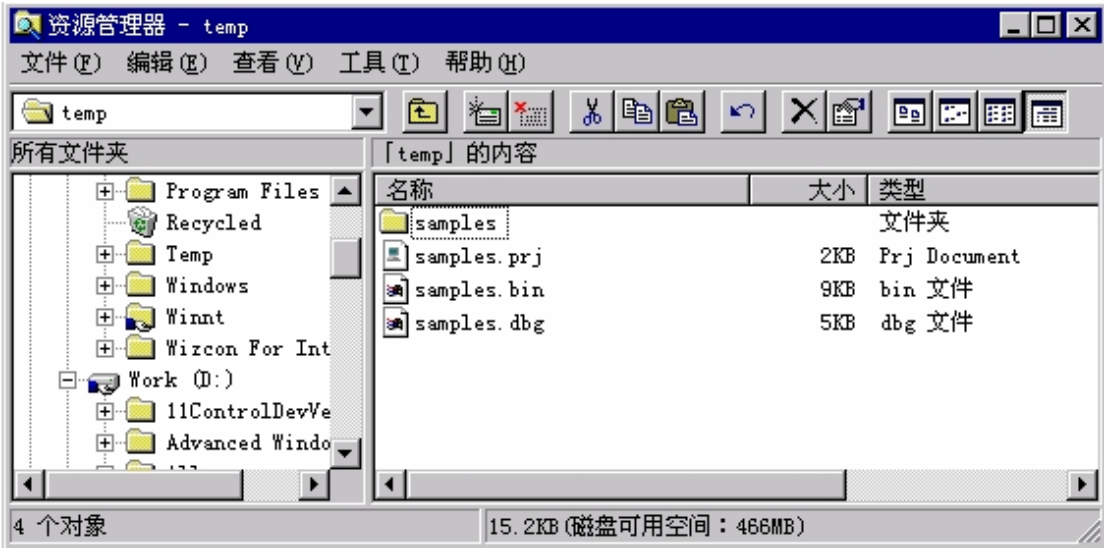


图 1-79 文件目录结构

如上例，samples.prj 为实际的工程文件。

Samples.dbg 包含了工程对控制站的内存映象，每个变量在控制站的地址保存在此文件中。删除此文件将导致工程对控制站内存映象重新建立。

Samples.bin 为生成的工程代码。此代码下载到控制站就执行相应的控制方案。

Samples 子目录包含了工程的段落文件。

1.4.7 在线调试

编程器与控制站连接后，可以下载程序，也可以进入在线调试（即联机调试）。进入联机调试时先校验上位机工程与实际控制站程序，不一致则报警。编程器中的当前程序与控制站实际连接，程序中的开关量和开关链路将根据实际数据显示通断状态。在程序中的调试文本（PV）将显示其实际值。用户可以通过 PV 设置控制站的数据。

- **连接**

只有当连接处于工作时，才能使用下载和调试命令。使用 **编译** 菜单上的 **连接** 命令或单击工具条上 **连接** 按钮就进入连接状态。这时，**下载** 命令和 **调试** 命令从不可用状态变为可用状态。

- **调试**

按 **调试** 按钮进入调试状态。在调试状态下，可以以动画形式观察布尔变量在变化，也

能从调试文本操纵模拟数据。如图：

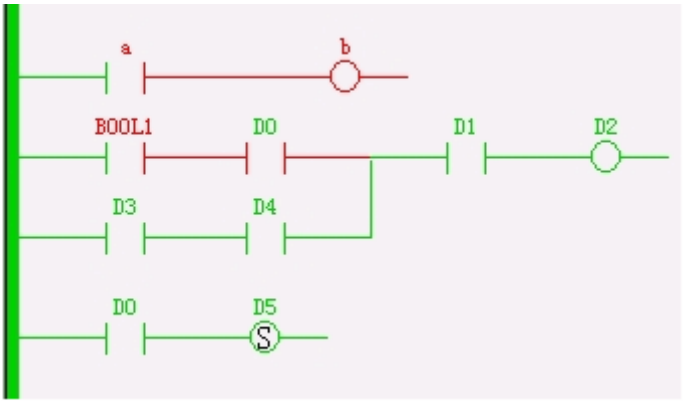


图 1-80 调试状态

变量和链路的布尔数值为 ON 时 动画显示为绿色，OFF 显示为红色。

按 **变量调试窗口** 按钮可进入调试变量对话框，在对话框内用户可以键入需调试变量的值，变量值将实时刷新。

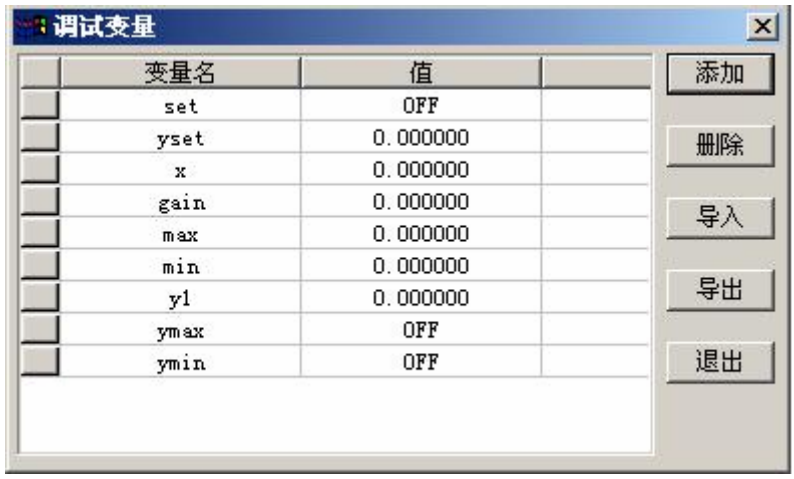


图 1-81 变量调试窗口

- **运行时间**

在调试状态下，打开运行时间窗口，可以观察每个程序在每次执行花费的时间 。

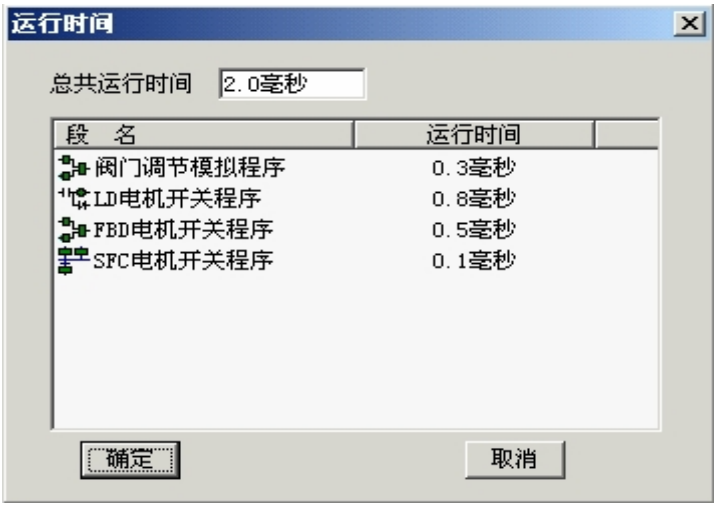


图 1-82 运行时间

1.4.8 密码保护

- 设置密码
- 删除密码
- 使用注意

1. 设置密码

密码可以有字符、数字或者下划线组成；左键选中已打开的段落，点击右键出现如下图所示的密码设置菜单，选择设置密码菜单，即可进行密码设置。

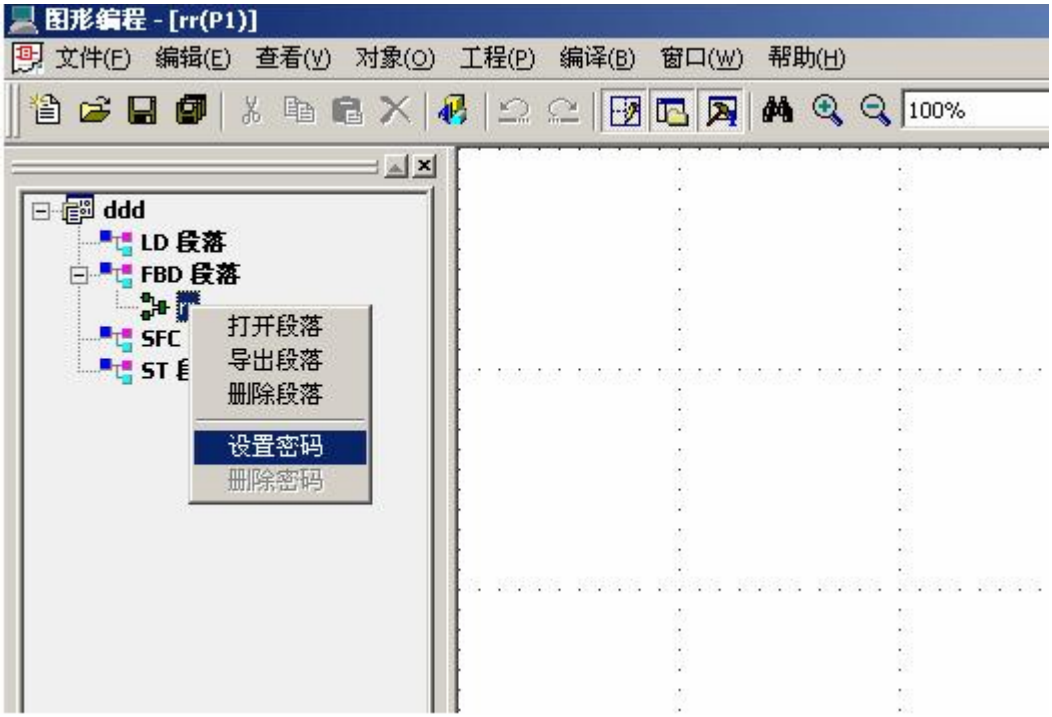


图 1-83 密码设置

段落必须在其编辑界面打开后，才能设置密码或删除密码的。

有密码保护段落的编辑界面一旦关闭，段落就处于密码保护状态，用户不能进行设置密码或删除密码操作。只有当编辑界面打开后，才可以进行设置密码和删除密码的操作。

2. 删除密码

段落在其编辑界面打开后，如果有密码保护就可以进行删除密码操作。删除密码时，将有确认对话框出现，如下图密码确认对话框。确定删除密码后，密码从工程中删除。

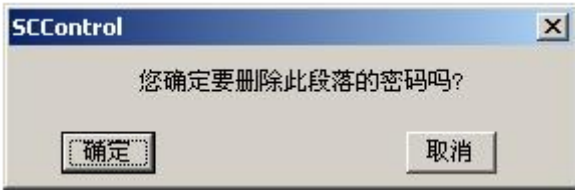


图 1-84 删除密码

3. 使用注意

工程文件的版本

工程文件中的某一段落设置密码后，以前的 SCControl 版本无法打开有密码保护段落的工程文件。

段落导入和导出的

当前导出的段落，如果有密码保护，则导出的段落始终存在密码保护。在导入的时候，其编辑界面默认处于关闭状态，密码跟导出前段落的一样。

段落的更名操作

如果当前有密码保护的段落的编辑界面处于打开状态，则修改段落名后，相应修改后的段落有密码保护，并且处于打开状态 如果当前有密码保护的段落的编辑界面处于关闭状态，则修改段落名后，相应修改后的段落有密码保护，并且处于关闭状态

输出栏中的操作

当段落处于密码保护状态时，且当前段落的编辑界面处于关闭状态，则当双击输出栏中的对象处于此段落时，将出现密码输入框。正确输入密码一次后，则段落的编辑界面处于打开状态，以后双击输出栏中的对象处于此段落时，就不出现密码输入框。

工程文件的操作

当前有密码保护段落的工程文件，如果对该工程文件选择另存为操作，则相应的段落还是处于密码保护中。

1.5 图形编程模块库

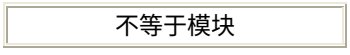
图形编程 模块库，包括了 IEC61131－3 中定义的功能块、一些常用的功能块。例如：

- 算术运算：加、减、乘、除、取模等；
- 触发器：RS、SR 等；
- 比较运算：大于等于、大于、小于等于、小于、等于、不等于；
- 转换运算：各种数据类型之间的相互转换；
- 选择运算：单选、多选；
- 逻辑运算：逻辑与、或、非、异或等；
- 数学运算：绝对值、余弦、正弦、指数、对数等；
- 计数器：加计数、减计数、加减计数等；
- 定时器：延时接通 TON、延时断开 TOFF、脉冲 TP；
- 输入处理：滤波、报警、温压补偿、折线插值；
- 控制模块：单回路 PID,串级控制；
- 通讯辅助：发送消息、接收消息；
- 累积函数：累积函数、累积量转换。

1.5.1 IEC 模 块 库

1. IEC 比较运算模块

IEC 比较运算模块包括了不等于、大于、大于等于、等于、小于以及小于等于 6 种模块，如下所示：



大于模块
大于等于模块
等于模块
小于模块
小于等于模块

1) 不等于模块

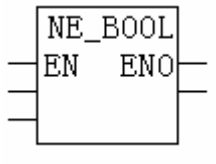
NE_BOOL 模块

简介

该模块的功能是对两个输入值进行比较，若输入值不等，则输出值为 ON，否则为 OFF。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if IN1 ≠ IN2 OUT = ON

if IN1 = IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	BOOL	第一输入
IN2	BOOL	第二输入
OUT	BOOL	输出

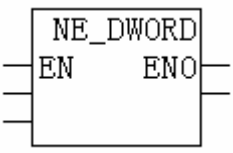
NE_DWORD 模块

简介

该模块的功能是对两个输入值进行比较，若输入值不等，则输出值为 ON，否则为 OFF。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if IN1 ≠ IN2 OUT = ON

if IN1 = IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	DWORD	第一输入
IN2	DWORD	第二输入

OUT	BOOL	输出

NE_FLOAT 模块

简介

该模块的功能是对两个输入值进行比较，若输入值不等，则输出值为 ON，否则为 OFF。EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

if IN1 ≠ IN2 OUT = ON

if IN1 = IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	FLOAT	第一输入
IN2	FLOAT	第二输入
OUT	BOOL	输出

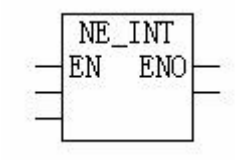
NE_INT 模块

简介

该模块的功能是对两个输入值进行比较，若输入值不等，则输出值为 ON，否则为 OFF。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if IN1 ≠ IN2 OUT = ON

if IN1 = IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
OUT	BOOL	输出

NE_SFLOAT 模块

简介

该模块的功能是对两个输入值进行比较，若输入值不等，则输出值为 ON，否则为 OFF。EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

if IN1 ≠ IN2 OUT = ON

if IN1 = IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
OUT	BOOL	输出

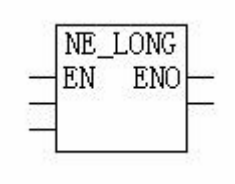
NE_LONG 模块

简介

该模块的功能是对两个输入值进行比较，若输入值不等，则输出值为 ON，否则为 OFF。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if IN1 ≠ IN2 OUT = ON

if IN1 = IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
OUT	BOOL	输出

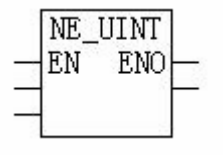
NE_UINT 模块

简介

该模块的功能是对两个输入值进行比较，若输入值不等，则输出值为 ON，否则为 OFF。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if IN1 ≠ IN2 OUT = ON

if IN1 = IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
OUT	BOOL	输出

NE_ULONG 模块

简介

该模块的功能是对两个输入值进行比较，若输入值不等，则输出值为 ON，否则为 OFF。EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

if IN1 ≠ IN2 OUT = ON

if IN1 = IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
OUT	BOOL	输出

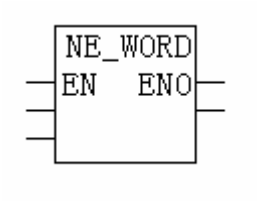
NE_WORD 模块

简介

该模块的功能是对两个输入值进行比较，若输入值不等，则输出值为 ON，否则为 OFF。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if IN1≠ IN2 OUT = ON

if IN1=IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	WORD	第一输入
IN2	WORD	第二输入
OUT	BOOL	输出

NE_BYTE 模块

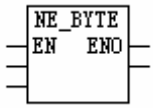
简介

该模块的功能是对两个输入值进行比较，若输入值不等，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if IN1≠ IN2 OUT = ON

if IN1=IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	BYTE	第一输入
IN2	BYTE	第二输入
OUT	BOOL	输出

2) 大于模块

GT_FLOAT 模块

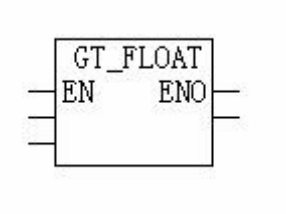
简介

该模块功能是检查第一个输入值是否大于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 > IN2

OUT =OFF if IN1 ≤ IN2

参数描述

参数	数据类型	含义
IN1	FLOAT	第一输入
IN2	FLOAT	第二输入
OUT	BOOL	输出值

GT_INT 模块

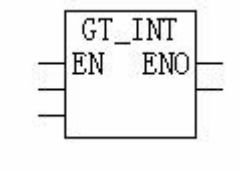
简介

模块功能是检查第一个输入值是否大于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 > IN2

OUT =OFF if IN1 ≤ IN2

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
OUT	BOOL	输出值

GT_LONG 模块

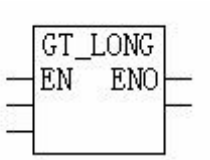
简介

模块功能是检查第一个输入值是否大于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 > IN2

OUT = OFF if IN1 ≤ IN2

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
OUT	BOOL	输出值

GT_SFLOAT 模块

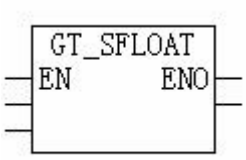
简介

模块功能是检查第一个输入值是否大于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 > IN2

OUT = OFF if IN1 ≤ IN2

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
OUT	BOOL	输出值

GT_UINT 模块

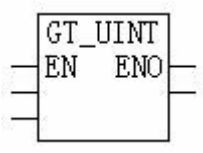
简介

模块功能是检查第一个输入值是否大于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 > IN2

OUT =OFF if IN1 ≤ IN2

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
OUT	BOOL	输出值

GT_ULONG 模块

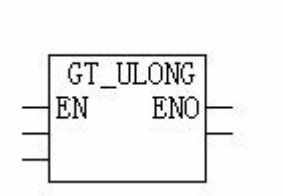
简介

模块功能是检查第一个输入值是否大于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 > IN2

OUT =OFF if IN1 ≤ IN2

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
OUT	BOOL	输出值

3) 大于等于模块

GE_FLOAT 模块

简介

模块功能是检查第一个输入值是否大于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT = ON if IN1 ≥ IN2

OUT = OFF if IN1 〈 IN2

参数描述

参数	数据类型	含义
IN1	FLOAT	第一输入
IN2	FLOAT	第二输入
OUT	BOOL	输出值

GE_INT 模块

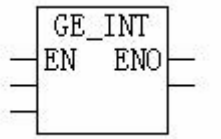
简介

模块功能是检查第一个输入值是否大于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 ≥ IN2

OUT = OFF if IN1 〈 IN2

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
OUT	BOOL	输出值

GE_LONG 模块

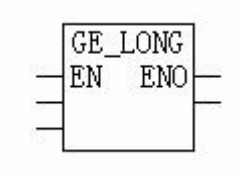
简介

模块功能是检查第一个输入值是否大于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 ≥ IN2

OUT = OFF if IN1 〈 IN2

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
OUT	BOOL	输出值

GE_UINT 模块

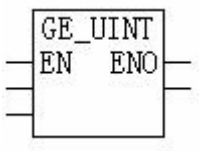
简介

模块功能是检查第一个输入值是否大于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 ≥ IN2

OUT = OFF if IN1 〈 IN2

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
OUT	BOOL	输出值

GE_SFLOAT 模块

简介

模块功能是检查第一个输入值是否大于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT = ON if IN1 ≥ IN2

OUT = OFF if IN1 〈 IN2

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
OUT	BOOL	输出值

GE_ULONG 模块

简介

模块功能是检查第一个输入值是否大于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT = ON if IN1 ≥ IN2

OUT = OFF if IN1 〈 IN2

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
OUT	BOOL	输出值

4) 等于模块

EQ_BOOL 模块

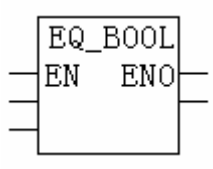
简介

该模块功能是检查第一个输入值是否等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON , if　IN1 = IN2

OUT = OFF , if　IN1 ≠ IN2

参数描述

参数	数据类型	含义
IN1	BOOL	第一输入
IN2	BOOL	第二输入
OUT	BOOL	输出值

EQ_DWORD 模块

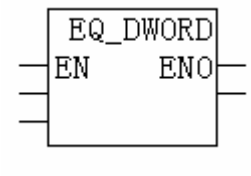
简介

该模块功能是检查第一个输入值是否等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON , if　IN1 = IN2

OUT = OFF , if　IN1 ≠ IN2

参数描述

参数	数据类型	含义
IN1	DWORD	第一输入
IN2	DWORD	第二输入
OUT	BOOL	输出值

EQ_INT 模块

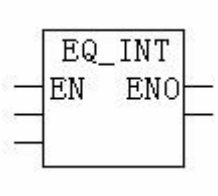
简介

该模块功能是检查第一个输入值是否等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON , if　IN1 = IN2

OUT = OFF , if　IN1 ≠ IN2

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
OUT	BOOL	输出值

EQ_FLOAT 模块

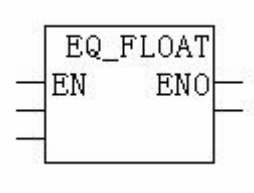
简介

该模块功能是检查第一个输入值是否等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON , if　IN1 = IN2

OUT = OFF , if　IN1 ≠ IN2

参数描述

参数	数据类型	含义
IN1	FLOAT	第一输入
IN2	FLOAT	第二输入
OUT	BOOL	输出值

EQ_LONG 模块

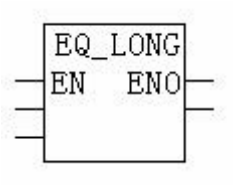
简介

该模块功能是检查第一个输入值是否等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON , if　IN1 = IN2

OUT = OFF , if　IN1 ≠ IN2

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
OUT	BOOL	输出值

EQ_SFLOAT 模块

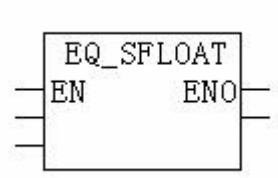
简介

该模块功能是检查第一个输入值是否等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON , if　IN1 = IN2

OUT = OFF , if　IN1 ≠ IN2

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
OUT	BOOL	输出值

EQ_UINT 模块

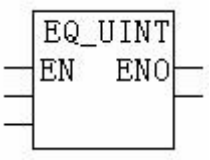
简介

该模块功能是检查第一个输入值是否等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON , if　IN1 = IN2

OUT = OFF , if　IN1 ≠ IN2

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
OUT	BOOL	输出值

EQ_ULONG 模块

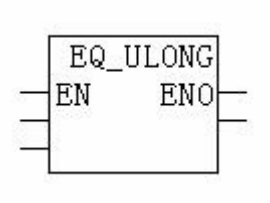
简介

该模块功能是检查第一个输入值是否等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON , if　IN1 = IN2

OUT = OFF , if　IN1 ≠ IN2

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
OUT	BOOL	输出值

EQ_WORD 模块

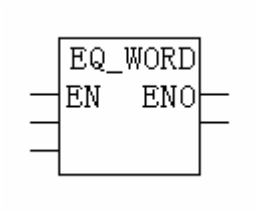
简介

该模块功能是检查第一个输入值是否等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON , if　IN1 = IN2

OUT = OFF , if　IN1 ≠ IN2

参数描述

参数	数据类型	含义
IN1	WORD	第一输入
IN2	WORD	第二输入
OUT	BOOL	输出值

EQ_BYTE 模块

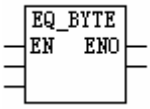
简介

该模块功能是检查第一个输入值是否等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON , if　IN1 = IN2

OUT = OFF , if　IN1 ≠ IN2

参数描述

参数	数据类型	含义
IN1	BYTE	第一输入
IN2	BYTE	第二输入
OUT	BOOL	输出值

5)　小于模块

LT_FLOAT 模块

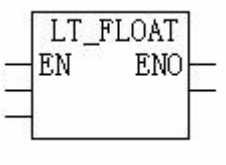
简介

该模块功能是检查第一个输入值是否小于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 < IN2

OUT =OFF if IN1 ≥ IN2

参数描述

参数	数据类型	含义
IN1	FLOAT	第一输入
IN2	FLOAT	第二输入
OUT	BOOL	输出值

LT_INT 模块

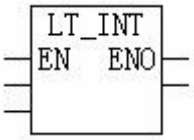
简介

该模块功能是检查第一个输入值是否小于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 < IN2

OUT =OFF if IN1 ≥ IN2

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
OUT	BOOL	输出值

LT_LONG 模块

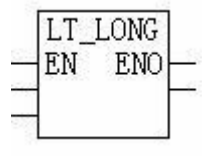
简介

该模块功能是检查第一个输入值是否小于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 < IN2

OUT = OFF if IN1 ≥ IN2

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
OUT	BOOL	输出值

LT_SFLOAT 模块

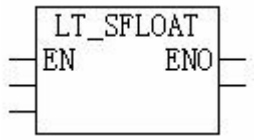
简介

该模块功能是检查第一个输入值是否小于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 < IN2

OUT = OFF if IN1 ≥ IN2

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
OUT	BOOL	输出值

LT_UINT 模块

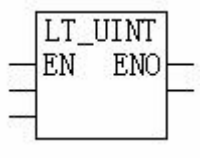
简介

该模块功能是检查第一个输入值是否小于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 < IN2

OUT = OFF if IN1 ≥ IN2

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
OUT	BOOL	输出值

LT_ULONG 模块

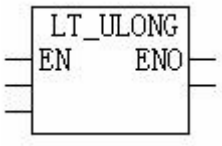
简介

该模块功能是检查第一个输入值是否小于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ON if IN1 < IN2

OUT =OFF if IN1 ≥ IN2

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
OUT	BOOL	输出值

6) 小于等于模块

LE_INT 模块

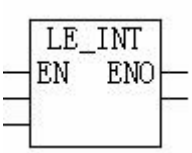
简介

该模块功能是检查第一个输入值是否小于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if IN1 ≤ IN2 OUT = ON

if IN1 > IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
OUT	BOOL	输出值

LE_FLOAT 模块

简介

该模块功能是检查第一个输入值是否小于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

if OUT = ON IN1 ≤ IN2

if OUT = OFF IN1 > IN2

参数描述

参数	数据类型	含义
IN1	FLOAT	第一输入
IN2	FLOAT	第二输入
OUT	BOOL	输出值

LE_LONG 模块

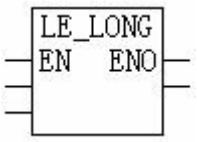
简介

该模块功能是检查第一个输入值是否小于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if IN1 ≤ IN2 OUT = ON

if IN1 > IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
OUT	BOOL	输出值

LE_SFLOAT 模块

简介

该模块功能是检查第一个输入值是否小于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

if IN1 ≤ IN2 OUT = ON

if IN1 > IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
OUT	BOOL	输出值

LE_UINT 模块

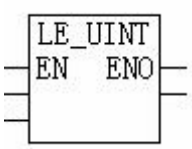
简介

该模块功能是检查第一个输入值是否小于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if IN1 ≤ IN2 OUT = ON

if IN1 > IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
OUT	BOOL	输出值

LE_ULONG 模块

简介

该模块功能是检查第一个输入值是否小于等于第二个输入值，若是，则输出值为 ON，否则为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

if IN1 ≤ IN2 OUT = ON

if IN1 > IN2 OUT = OFF

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
OUT	BOOL	输出值

2. IEC 计数定时模块

IEC 计数定时模块包括了 10 种不同功能的模块，分别是 RS 触发器、SR 触发器、上升沿触发器、下降沿触发器、定时器 TOFF、定时器 TON、定时器 TP、减计数器、增计数器以及增减计数器，如下所示：

RS 触发器模块

简介

该模块功能是用于 RS 存储，其中复位优先。

当 R1 = ON， Q1 就变为 OFF。

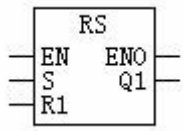
当 R1 = OFF，S = ON，则 Q1 = ON；

当 R1 = OFF，S = OFF，则 Q1 保持原状态。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



VOID RS(S,R1,Q1)

参数描述

参数	数据类型	含义
S	BOOL	置位
R1	BOOL	复位 (优先)
Q1	BOOL	输出

SR 触发器模块

简介

该模块功能是用于 RS 存储，其中置位优先。

当 S1 = ON，Q1 就变为 ON。

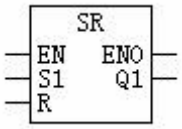
当 S1 = OFF，R = OFF 时, Q1 保持以前状态；

当 S1 = OFF，R = ON 时,Q1 = OFF 。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



VOID SR(S1,R,Q1)

参数描述

参数	数据类型	含义
S1	BOOL	置位(优先)
R	BOOL	复位
Q1	BOOL	输出

R_TRIG 模块

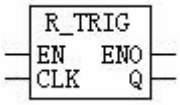
简介

该模块的功能是上升沿触发，即当 CLK 从 OFF 跳变为 ON 时，Q 在下一周期为 ON；其余情况下，Q 都为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



VOID R_TRIG(CLK,Q)

参数描述

参数	数据类型	含义
CLK	BOOL	输入
Q	BOOL	输出

F_TRIG 模块

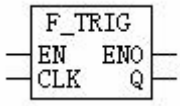
简介

该模块的功能是下降沿触发，即当 CLK 从 ON 跳变为 OFF 时，Q 在下一周期为 ON 其余情况下，Q 都为 OFF。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



VOID F_TRIG(CLK,Q)

参数描述

参数	数据类型	含义
CLK	BOOL	输入
Q	BOOL	输出值

TOFF 定时器模块

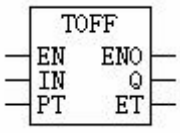
简介

该模块的功能是当 IN 从 ON 跳变为 OFF 时，产生一个延时输出。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

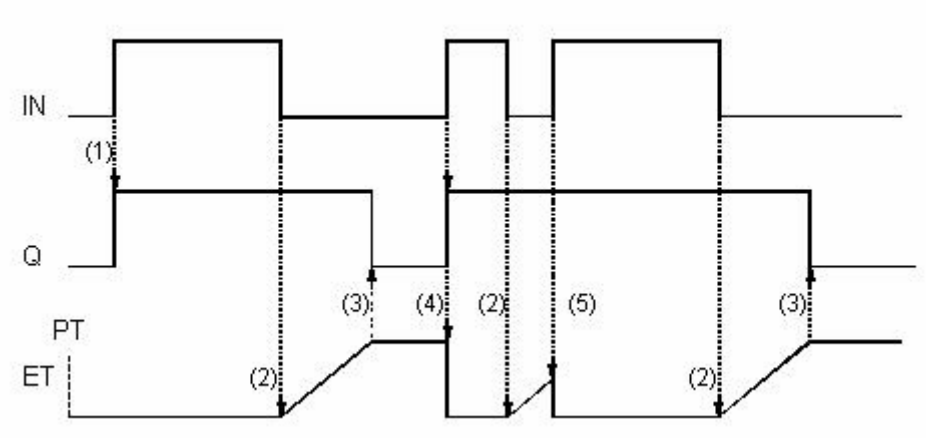


VOID TOFF(IN,PT,Q,ET)

参数描述

参数	数据类型	含义
IN	BOOL	输入
PT	ULONG	预置延时时间(单位毫秒)
Q	BOOL	输出状态
ET	ULONG	内部时钟

详细描述



任何时候如果 IN 为 ON，则 Q＝ON，ET＝0。

如果 IN 变为 OFF， 内部时钟 ET 将启动，以（系统运行周期×任务运行周期数）为单位增加，延时开始。（例如：系统运行周期在 SCKey 中设定为 500ms，SCControl 的任务管理中选定占 5 个周期，那么延时就以 2500ms 为单位增加）。当内部时钟 ET 达到 PT 值时, Q 将变为 OFF，ET＝PT。如果 IN 在 ET 达到 PT 值之前变为 ON，则 ET=0，Q＝ON。

TON 定时器模块

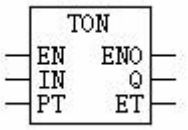
简介

该模块的功能是当 IN 从 OFF 跳变为 ON 时，产生一个延时输出。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

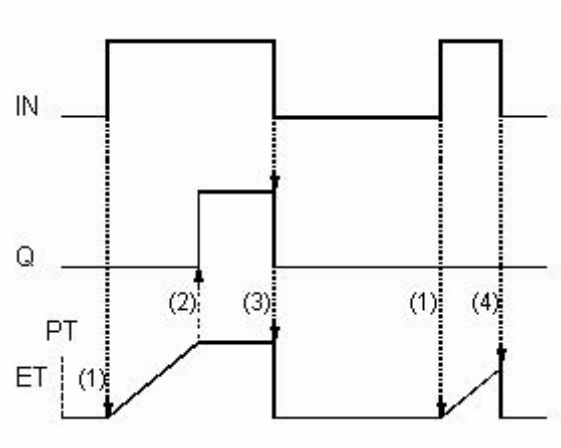


VOID TON(IN,PT,Q,ET)

参数描述

参数	数据类型	含义
IN	BOOL	输入
PT	ULONG	预置延时时间（单位毫秒）
Q	BOOL	输出状态
ET	ULONG	内部时钟

详细描述



任何时候如果 IN 为 OFF，则 Q＝OFF，ET=0。

如果 IN 变为 ON，内部时钟 ET 启动，以（系统运行周期×任务运行周期数）为单位增加，延时开始。（例如：系统运行周期在 SCKey 中设定为 500ms，SCControl 的任务管理中选定占 5 个周期，那么延时就以 2500ms 为单位增加）。当内部时钟 ET 达到 PT 值时, Q 变为 ON，ET＝PT。如果 IN 在 ET 达到 PT 值前变为 OFF，则 Q＝OFF，ET=0。

TP 定时器模块

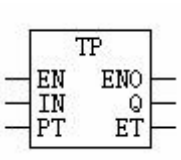
简介

该模块用于产生持续时间一定的脉冲。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

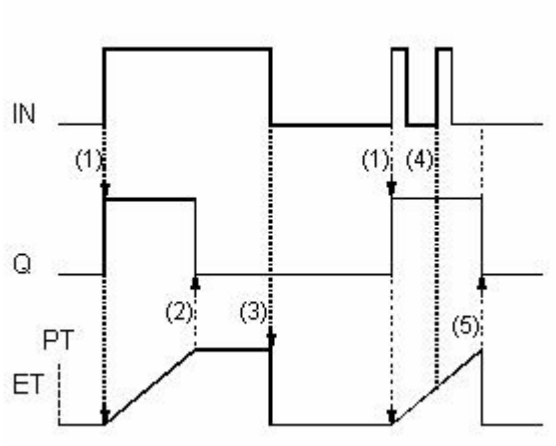


VOID TP(IN,PT,Q,ET)

参数描述

参数	数据类型	含义
IN	BOOL	输入
PT	ULONG	预置延时时间（单位毫秒）
Q	BOOL	输出状态
ET	ULONG	内部时钟

详细描述



如果 ET＝0，IN 变为 ON，则 Q 变为 ON，内部时钟(ET)启动，以（系统运行周期×任务运行周期数）为单位增加。（例如：系统运行周期在 SCKey 中设定为 500ms，SCControl 的任务管理中选定占 5 个周期，那么延时就以 2500ms 为单位增加）。如果 ET 尚未达到 PT 值，IN 变为 OFF，ET 与 Q 都不会受 IN 影响。当 ET 达到 PT 值, Q 将变为 OFF(与 IN 无关)。ET 达到 PT 值之后 IN 变为 OFF，则内部时钟停止，ET=0，Q＝OFF。

CTD 模块

简介

该模块（减计数器）的功能是：

当 LD＝ON 时，CV=PV。

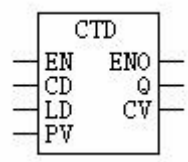
当 LD＝OFF，CD=ON 时，CV 每个周期自减 1，直到-32767 为止。

不管 LD 和 CD 的状态，当 CV>0 时 Q 为 OFF，当 CV≤0 时 Q 变为 ON。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



VOID CTD(CD,LD,PV,Q,CV)

参数描述

参数	数据类型	含义
CD	BOOL	计数开关
LD	BOOL	数据载入
PV	INT	预置数值
Q	BOOL	输出指示
CV	INT	计算值

CTU 模块

简介

该模块（增计数器）功能是：

当 R＝ON 时，CV=0。

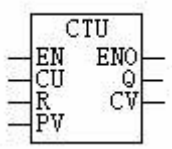
当 R=OFF，CU=ON 时，CV 每个周期自加 1。

不管 R 和 CU 的状态，当 CV<PV 时，Q=OFF，当 CV≥PV 时，Q 变为 ON。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



VOID CTU(CU,R,PV,Q,CV)

参数描述

参数	数据类型	含义
CU	BOOL	计数开关
R	BOOL	复位
PV	INT	预置数值
Q	BOOL	输出指示
CV	INT	计算值

CTUD 模块

简介

该模块（增减计数器）的功能是：

当 R=ON 时，CV=0，QU=OFF，QD=ON（增计数器清零）；

当 LD=ON 时，CV=PV，QU=ON，QD=OFF（减计数器清零）；

如果在 R 和 LD 端同时为 ON，则 R（复位）优先。

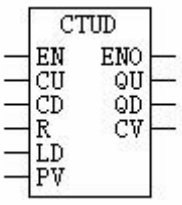
当 R=LD=OFF 时，若 CU=ON，CD=OFF，则为增计数器，CV 自加 1；至 CV≥PV 时，QU 变为 ON，CV 继续自加至 32767。

当 R=LD=OFF 时，若 CU=OFF，CD=ON，则为减计数器，CV 自减 1；至 CV≤0 时 QD 变为 ON，CV 继续自减到-32767 为止。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



VOID CTUD(CU,CD,R,LD,PV,QU,QD,CV)

参数描述

参数	数据类型	含义
CU	BOOL	增计数触发输入
CD	BOOL	减计数触发输入
R	BOOL	复位

LD	BOOL	数据载入
PV	INT	预置数值
QU	BOOL	增指示开关
QD	BOOL	减指示开关
CV	INT	计算值（真实值）

3. IEC 逻辑运算模块

IEC 逻辑运算模块包括了与、或、取反（非）、循环左移、循环右移、逻辑左移、逻辑右移及异或等功能运算。

逻辑运算

位与：对输入值每一位进行逻辑与操作，即 1&1=1,1&0=0,0&0=0。例如，10011001&00101101=00001001。

位或：对输入值每一位进行逻辑或操作，即 1|1=1,1|0=1,0|0=0。例如，10011001|00101101=10111101。

求反：对输入值每一位进行逻辑求反操作，即~1=0，~0=1。例如，~10011001=01100110。

逻辑左移：对输入值进行左移操作，从右边补零，例如 a=01000000，a<<1=10000000,a<<2=00000000。

逻辑右移：对输入值进行右移操作，从左边补零，例如 b=1001011111101101,b>>1=0100101111110110。

循环左移：对输入值进行左移操作，高位溢出项向低位填。例如，c=10101001,循环左移两位变为 10100110。

循环右移：对输入值进行右移操作，低位溢出项向高位填。例如，d=10101001,循环右移两位变为 01101010。

异或：对输入值每一位进行按位异或操作，即 1^1=0,1^0=1,0^0=0, 例如,10100101^01101100=11001001。

AND_BOOL 模块

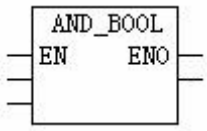
简介

该模块的功能是将输入值进行逻辑与操作，并将结果赋给输出值。输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT=IN1 AND IN2 AND ··· AND INn

参数描述

参数	数据类型	含义
IN1	BOOL	第一输入
IN2	BOOL	第二输入

INn	BOOL	第 n 个输入
OUT	BOOL	输出

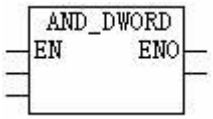
AND_DWORD 模块

简介

该模块的功能是将输入值进行逻辑与操作，并将结果赋给输出值。输入值个数不限。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT=IN1 AND IN2

参数描述

参数	数据类型	含义
IN1	DWORD	第一输入
IN2	DWORD	第二输入
OUT	DWORD	输出

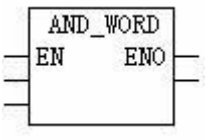
AND_WORD 模块

简介

该模块的功能是将输入值进行逻辑与操作，并将结果赋给输出值。输入值个数不限。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT=IN1 AND IN2

参数描述

参数	数据类型	含义
IN1	WORD	第一输入
IN2	WORD	第二输入
OUT	WORD	输出

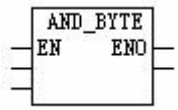
AND_BYTE 模块

简介

该模块的功能是将输入值进行逻辑与操作，并将结果赋给输出值。输入值个数不限。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT=IN1 AND IN2

参数描述

参数	数据类型	含义
IN1	BYTE	第一输入
IN2	BYTE	第二输入
OUT	BYTE	输出

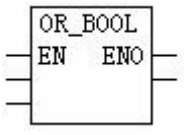
OR_BOOL 模块

简介

该模块功能是将输入值进行逻辑或操作，并将结果赋给输出值。输入值的个数不限。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT=IN1 OR IN2 OR ··· OR INn

参数描述

参数	数据类型	含义
IN1	BOOL	第一输入
IN2	BOOL	第二输入
INn	BOOL	第 n 个输入
OUT	BOOL	输出

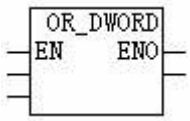
OR_DWORD 模块

简介

该模块功能是将输入值进行逻辑或操作，并将结果赋给输出值。输入值个数不限。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT=IN1 OR IN2

参数描述

参数	数据类型	含义
IN1	DWORD	第一输入
IN2	DWORD	第二输入
OUT	DWORD	输出

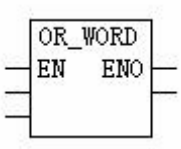
OR_WORD 模块

简介

该模块的功能是将输入值进行逻辑或操作，并将结果赋给输出值。输入值个数不限。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT=IN1 OR IN2

参数描述

参数	数据类型	含义
IN1	WORD	第一输入
IN2	WORD	第二输入
OUT	WORD	输出

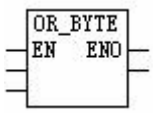
OR_BYTE 模块

简介

该模块的功能是将输入值进行逻辑或操作，并将结果赋给输出值。输入值个数不限。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT=IN1 OR IN2

参数描述

参数	数据类型	含义
IN1	BYTE	第一输入
IN2	BYTE	第二输入
OUT	BYTE	输出

NOT_BOOL 模块

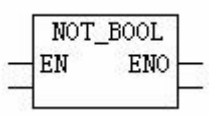
简介

该模块的功能是对 BOOL 型的输入值进行逻辑取反的操作，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = NOT IN1

参数描述

参数	数据类型	含义
IN1	BOOL	输入
OUT	BOOL	输出

NOT_DWORD 模块

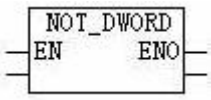
简介

该模块的功能是将输入值进行逻辑取反操作，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = NOT IN1

参数描述

参数	数据类型	含义
IN1	DWORD	输入
OUT	DWORD	输出

NOT_WORD 模块

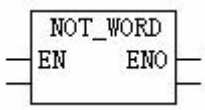
简介

该模块的功能是将输入值进行逻辑取反操作，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = NOT IN1

参数描述

参数	数据类型	含义
IN1	WORD	输入
OUT	WORD	输出

NOT_BYTE 模块

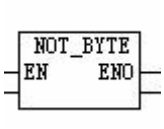
简介

该模块的功能是将输入值进行逻辑取反操作，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = NOT IN1

参数描述

参数	数据类型	含义
IN1	BYTE	输入
OUT	BYTE	输出

SHL_DWORD 模块

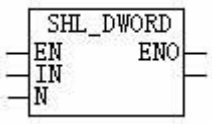
简介

该模块功能是将输入值 IN 进行左移（从右边填零），并将结果赋给输出值 OUT。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



OUT = SHL_DWORD(IN,N)

参数描述

参数	数据类型	含义
IN	DWORD	输入
N	UINT	左移位数
OUT	DWORD	输出

SHL_WORD 模块

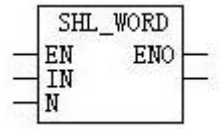
简介

该模块功能是将输入值 IN 进行左移（从右边填零），并将结果赋给输出值 OUT。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



OUT = SHL_WORD(IN,N)

参数描述

参数	数据类型	含义
IN	WORD	输入
N	UINT	左移位数
OUT	WORD	输出

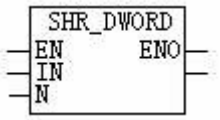
SHR_DWORD 模块

简介

该模块功能是将输入值 IN 进行右移（从左边填零），并将结果赋给输出值 OUT。EN 和 ENO 能作为附加参数加以设置。

表示

符号



OUT = SHR_DWORD(IN,N)

参数描述

参数	数据类型	含义
IN	DWORD	输入
N	UINT	右移位数
OUT	DWORD	输出

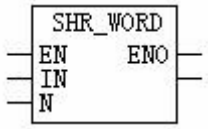
SHR_WORD 模块

简介

该模块功能是将输入值 IN 进行右移（从左边填零），并将结果赋给输出值 OUT。EN 和 ENO 能作为附加参数加以设置。

表示

符号



OUT = SHR_WORD(IN,N)

参数描述

参数	数据类型	含义
IN	WORD	输入
N	UINT	右移位数

OUT	WORD	输出

ROL_DWORD 模块

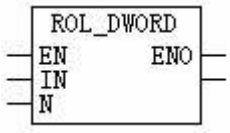
简介

该模块功能是将输入值 IN 进行循环左移，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



OUT = ROL_DWORD(IN,N)

参数描述

参数	数据类型	含义
IN	DWORD	输入
N	UINT	移位位数
OUT	DWORD	输出

ROL_WORD 模块

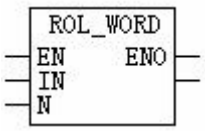
简介

该模块功能是将输入值 IN 进行循环左移，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



OUT = ROL_WORD(IN,N)

参数描述

参数	数据类型	含义
IN	WORD	输入值
N	UINT	移位位数
OUT	WORD	输出

ROR_DWORD 模块

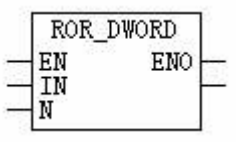
简介

该模块功能是将输入值 IN 进行循环右移，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



参数描述

参数	数据类型	含义
IN	DWORD	输入值
N	UINT	移位位数
OUT	DWORD	输出

ROR_WORD 模块

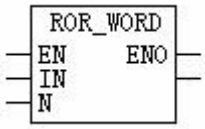
简介

该模块功能是将输入值 IN 进行循环右移，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



OUT = ROR_WORD(IN,N)

参数描述

参数	数据类型	含义
IN	WORD	输入值
N	UINT	移位位数
OUT	WORD	输出

XOR_BOOL 模块

简介

该模块功能是将输入值进行逻辑异或操作，并将结果赋给输出值。输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT=IN1 XOR IN2 XOR ··· XOR INn

参数描述

参数	数据类型	含义
IN1	BOOL	第一输入

IN2	BOOL	第二输入
INn	BOOL	第 n 个输入
OUT	BOOL	输出

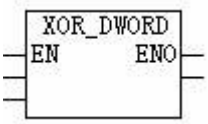
XOR_DWORD 模块

简介

该模块的功能是将输入值进行逻辑异或操作，并将结果赋给输出值。输入值个数不限。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT=IN1 XOR IN2

参数描述

参数	数据类型	含义
IN1	DWORD	第一输入
IN2	DWORD	第二输入
OUT	DWORD	输出

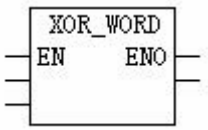
XOR_WORD 模块

简介

该模块的功能是将输入值进行逻辑异或操作，并将结果赋给输出值。输入值个数不限。EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT=IN1 XOR IN2

参数描述

参数	数据类型	含义
IN1	WORD	第一输入
IN2	WORD	第二输入
OUT	WORD	输出

XOR_BYTE 模块

简介

该模块的功能是将输入值进行逻辑异或操作，并将结果赋给输出值。输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT=IN1 XOR IN2

参数描述

参数	数据类型	含义
IN1	BYTE	第一输入
IN2	BYTE	第二输入
OUT	BYTE	输出

4. IEC 数学函数模块

IEC 数学函数模块包容了对数、反正弦、反余弦、反正切、工程正弦、工程余弦、工程正切、绝对值、幂函数、平方根、指数函数、正弦、余弦、正切等 20 种功能模块，如下所示：

ACOS 模块

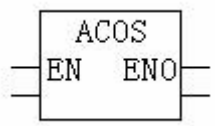
简介

该模块功能是计算输入值的反余弦值，并将结果以弧度的形式赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT= arccos (IN)

IN 的取值范围：[-π/2，π/2]

OUT 的取值范围：[0 ，JI]

参数描述

参数	数据类型	含义
IN	FLOAT 型	输入值
OUT	FLOAT 型	输出值 (弧度)

ATAN 模块

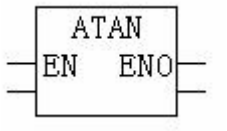
简介

该模块的功能是计算输入值的反正切值，并将结果以弧度的形式赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT =arctan (IN)

OUT 的取值范围：[-Л/2 ，+Л/2]

参数描述

参数	数据类型	含义
IN	FLOAT 型	输入
OUT	FLOAT 型	输出（弧度）

ATAN2 模块

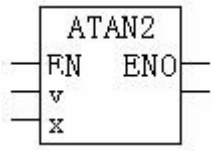
简介

该模块的功能是计算坐标（x,y)对应的反正切值，并将结果以弧度的形式赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = ± arctan (|y/x|)

OUT 的取值范围：（-Л/2，+Л/2）

参数描述

参数	数据类型	含义
Y	FLOAT 型	输入 Y 坐标
X	FLOAT 型	输入 X 坐标
OUT	FLOAT 型	输出（弧度）

ASIN 模块

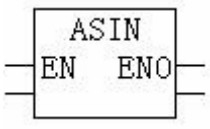
简介

该模块功能是计算输入值的反正弦值，并将结果以弧度的形式赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT=arcsin (IN)

IN 的取值范围：[-JI/2 ， +JI/2]

OUT 的取值范围：[-JI/2 ， +JI/2]

参数描述

参数	数据类型	含义
IN	FLOAT 型	输入
OUT	FLOAT 型	输出（弧度）

COSH 模块

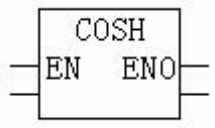
简介

该模块功能是计算输入值的工程余弦值，并将结果赋给输出值。输入值必须是弧度形式。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT= cosh (IN)

注：

$$\cosh(x) = \frac{e^{x} + e^{-x}}{2}$$

参数描述

参数	数据类型	含义
IN	FLOAT	输入值（ 弧度型 ）
OUT	FLOAT	输出值

TANH 模块

简介

该模块功能是计算输入值 IN 的工程正切值，并将结果赋给输出值 OUT。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT= TANH (IN)

注：

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

参数描述

参数	数据类型	含义
IN	FLOAT	输入
OUT	FLOAT	输出（弧度）

SINH 模块

简介

该模块功能是计算输入值 IN 的工程正弦值,并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT = sinh (IN)

注：

$$\cosh(x) = \frac{e^{x} - e^{-x}}{2}$$

参数描述

参数	数据类型	含义
IN	FLOAT	输入
OUT	FLOAT	输出

ABS_FLOAT 模块

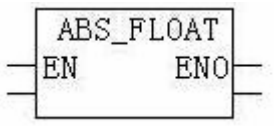
简介

该模块的功能是计算输入值的绝对值并将结果赋给输出值。

EN 和 ENO 可作为附加参数加以设置。

表示

符号



公式

OUT=|IN|

参数描述

参数	数据类型	含义
IN	FLOAT	输入值

OUT	FLOAT	输出值

ABS_INT 模块

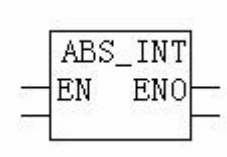
简介

该模块的功能是计算输入值的绝对值，并将结果赋给输出值。

EN 和 ENO 可作为附加参数加以设置。

表示

符号



公式

OUT=|IN|

参数描述

参数	数据类型	含义
IN	INT	输入值
OUT	INT	输出值

ABS_LONG 模块

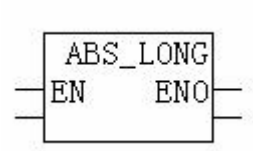
简介

该模块的功能是计算输入值的绝对值，并将结果赋给输出值。

EN 和 ENO 可作为附加参数加以设置。

表示

符号



公式

OUT=|IN|

参数描述

参数	数据类型	含义
IN	LONG	输入值
OUT	LONG	输出值

ABS_SFLOAT 模块

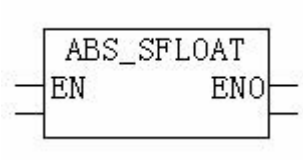
简介

该模块的功能是计算 SFLOAT 型输入值的绝对值并将结果赋给输出值。

EN 和 ENO 可作为附加参数加以设置。

表示

符号



公式

OUT=|IN|

参数描述

参数	数据类型	含义
IN	SFLOAT	输入值
OUT	SFLOAT	输出值

POW 模块

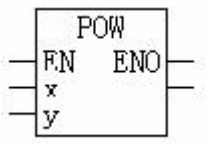
简介

该模块功能是计算 y 为指数，x 为底的幂级数，并将结果赋给输出值 OUT。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



OUT = POW(X,Y)

参数描述

参数	数据类型	含义
IN1	FLOAT	底数
IN2	FLOAT	指数
OUT	FLOAT	输出

SQRT_FLOAT 模块

简介

该模块功能是计算 FLOAT 型输入值 IN 的平方根，并将结果赋给输出值 OUT。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT= SQRT（IN）

参数描述

参数	数据类型	含义

IN	FLOAT	输入
OUT	FLOAT	输出

SQRT_SFLOAT 模块

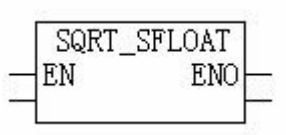
简介

该模块功能是计算 SFLOAT 型输入值 IN 的平方根，并将结果赋给输出值 OUT。SQRT_SFLOAT 的输入范围为 0－1。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT= SQRT（IN）

参数描述

参数	数据类型	含义
IN	SFOLAT	输入
OUT	SFLOAT	输出

COS 模块

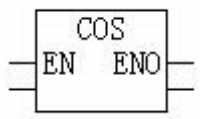
简介

该模块功能是计算输入值的余弦值，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT= COS(IN)

参数描述

参数	数据类型	含义
IN	FLOAT	输入值（弧度型）
OUT	FLOAT	输出值

TAN 模块

简介

该模块功能是计算输入值 IN 的正切值，并将结果赋给输出值 OUT。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT=TAN(IN)

参数描述

参数	数据类型	含义
IN	FLOAT	输入（弧度）
OUT	FLOAT	输出

SIN 模块

简介

该模块功能是计算输入值 IN 的正弦值，并将结果赋给输出值 OUT。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT = SIN(IN)

参数描述

参数	数据类型	含义
IN	FLOAT	输入
OUT	FLOAT	输出

EXP 模块

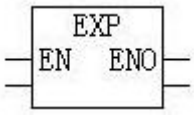
简介

该模块功能是计算以 e 为底，输入值 IN 为指数的幂级数，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = EXP(IN)

式中： $EXP(x) = e^x$

参数描述

参数	数据类型	含义

IN	FLOAT	输入
OUT	FLOAT	输出值

LN 模块

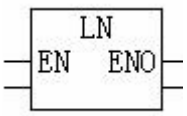
简介

该模块功能是计算输入值自然对数，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = LN(IN)　其中，IN>0

参数描述

参数	数据类型	含义
IN	FLOAT	输入
OUT	FLOAT	输出值

LOG 模块

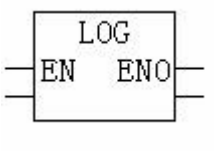
简介

该模块功能是计算以 10 为底的对数，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = LOG(IN)

其中，IN>0

参数描述

参数	数据类型	含义
IN	FLOAT	输入
OUT	FLOAT	输出值

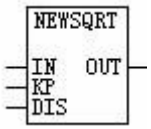
扩展开方模块

简介

该模块用来求[0.0，1.0]范围内半浮点数的平方根，并根据输入的斜率和偏移量对开平方后的值进行线性补偿。

表示

符号



算法

该模块只适用于[0.0，1.0]内的半浮点数：

当输入 IN 小于 0.0，输出 OUT=0.0；

当输入 IN 大于 1.0，输出 OUT=1.0；

当输入在[0.0，1.0]内，该模块先把这段区间分成 32 等分，先求出个各分点对应的平方根的值，把它们存放在一个数组内，这样就把一段曲线分成一段段的折线，当输入 IN 落在哪段折线内，在这段折线内就可以按线段来求值了。

最后对所得的平方根值进行线性补偿后输出，其公式为：

OUT=Kp×sqrt（IN）＋DIS

参数描述

参数	数据类型	含义
IN	SFLOAT	输入
KP	SFLOAT	线性补偿时的斜率
DIS	SFLOAT	线性补偿时的位移
OUT	SFLOAT	输出

5. IEC 算术运算模块

IEC 算术运算模块包括了乘法、除法、赋值、加法、减法、平均值及取模等几种运算功能模块，如下所示：

- 乘法模块
- 除法模块
- 赋值模块
- 加法模块
- 减法模块
- 平均值模块
- 取模模块

1) 乘法模块

MUL_FLOAT 模块

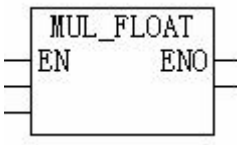
简介

该模块的功能是将输入值相乘，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1×IN2

参数描述

参数	数据类型	含义
IN1	FLOAT	第一输入
IN2	FLOAT	第二输入
OUT	FLOAT	输出

MUL_INT 模块

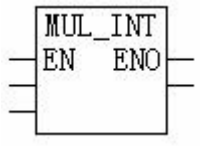
简介

该模块的功能是将输入值相乘，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1×IN2

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
OUT	INT	输出

MUL_LONG 模块

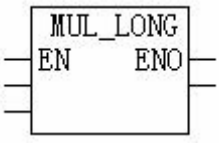
简介

该模块的功能是将输入值相乘，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1×IN2

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
OUT	LONG	输出

MUL_SFLOAT 模块

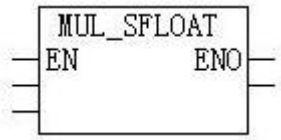
简介

该模块的功能是将输入值相乘，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1×IN2

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
OUT	SFLOAT	输出

MUL_UINT 模块

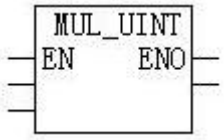
简介

该模块的功能是将输入值相乘，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1×IN2

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
OUT	UINT	输出

MUL_ULONG 模块

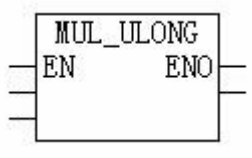
简介

该模块的功能是将输入值相乘，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1×IN2

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
OUT	ULONG	输出

2) 除法模块

DIV_FLOAT 模块

简介

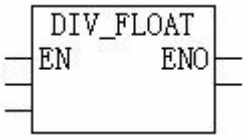
该模块的功能是将输入值相除，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

如果除数为 0，返回 32 进制：0x0000ffff。

表示

符号



公式

OUT = IN1/IN2

参数描述

参数	数据类型	含义
IN1	FLOAT	第一输入
IN2	FLOAT	第二输入
OUT	FLOAT	输出值

DIV_INT 模块

简介

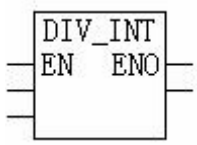
该模块的功能是将输入值相除，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

如果除数为 0，返回 32 进制：0x0000ffff。

表示

符号



公式

OUT = IN1/IN2

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
OUT	INT	输出值

DIV_LONG 模块

简介

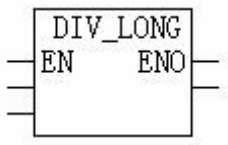
该模块的功能是将输入值相除，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

如果除数为 0，返回 32 进制：0x0000ffff。

表示

符号



公式

OUT = IN1/IN2

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
OUT	LONG	输出值

DIV_SFLOAT 模块

简介

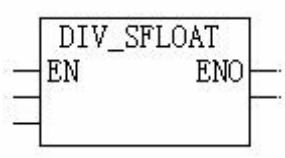
该模块的功能是将输入值相除，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

如果除数为 0，返回 32 进制：0x0000ffff。

表示

符号



公式

OUT = IN1/IN2

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
OUT	SFLOAT	输出值

DIV_UINT 模块

简介

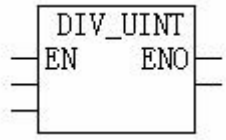
该模块的功能是将输入值相除，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

如果除数为 0，返回 32 进制：0x0000ffff。

表示

符号



公式

OUT = IN1/IN2

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
OUT	UINT	输出值

DIV_ULONG 模块

简介

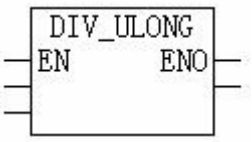
该模块的功能是将输入值 IN1 除以 IN2，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

如果除数为 0，返回 32 进制：0x0000ffff。

表示

符号



公式

OUT = IN1/IN2

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
OUT	ULONG	输出值

3) 赋值模块

MOVE_BOOL 模块

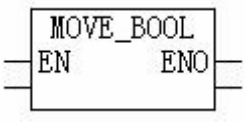
简介

该模块的功能是将输入值赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1

参数描述

参数	数据类型	含义
IN1	BOOL	第一输入
OUT	BOOL	输出

MOVE_DWORD 模块

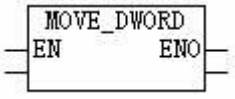
简介

该模块的功能是将输入值赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1

参数描述

参数	数据类型	含义
IN1	DWORD	第一输入
OUT	DWORD	输出

MOVE_FLOAT 模块

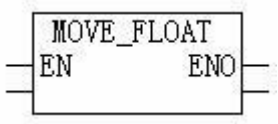
简介

该模块的功能是将输入值赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1

参数描述

参数	数据类型	含义
IN1	FLOAT	第一输入
OUT	FLOAT	输出

MOVE_INT 模块

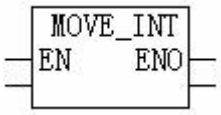
简介

该模块的功能是将输入值赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1

参数描述

参数	数据类型	含义
IN1	INT	第一输入
OUT	INT	输出

MOVE_LONG 模块

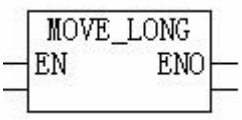
简介

该模块的功能是将输入值赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
OUT	LONG	输出

MOVE_SFLOAT 模块

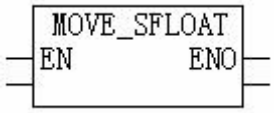
简介

该模块的功能是将输入值赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
OUT	SFLOAT	输出

MOVE_UINT 模块

简介

该模块的功能是将输入值赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT = IN1

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
OUT	UINT	输出

MOVE_ULONG 模块

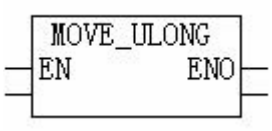
简介

该模块的功能是将输入值赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
OUT	ULONG	输出

MOVE_WORD 模块

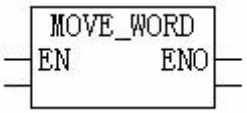
简介

该模块的功能是将输入值赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1

参数描述

参数	数据类型	含义
IN1	WORD	第一输入
OUT	WORD	输出

MOVE_BYTE 模块

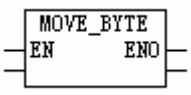
简介

该模块的功能是将输入值赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1

参数描述

参数	数据类型	含义
IN1	BYTE	第一输入

OUT	BYTE	输出

4) 加法模块

加法模块中，可分为 FLOAT、INT、LONG、SFLOAT、UINT、ULONG 等数据类型的 6 个模块，如下所示：

● 加（FLOAT）
● 加（INT）
● 加（LONG）
● 加（SFLOAT）
● 加（UINT）
● 加（ULONG）

ADD_FLOAT 模块

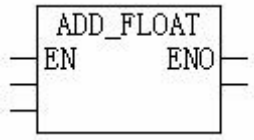
简介

该模块的功能是将输入值相加，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1+IN2

参数描述

参数	数据类型	含义
IN1	FLOAT	第一输入
IN2	FLOAT	第二输入
OUT	FLOAT	输出

ADD_INT 模块

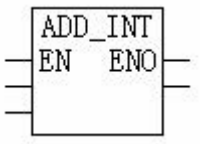
简介

该模块的功能是将输入值相加，并将结果赋给输出值。输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1+IN2

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
OUT	INT	输出

ADD_LONG 模块

简介

该模块的功能是将输入值相加，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT = IN1+IN2

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
OUT	LONG	输出

ADD_SFLOAT 模块

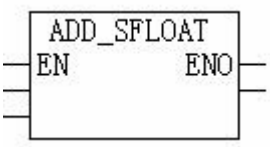
简介

该模块的功能是将输入值相加，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1+IN2

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
OUT	SFLOAT	输出

ADD_UINT 模块

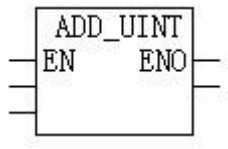
简介

该模块的功能是将输入值相加，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1+IN2

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
OUT	UINT	输出

ADD_ULONG 模块

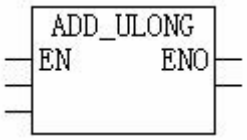
简介

该模块的功能是将输入值相加，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1+IN2

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
OUT	ULONG	输出

5) 减法模块

SUB_FLOAT 模块

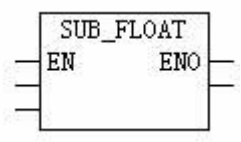
简介

该模块的功能是将输入值相减，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1-IN2

参数描述

参数	数据类型	含义
IN1	FLOAT	第一输入
IN2	FLOAT	第二输入
OUT	FLOAT	输出

SUB_INT 模块

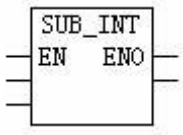
简介

该模块的功能是将输入值相减，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1-IN2

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
OUT	INT	输出

SUB_LONG 模块

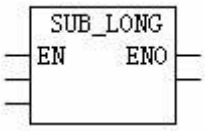
简介

该模块的功能是将输入值相减，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1-IN2

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
OUT	LONG	输出

SUB_SFLOAT 模块

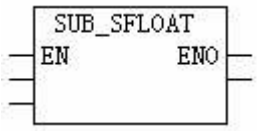
简介

该模块的功能是将输入值相减，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1-IN2

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
OUT	SFLOAT	输出

SUB_UINT 模块

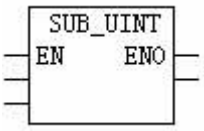
简介

该模块的功能是将输入值相减，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1-IN2

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
OUT	UINT	输出

SUB_ULONG 模块

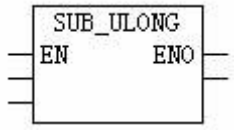
简介

该模块的功能是将输入值相减，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = IN1-IN2

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
OUT	ULONG	输出

6) 平均值模块

AVE_INT 模块

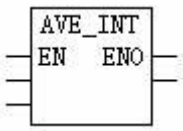
简介

该模块的功能是求输入值的平均值，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = （IN1+IN2+ ---）/N

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
OUT	INT	输出

AVE_FLOAT 模块

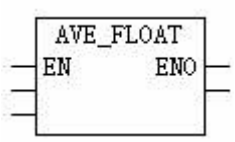
简介

该模块的功能是求输入值的平均值，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = （IN1+IN2+ ---）/ N

参数描述

参数	数据类型	含义
IN1	FLOAT	第一输入
IN2	FLOAT	第二输入
OUT	FLOAT	输出

AVE_LONG 模块

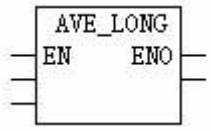
简介

该模块的功能是求输入值的平均值，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = （IN1+IN2+ ---）/ N

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
OUT	LONG	输出

AVE_SFLOAT 模块

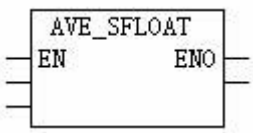
简介

该模块的功能是求输入值的平均值，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = （IN1+IN2+ --- ）/ N

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
OUT	SFLOAT	输出

AVE_UINT 模块

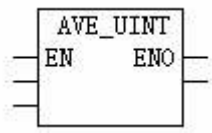
简介

该模块的功能是求输入值的平均值，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = （IN1+IN2+ ---）/ N

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
OUT	UINT	输出

AVE_ULONG 模块

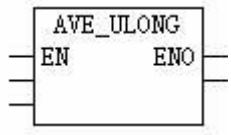
简介

该模块的功能是求输入值的平均值，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = （IN1+IN2+ --- ）/ N

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
OUT	ULONG	输出

7) 取模模块

MOD_INT 模块

简介

该模块的功能是将输入值相除，并将余数赋给 OUT1，商数赋给 OUT2。

EN 和 ENO 能作为附加参数加以设置。

如果除数为 0，那么输出为 0。

表示

符号

公式

OUT1 = IN1 ％ IN2

OUT2 = (IN1- OUT1)/IN2

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
OUT1	INT	余数
OUT2	INT	取整

MOD_LONG 模块

简介

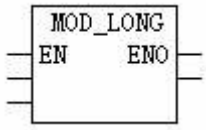
该模块的功能是将输入值相除，并将余数赋给 OUT1，商数赋给 OUT2。

EN 和 ENO 能作为附加参数加以设置。

如果除数为 0，那么输出为 0。

表示

符号



公式

OUT1 = IN1 ％ IN2

OUT2 = (IN1- OUT1)/IN2

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
OUT1	LONG	余数
OUT2	LONG	商数

MOD_ULONG 模块

简介

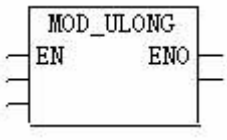
该模块的功能是将输入值相除，并将余数赋给 OUT1，商数赋给 OUT2。

EN 和 ENO 能作为附加参数加以设置。

如果除数为 0，那么输出为 0。

表示

符号



公式

OUT1 = IN1 ％ IN2

OUT2 = (IN1- OUT1)/IN2

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
OUT1	ULONG	余数
OUT2	ULONG	商数

MOD_UINT 模块

简介

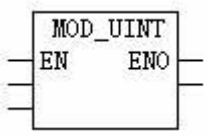
该模块的功能是将输入值相除，并将余数赋给 OUT1，商数赋给 OUT2。

EN 和 ENO 能作为附加参数加以设置。

如果除数为 0，那么输出为 0。

表示

符号



公式

OUT1 = IN1 ％ IN2

OUT2 = (IN1- OUT1)/IN2

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
OUT1	UINT	余数
OUT2	UINT	商数

6. IEC 转换函数模块

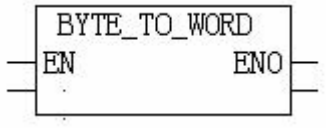
BYTE_TO_WORD 模块

简介

该模块将 BYTE 型变量转换成 WORD 型变量。

表示

符号



OUT = (WORD)IN

参数描述

参数	数据类型	含义
IN	BYTE	输入值
OUT	WORD	输出值

DENORM 模块

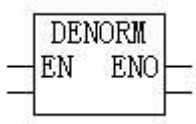
简介

该模块功能是将 INT 型的输入值解释为 SFLOAT 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = DENORM(IN)

参数描述

参数	数据类型	含义
IN	INT	输入值
OUT	SFLOAT	输出值

DWORD_TO_LONG 模块

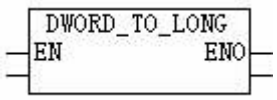
简介

该模块功能是将 DWORD 型的输入值转化为 LONG 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = DWORD_TO_LONG(IN)

参数描述

参数	数据类型	含义

IN	DWORD	输入值
OUT	LONG	输出值

DWORD_TO_ULONG 模块

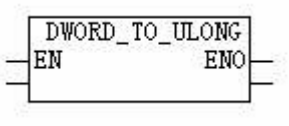
简介

该模块功能是将 DWORD 型的输入值转化为 ULONG 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = DWORD_TO_ULONG(IN)

参数描述

参数	数据类型	含义
IN	DWORD	输入值
OUT	ULONG	输出值

FLOAT_TO_INT 模块

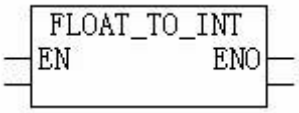
简介

该模块功能是将 FLOAT 型的输入值转化为 INT 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = FLOAT_TO_INT(IN)

参数描述

参数	数据类型	含义
IN	FLOAT	输入
OUT	INT	输出值

FLOAT_TO_LONG 模块

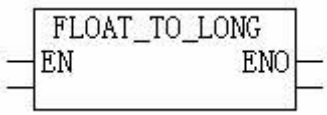
简介

该模块功能是将 FLOAT 型的输入值转化为 LONG 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = FLOAT_TO_LONG(IN)

参数描述

参数	数据类型	含义
IN	FLOAT	输入
OUT	LONG	输出值

FLOAT_TO_SFLOAT 模块

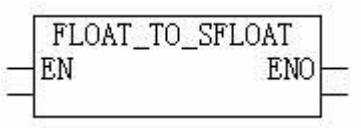
简介

该模块功能是将 FLOAT 型的输入值转化为 SFLOAT 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = FLOAT_TO_SFLOAT(IN)

参数描述

参数	数据类型	含义
IN	FLOAT	输入
OUT	SFLOAT	输出值

INT_TO_FLOAT 模块

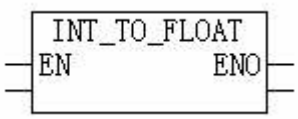
简介

该模块功能是将 INT 型的输入值转化为 FLOAT 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = INT_TO_FLOAT(IN)

参数描述

参数	数据类型	含义
IN1	INT	输入
OUT	FLOAT	输出

INT_TO_LONG 模块

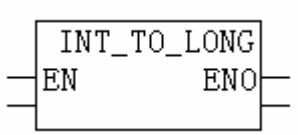
简介

该模块功能是将 INT 型的输入值转化为 LONG 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = INT_TO_LONG(IN)

参数描述

参数	数据类型	含义
IN	INT	输入
OUT	LONG	输出

INT_TO_SFLOAT 模块

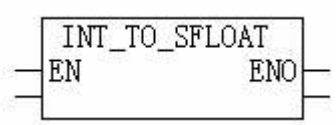
简介

该模块功能是将 INT 型的输入值转化为 SFLOAT 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = INT_TO_SFLOAT(IN)

参数描述

参数	数据类型	含义
IN1	INT	输入
OUT	SFLOAT	输出

INT_TO_LONG 模块

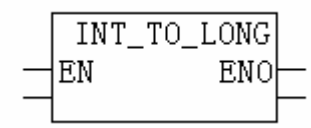
简介

该模块功能是将 INT 型的输入值转化为 LONG 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = INT_TO_LONG(IN)

参数描述

参数	数据类型	含义

IN	INT	输入
OUT	LONG	输出

INT_TO_WORD 模块

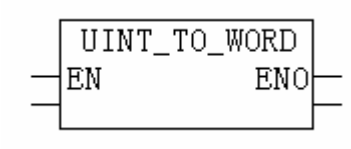
简介

该模块功能是将 INT 型的输入值转化为 WORD 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = INT_TO_WORD(IN)

参数描述

参数	数据类型	含义
IN1	INT	输入
OUT	WORD	输出

ULONG_TO_DWORD 模块

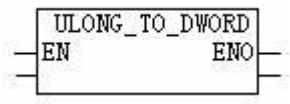
简介

该模块功能是将 ULONG 型的输入值转化为 DWORD 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = ULONG _TO_DWORD (IN)

参数描述

参数	数据类型	含义
IN	ULONG	输入
OUT	DWORD	输出

ULONG_TO_LONG 模块

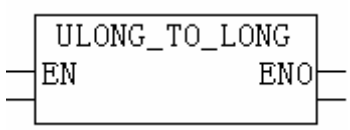
简介

该模块功能是将 ULONG 型的输入值转化为 LONG 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = ULONG _TO_LONG (IN)

参数描述

参数	数据类型	含义
IN	ULONG	输入
OUT	LONG	输出

ULONG_TO_UINT 模块

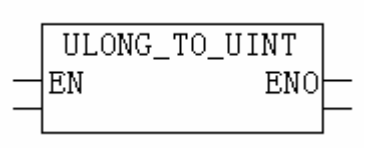
简介

该模块功能是将 ULONG 型的输入值转化为 UINT 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = ULONG _TO_UINT (IN)

参数描述

参数	数据类型	含义
IN	ULONG	输入
OUT	UINT	输出

WORD_TO_INT 模块

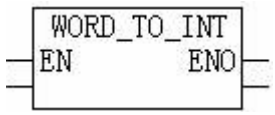
简介

该模块功能是将 WORD 型的输入值转化为 INT 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = WORD _TO_INT (IN)

参数描述

参数	数据类型	含义
IN	WORD	输入
OUT	INT	输出

WORD_TO_UINT 模块

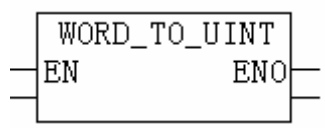
简介

该模块功能是将 WORD 型的输入值转化为 UINT 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



OUT = WORD _TO_UINT (IN)

参数描述

参数	数据类型	含义
IN	WORD	输入
OUT	UINT	输出

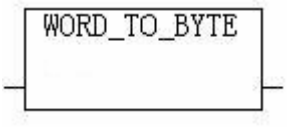
WORD_TO_BYTE 模块

简介

该模块将 WORD 型变量转换成 BYTE 型，当 WORD 型变量超出 BYTE 型变量范围时，超出部分被截去。

表示

符号



OUT = (BYTE)IN

参数描述

参数	数据类型	含义
IN	WORD	输入值
OUT	BYTE	输出值

7. IEC 选择运算模块

IEC 选择运算模块有如下所示，多选、限幅、选择、最大值、最小值等 5 种功能运算模块：

- 多选模块
- 限幅模块
- 选择模块
- 最大值模块
- 最小值模块

1) 多选模块

MUX_BOOL 模块

简介

该模块的功能是当 K=0 时将输入值 IN1 赋给输出值，当 K=1 时，将输入值 IN2 赋给输出值；当 K=n-1 时将输入值 INn 赋给输出值。输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

参数描述

参数	数据类型	含义
K	INT	选择的序号（从 0 开始）
IN1	BOOL	第一输入
IN2	BOOL	第二输入
INn	BOOL	第 n 个输入
OUT	BOOL	输出

SEL_DWORD 模块

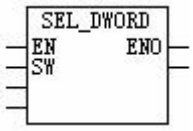
简介

该模块的功能是当 SW=OFF 时，将输入值 IN1 赋给输出值；当 SW=ON 时，将输入值 IN2 赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = SEL_DWORD(SW,IN1,IN2)

参数描述

参数	数据类型	含义
SW	BOOL	输入选择开关
IN1	DWORD	第一输入
IN2	DWORD	第二输入
OUT	DWORD	输出

MUX_FLOAT 模块

简介

该模块的功能是当 K=0 时将输入值 IN1 赋给输出值，当 K=1 时，将输入值 IN2 赋给输出值；当 K=n-1 时将输入值 INn 赋给输出值。输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

参数描述

参数	数据类型	含义
K	INT	选择的序号（从 0 开始）
IN1	FLOAT	第一输入
IN2	FLOAT	第二输入
INn	FLOAT	第 n 个输入
OUT	FLOAT	输出

MUX_INT 模块

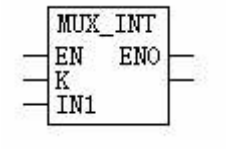
简介

该模块的功能是当 K=0 时将输入值 IN1 赋给输出值，当 K=1 时，将输入值 IN2 赋给输出值；当 K=n-1 时将输入值 INn 赋给输出值。输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



参数描述

参数	数据类型	含义
K	INT	选择的序号（从 0 开始）
IN1	INT	第一输入
IN2	INT	第二输入
INn	INT	第 n 个输入
OUT	INT	输出

MUX_LONG 模块

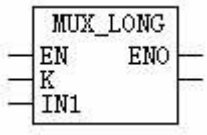
简介

该模块的功能是当 K=0 时将输入值 IN1 赋给输出值，当 K=1 时，将输入值 IN2 赋给输出值；当 K=n-1 时将输入值 INn 赋给输出值。输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



参数描述

参数	数据类型	含义
K	INT	选择的序号（从 0 开始）
IN1	LONG	第一输入
IN2	LONG	第二输入
INn	LONG	第 n 个输入
OUT	LONG	输出

MUX_SFLOAT 模块

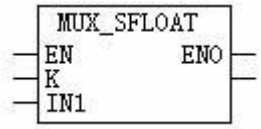
简介

该模块的功能是当 K=0 时将输入值 IN1 赋给输出值，当 K=1 时，将输入值 IN2 赋给输出值；当 K=n-1 时将输入值 INn 赋给输出值。输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



参数描述

参数	数据类型	含义
K	INT	选择的序号（从 0 开始）
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
INn	SFLOAT	第 n 个输入
OUT	SFLOAT	输出

MUX_ULONG 模块

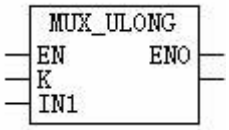
简介

该模块的功能是当 K=0 时将输入值 IN1 赋给输出值，当 K=1 时，将输入值 IN2 赋给输出值；当 K=n-1 时将输入值 INn 赋给输出值。输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



参数描述

参数	数据类型	含义
K	INT	选择的序号（从 0 开始）
IN1	ULONG	第一输入
IN2	ULONG	第二输入
INn	ULONG	第 n 个输入

OUT	ULONG	输出

MUX_UINT 模块

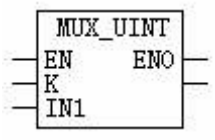
简介

该模块的功能是当 K=0 时将输入值 IN1 赋给输出值，当 K=1 时，将输入值 IN2 赋给输出值；当 K=n-1 时将输入值 INn 赋给输出值。输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



参数描述

参数	数据类型	含义
K	INT	选择的序号（从 0 开始）
IN1	UINT	第一输入
IN2	UINT	第二输入
INn	UINT	第 n 个输入
OUT	UINT	输出

MUX_WORD 模块

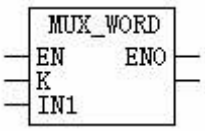
简介

该模块的功能是当 K=0 时将输入值 IN1 赋给输出值，当 K=1 时，将输入值 IN2 赋给输出值；当 K=n-1 时将输入值 INn 赋给输出值。输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



参数描述

参数	数据类型	含义
K	INT	选择的序号（从 0 开始）
IN1	WORD	第一输入
IN2	WORD	第二输入
INn	WORD	第 n 个输入
OUT	WORD	输出

2) 限幅模块

LIM_FLOAT 模块

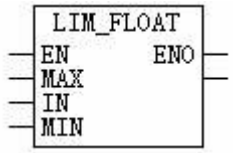
简介

该模块的功能是限幅，即当输入大于上限值时输出上限值，当输入小于下限值时输出下限值，否则输出输入值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if (IN ≥ MIN) & (IN ≤ MAX)　　OUT = IN

if IN < MIN　　OUT = MIN

if IN > MAX　　OUT = MAX

参数描述

参数	数据类型	含义
IN	FLOAT	第一输入
MAX	FLOAT	上限值
MIN	FLOAT	下限值
OUT	FLOAT	输出值

LIM_INT 模块

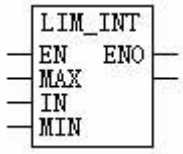
简介

该模块的功能是限幅，即当输入大于上限值时输出上限值，当输入小于下限值时输出下限值，否则输出输入值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if (IN ≥ MIN) & (IN ≤ MAX)　　OUT = IN

if IN < MIN　　OUT = MIN

if IN > MAX　　OUT = MAX

参数描述

参数	数据类型	含义
IN	INT	第一输入
MAX	INT	上限值
MIN	INT	下限值
OUT	INT	输出值

LIM_LONG 模块

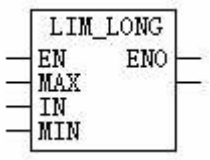
简介

该模块的功能是限幅，即当输入大于上限值时输出上限值， 当输入小于下限值时输出下限值，否则输出输入值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if (IN ≥ MIN) & (IN ≤ MAX)　　OUT = IN

if IN < MIN　　OUT = MIN

if IN > MAX　　OUT = MAX

参数描述

参数	数据类型	含义
IN	LONG	第一输入
MAX	LONG	上限值
MIN	LONG	下限值
OUT	LONG	输出值

LIM_SFLOAT 模块

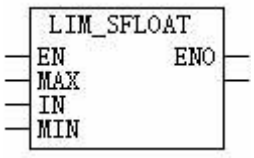
简介

该模块的功能是限幅，即当输入大于上限值时输出上限值，当输入小于下限值时输出下限值，否则输出输入值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if (IN ≥ MIN) & (IN ≤ MAX)　　OUT = IN

if IN < MIN　　OUT = MIN

if IN > MAX　　OUT = MAX

参数描述

参数	数据类型	含义
IN	SFLOAT	第一输入
MAX	SFLOAT	上限值
MIN	SFLOAT	下限值
OUT	SFLOAT	输出值

LIM_UINT 模块

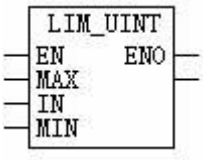
简介

该模块的功能是限幅，即当输入大于上限值时输出上限值，当输入小于下限值时输出下限值，否则输出输入值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if (IN ≥ MIN) & (IN ≤ MAX)　　OUT = IN

if IN < MIN　　OUT = MIN

if IN > MAX　　OUT = MAX

参数描述

参数	数据类型	含义
IN	UINT	第一输入
MAX	UINT	上限值
MIN	UINT	下限值
OUT	UINT	输出值

LIM_ULONG 模块

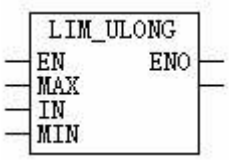
简介

该模块的功能是限幅，即当输入大于上限值时输出上限值，当输入小于下限值时输出下限值，否则输出输入值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

if (IN ≥ MIN) & (IN ≤ MAX)　　OUT = IN

if IN < MIN　　OUT = MIN

if IN > MAX　　OUT = MAX

参数描述

参数	数据类型	含义
IN	ULONG	第一输入
MAX	ULONG	上限值

MIN	ULONG	下限值
OUT	ULONG	输出值

3) 选择模块

SEL_BOOL 模块

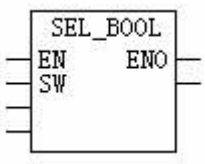
简介

该模块的功能是当 SW=OFF 时，将输入值 IN1 赋给输出值；当 SW=ON 时，将输入值 IN2 赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = SEL_BOOL(SW,IN1,IN2)

参数描述

参数	数据类型	含义
SW	BOOL	输入选择开关
IN1	BOOL	第一输入
IN2	BOOL	第二输入
OUT	BOOL	输出

SEL_DWORD 模块

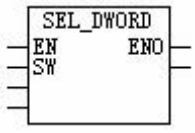
简介

该模块的功能是当 SW=OFF 时，将输入值 IN1 赋给输出值；当 SW=ON 时，将输入值 IN2 赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = SEL_DWORD(SW,IN1,IN2)

参数描述

参数	数据类型	含义
SW	BOOL	输入选择开关
IN1	DWORD	第一输入
IN2	DWORD	第二输入

OUT	DWORD	输出

SEL_FLOAT 模块

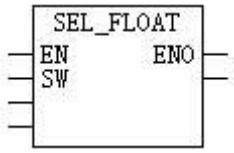
简介

该模块的功能是当 SW=OFF 时，将输入值 IN1 赋给输出值；当 SW=ON 时，将输入值 IN2 赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = SEL_FLOAT(SW,IN1,IN2)

参数描述

参数	数据类型	含义
SW	BOOL	输入选择开关
IN1	FLOAT	第一输入
IN2	FLOAT	第二输入
OUT	FLOAT	输出

SEL_INT 模块

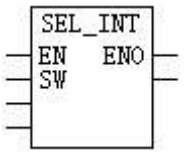
简介

该模块的功能是当 SW=OFF 时，将输入值 IN1 赋给输出值；当 SW=ON 时，将输入值 IN2 赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = SEL_INT(SW,IN1,IN2)

参数描述

参数	数据类型	含义
SW	BOOL	输入选择开关
IN1	INT	第一输入
IN2	INT	第二输入
OUT	INT	输出

SEL_LONG 模块

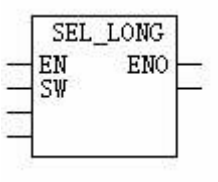
简介

该模块的功能是当 SW=OFF 时，将输入值 IN1 赋给输出值；当 SW=ON 时，将输入值 IN2 赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = SEL_LONG(SW,IN1,IN2)

参数描述

参数	数据类型	含义
SW	BOOL	输入选择开关
IN1	LONG	第一输入
IN2	LONG	第二输入
OUT	LONG	输出

SEL_SFLOAT 模块

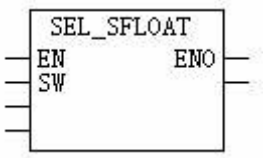
简介

该模块的功能是当 SW=OFF 时，将输入值 IN1 赋给输出值；当 SW=ON 时，将输入值 IN2 赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = SEL_SFLOAT(SW,IN1,IN2)

参数描述

参数	数据类型	含义
SW	BOOL	输入选择开关
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
OUT	SFLOAT	输出

SEL_UINT 模块

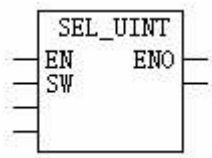
简介

该模块的功能是当 SW=OFF 时，将输入值 IN1 赋给输出值；当 SW=ON 时，将输入值 IN2 赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = SEL_UINT(SW,IN1,IN2)

参数描述

参数	数据类型	含义
SW	BOOL	输入选择开关
IN1	UINT	第一输入
IN2	UINT	第二输入
OUT	UINT	输出

SEL_ULONG 模块

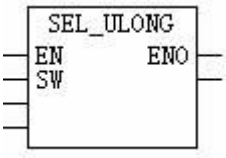
简介

该模块的功能是当 SW=OFF 时，将输入值 IN1 赋给输出值；当 SW=ON 时，将输入值 IN2 赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = SEL_ULONG(SW,IN1,IN2)

参数描述

参数	数据类型	含义
SW	BOOL	输入选择开关
IN1	ULONG	第一输入
IN2	ULONG	第二输入
OUT	ULONG	输出

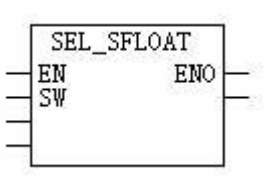
SEL_WORD 模块

简介

该模块的功能是当 SW=OFF 时，将输入值 IN1 赋给输出值；当 SW=ON 时，将输入值 IN2 赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示



公式

OUT = SEL_WORD(SW,IN1,IN2)

参数描述

参 数	数据类型	含义
SW	BOOL	输入选择开关
IN1	WORD	第一输入
IN2	WORD	第二输入
OUT	WORD	输出

4) 最大值模块

MAX_FLOAT 模块

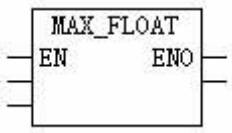
简介

该模块的功能是将输入值中的最大值赋给输出值；输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = MAX_INT(IN1,IN2)

参数描述

参数	数据类型	含义
IN1	FLOAT 型	第一输入
IN2	FLOAT 型	第二输入
INn	FLOAT 型	第 n 个输入
OUT	FLOAT 型	输出

MAX_INT 模块

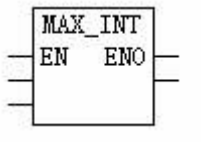
简介

该模块的功能是将输入值中的最大值赋给输出值；输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = MAX_INT(IN1,IN2)

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
INn	INT	第 n 个输入
OUT	INT	输出

MAX_LONG 模块

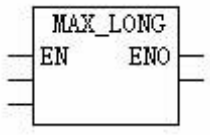
简介

该模块的功能是将输入值中的最大值赋给输出值；输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = MAX_LONG(IN1,IN2)

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
INn	LONG	第 n 个输入
OUT	LONG	输出

MAX_UINT 模块

简介

该模块的功能是将输入值中的最大值赋给输出值；输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT = MAX_UINT(IN1,IN2)

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
INn	UINT	第 n 个输入
OUT	UINT	输出

MAX_SFLOAT 模块

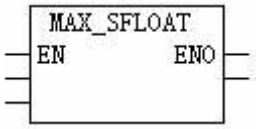
简介

该模块的功能是将输入值中的最大值赋给输出值；输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = MAX_SFLOAT(IN1,IN2)

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
INn	SFLOAT	第 n 个输入
OUT	SFLOAT	输出

MAX_ULONG 模块

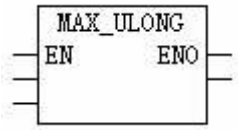
简介

该模块的功能是将输入值中的最大值赋给输出值；输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = MAX_ULONG(IN1,IN2)

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
INn	ULONG	第 n 个输入

OUT	ULONG	输出

5) 最小值模块

MIN_FLOAT 模块

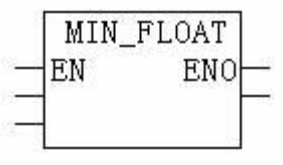
简介

该模块的功能是将输入值中的最小值赋给输出值；输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = MIN_FLOSAT(IN1,IN2)

参数描述

参数	数据类型	含义
IN1	FLOAT 型	第一输入
IN2	FLOAT 型	第二输入
INn	FLOAT 型	第 n 个输入
OUT	FLOAT 型	输出

MIN_INT 模块

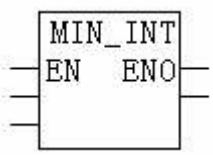
简介

该模块的功能是将输入值中的最小值赋给输出值；输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = MIN_INT(IN1,IN2)

参数描述

参数	数据类型	含义
IN1	INT	第一输入
IN2	INT	第二输入
INn	INT	第 n 个输入
OUT	INT	输出

MIN_LONG 模块

简介

该模块的功能是将输入值中的最小值赋给输出值；输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT = MIN_LONG(IN1,IN2)

参数描述

参数	数据类型	含义
IN1	LONG	第一输入
IN2	LONG	第二输入
INn	LONG	第 n 个输入
OUT	LONG	输出

MIN_SFLOAT 模块

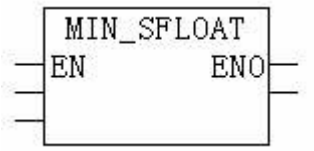
简介

该模块的功能是将输入值中的最小值赋给输出值；输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = MIN_SFLOAT(IN1,IN2)

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
INn	SFLOAT	第 n 个输入
OUT	SFLOAT	输出

MIN_UINT 模块

简介

该模块的功能是将输入值中的最小值赋给输出值；输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号

公式

OUT = MIN_UINT(IN1,IN2)

参数描述

参数	数据类型	含义
IN1	UINT	第一输入
IN2	UINT	第二输入
INn	UINT	第 n 个输入
OUT	UINT	输出

MIN_ULONG 模块

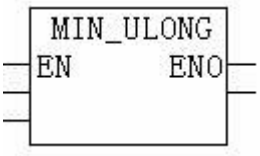
简介

该模块的功能是将输入值中的最小值赋给输出值；输入值个数不限。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



公式

OUT = MIN_ULONG(IN1,IN2)

参数描述

参数	数据类型	含义
IN1	ULONG	第一输入
IN2	ULONG	第二输入
INn	ULONG	第 n 个输入
OUT	ULONG	输出

1.5.2 辅助模块库

1. 控制模块

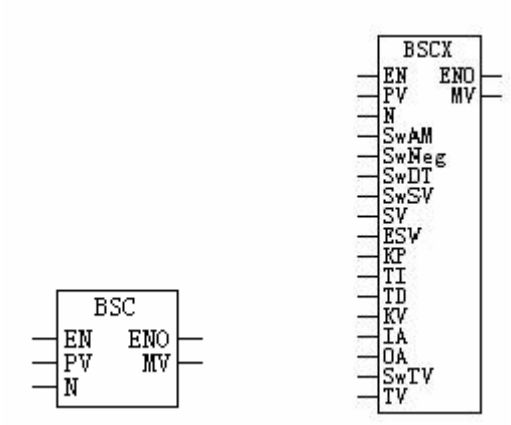
单回路模块

简介

该模块是对在自定义回路中声明的单回路进行定义，确定它的输入输出，组成一个控制回路。所要控制的对象做为系统的输入(PV)，回路的输出（MV）到能够改变控制对象值的执行机构上。将它在自定义回路中所对应的位号组入监控画面中，可在监控画面中对其进行参数设置。其中 BSCX 可以有更多的参数让用户来设置。

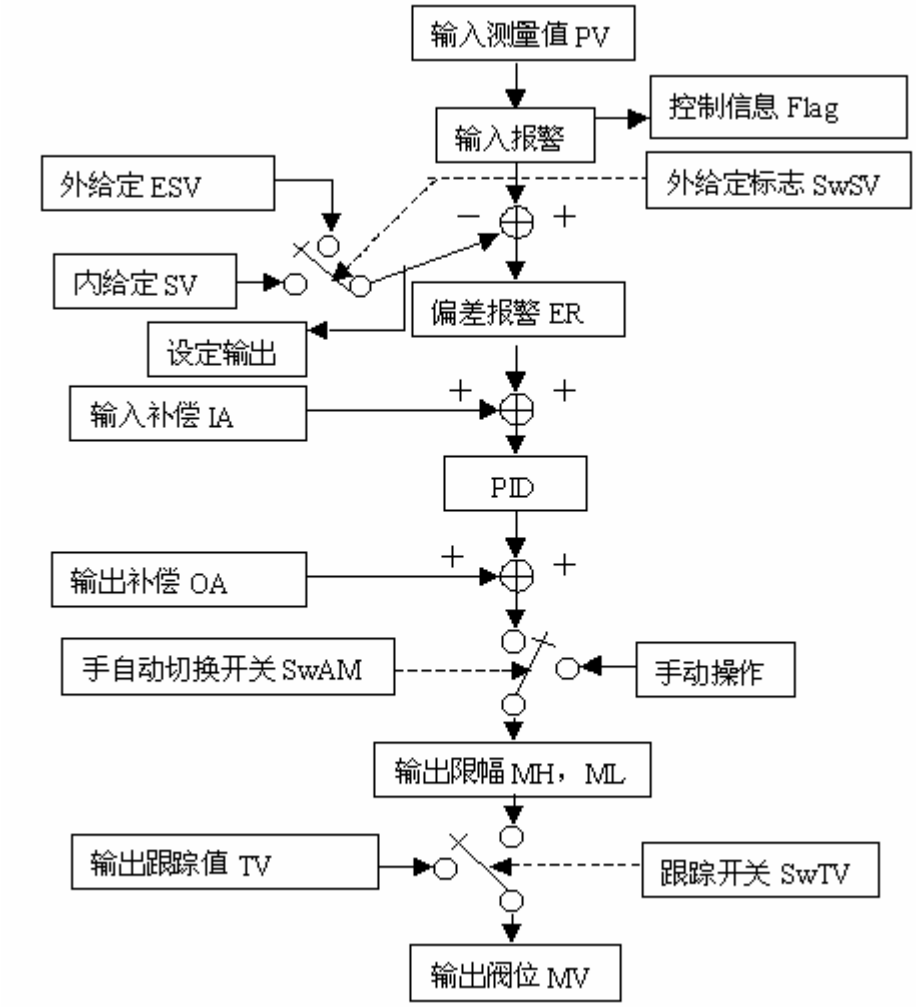
表示

符号



算法

该模块是 PID 单回路控制模块，流程图如下所示：



参数描述

参数	数据类型	含义	补充说明
PV	SFLOAT	测量值	

N	UINT	BSC 序号	范围[0,31]
MV	SFLOAT	输出阀位	
g_bsc[N].SwSV	BOOL	内/外给定开关	ON—外给定
g_bsc[N].SwAM	BOOL	手/自动开关	ON—自动
g_bsc[N].SwNeg	BOOL	正/反作用开关	ON—反作用
g_bsc[N].SwTV	BOOL	输出跟踪开关	ON—跟踪
g_bsc[N].SwDT	BOOL	微分方式切换开关	ON=dPV/dt OFF=dErr/dt
g_bsc[N].SV	SFLOAT	内给定值	
g_bsc[N].ESV	SFLOAT	外给定值	
g_bsc[N].KP	SFLOAT	比例常数	Kp*2 = 1/P
g_bsc[N].TI	INT	积分时间	单位为 0.1 秒
g_bsc[N].TD	INT	微分时间	单位为 0.1 秒
g_bsc[N].TV	SFLOAT	输出跟踪量	
g_bsc[N].IA	SFLOAT	输入补偿	
g_bsc[N].OA	SFLOAT	输出补偿	
g_bsc[N].ER	SFLOAT	偏差报警值	
g_bsc[N].ML	SFLOAT	输出限幅下限	
g_bsc[N].MH	SFLOAT	输出限幅上限	
g_bsc[N].KV	SFLOAT	可变增益	
g_bsc[N].RESERVED_6B	BOOL	比例微分先行开关	ON＝比例微分先行

修改说明

本次的修改时将单回路添加一个比例微分先行功能，通过选择开关 g_bsc[N].RESERVED_6B 来选择是否采用比例微分先行。PV 比例和微分类型 PID 控制算法（I－PD）和基本类型 PID 算法不同。当设定值改变时，它仅仅执行积分作用。当设定值（SV）是通过数字值传入时，即使设定值剧烈变化，这种算法保证控制特性的稳定。与此同时，如果在控制过程中发生特性变化，这种算法还保证了有适当的控制来响应这些变化；通过执行相应的比例、微分和积分控制作用来承担变化和扰动。

注意：

对 BSCX 模块，需要特别注意所设置的参数不能与 AdvanTrol 等监控画面中的相关参数相冲突，否则将导致 AdvanTrol 中监控画面中的相关参数设置功能无效！

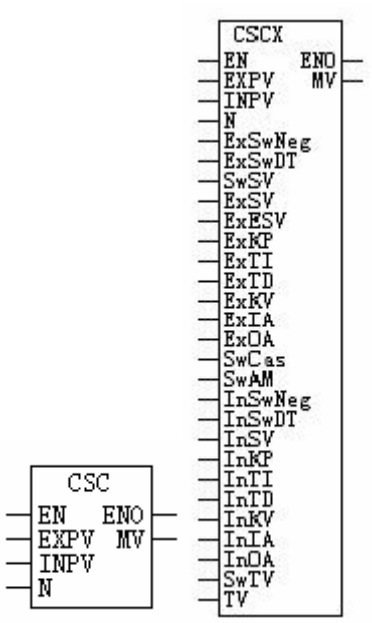
串级控制模块

简介

该模块是对自定义回路中的双回路进行设置，使其组成一个串级控制回路。将它在自定义回路中所对应的位号组入监控画面中，可在监控画面中对其进行参数设置。CSCX 是 CSC 模块的扩展，它开放了更多的参数给用户。

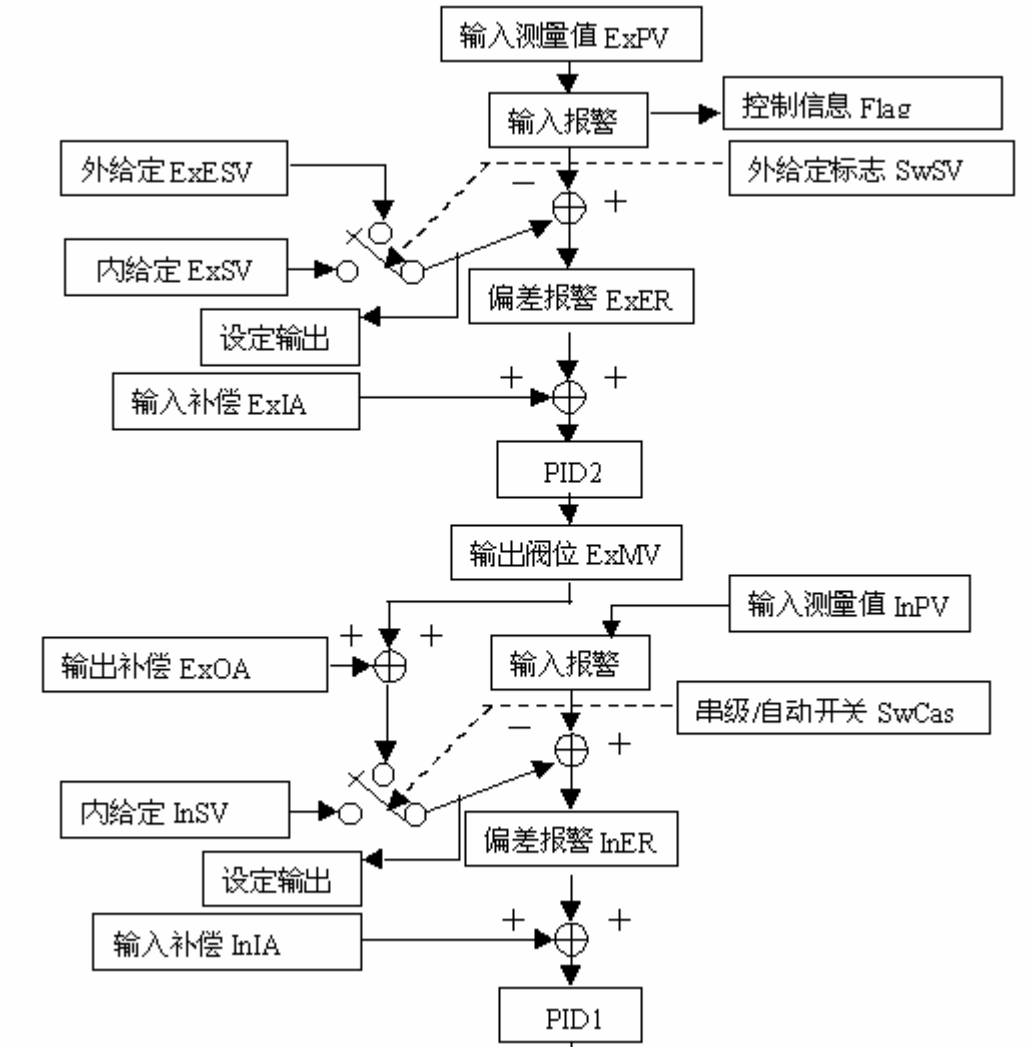
表示

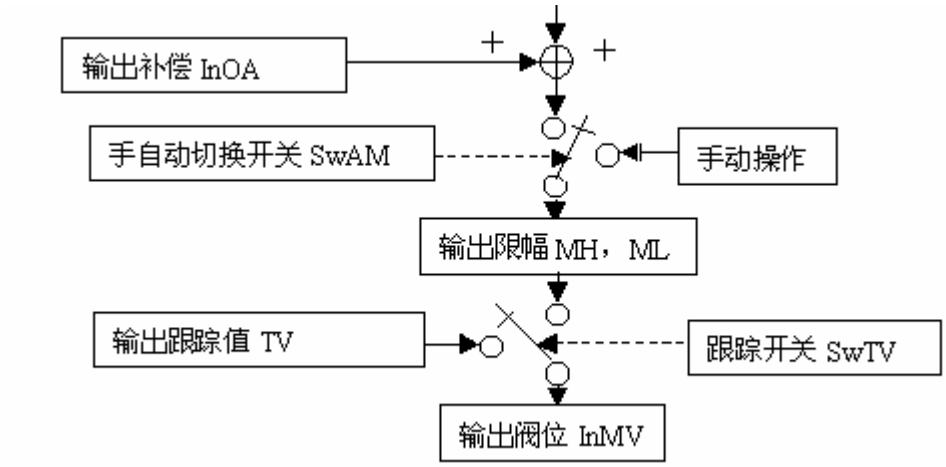
符号



算法

算法流程图如下所示：





参数描述

参数	数据类型	含义	说明
ExPV	SFLOAT	外环反馈信号	
InPV	SFLOAT	内环反馈信号	
N	UINT	CSC 序号	0~31
g_csc[N].SwCas	BOOL	串级/单回路切换开关	ON—串级
g_csc[N].SwSV	BOOL	内/外给定开关	ON—外给定
g_csc[N].SwAM	BOOL	手/自动开关	ON—自动
g_csc[N].InSwNeg	BOOL	内环正/反作用开关	ON—反作用
g_csc[N].ExSwNeg	BOOL	外环正/反作用开关	ON—反作用
g_csc[N].SwTV	BOOL	输出跟踪开关	ON—跟踪
g_csc[N].InSwDT	BOOL	内环微分方式开关	ON=dPV/dt OFF=dErr/dt
g_csc[N].ExSwDT	BOOL	外环微分方式开关	ON=dPV/dt OFF=dErr/dt
g_csc[N].InMV	SFLOAT	内环控制量	InMV = 最终输出
g_csc[N].ExMV	SFLOAT	外环控制量	ExMV = InSV - ExOA
g_csc[N].InSV	SFLOAT	内环内给定值	
g_csc[N].ExSV	SFLOAT	外环内给定值	
g_csc[N].ExESV	SFLOAT	外环外给定值	
g_csc[N].InKP	SFLOAT	内环比例常数	Kp*2 = 1/P
g_csc[N].ExKP	SFLOAT	外环比例常数	Kp*2 = 1/P
g_csc[N].InTI	INT	内环积分时间	单位为 0.1 秒
g_csc[N].ExTI	INT	外环积分时间	单位为 0.1 秒
g_csc[N].InTD	INT	内环微分时间	单位为 0.1 秒
g_csc[N].ExTD	INT	外环微分时间	单位为 0.1 秒
g_csc[N].TV	SFLOAT	输出跟踪量	
g_csc[N].InIA	SFLOAT	内环输入补偿	
g_csc[N].ExIA	SFLOAT	外环输入补偿	
g_csc[N].InOA	SFLOAT	内环输出补偿	
g_csc[N].ExOA	SFLOAT	外环输出补偿	
g_csc[N].InER	SFLOAT	内环偏差报警值	
g_csc[N].ExER	SFLOAT	外环偏差报警值	
g_csc[N].ML	SFLOAT	输出限幅下限	限幅在手自动后跟踪前

g_csc[N].MH	SFLOAT	输出限幅上限	
g_csc[N].InKV	SFLOAT	内环可变增益	
g_csc[N].ExKV	SFLOAT	外环可变增益	

注意：

对 CSCX 模块，需要特别注意所设置的参数不能与 AdvanTrol 等监控画面中的相关参数相冲突，否则将导致 AdvanTrol 中监控画面中的相关参数设置功能无效！

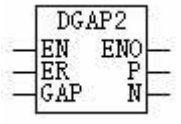
二位式二状态控制模块

简介

该模块是一种二位式差隙调节器，用于二状态控制应用场合，它的功能输出相当于具有滞环的继电器特性。

表示

符号



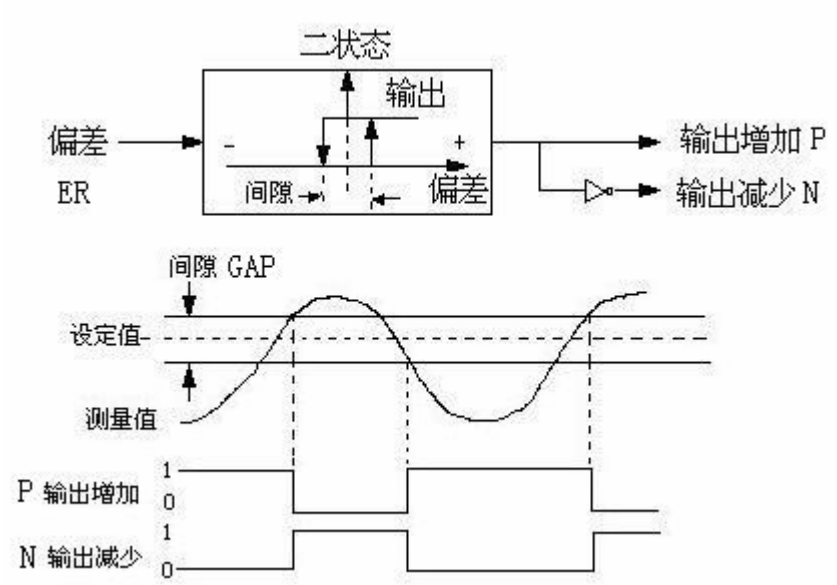
算法

当偏差 ER >= GAP/2，正向输出 P=ON，反向输出 N=OFF；

当偏差 ER <= -GAP/2，正向输出 P=OFF，反向输出 N=ON；

当偏差 ER 处于（-GAP/2,GAP/2）区间内，正向输出和反向输出值都和上次一样。

用图形表示就是：



参数描述

参数	数据类型	含义
ER	SFLOAT	输入偏差值
GAP	SFLOAT	给定间隙值
P	BOOL	正向输出
N	BOOL	反向输出

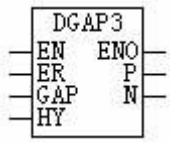
二位式三状态控制模块

简介

该模块也是一种二位式差隙调节器，用于三状态控制应用场合，它的功能相当于具有死区和滞环的继电特性。

表示

符号



算法

对于输出 P：

当偏差 ER >= GAP/2，输出 P＝ON；

当偏差 ER <=GAP/2－HY，输出 P＝OFF；

当偏差 ER 在区间（GAP/2－HY，GAP/2）内时，输出 P 保持前一次的输出值；

对于输出 N：

当偏差 ER<=－GAP/2，输出 N=ON；

当偏差 ER>=－GAP/2＋HY，输出 N＝OFF；

当偏差 ER 在区间内（－GAP/2，－GAP/2+HY），输出 N 保持上一次的输出值；

这样一来，总的逻辑就是：

当偏差 ER 是递增时：

当偏差 ER<= －GAP/2＋HY，输出 P=OFF，N=ON；

当偏差 ER 在区间（－GAP/2＋HY，GAP/2）内时，输出 P=OFF，N=OFF；

当偏差 ER>=GAP/2；输出 P=ON，N=OFF；

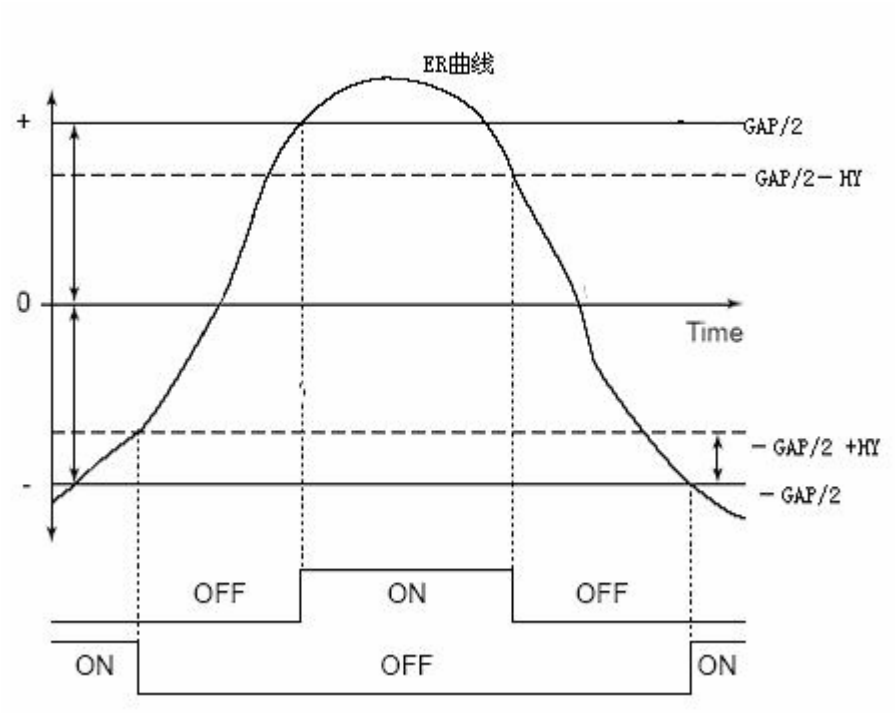
当偏差 ER 是递减时：

当偏差 ER>=GAP/2－HY，输出 P=ON，输出 N=OFF；

当偏差 ER 在区间（GAP/2－HY，－GAP/2）内时，输出 P=OFF，输出 N=OFF；

当偏差 ER<=－GAP/2；输出 P=OFF，输出 N=ON；

如图所示：



参数描述

参数	数据类型	含义
ER	SFLOAT	偏差（PV－SV）
GAP	SFLOAT	间隙
HY	SFLOAT	间隙死区
P	BOOL	正向输出
N	BOOL	反向输出

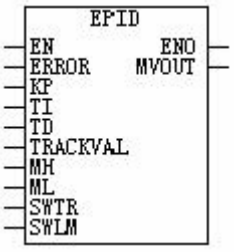
误差 PID 运算模块

简介

该模块输入一个误差信号，根据这个误差信号得到一个控制输出值。

表示

符号



算法

MVOUT = KP*(ERROR-ERROR_LASTTIME ＋ ERROR*TS/TI +UDn)

UDn＝(UDn-1 + 8*(ERROR-ERROR_LASTTIME))* TD / (8*TS + TD)

其中 MVOUT 为控制输出值，KP 为比例系数，ERROR 为输入误差，ERROR_LASTTIME 为上一控制周期的误差，TS 为控制周期，TI 为积分时间，UDn 为微分项输出值，UDn-1 为上一控制周期微分项输出值，TD 为微分时间。

参数描述

参数	数据类型	含义	说明
ERROR	FLOAT	输入误差	
KP	FLOAT	比例系数	
TI	FLOAT	积分时间	单位为秒
TD	FLOAT	微分时间	单位为秒
TRACKVAL	FLOAT	输出跟踪值	
MH	FLOAT	输出上限	
ML	FLOAT	输出下限	
SWTR	BOOL	跟踪开关	但 SWTR＝ON 时控制输出值跟踪 TRACKVAL 变化而变化，当为 OFF 时控制输出值等于运算输出值
SWLM	BOOL	输出限幅开关	当 SWLM＝ON 时，对控制输出值进行限幅
MVOUT	FLOAT	控制输出值	

闪光模块

简介

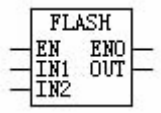
该模块用来产生一个脉冲的输出信号，当输入 IN1=OFF，则输出为 OUT=OFF；

当输入 IN1=ON 且 IN2=OFF，则输出跟踪 IN1 状态，即 OUT=ON；

当输入 IN1=IN2=ON 时，输出将交替为逻辑 0 和逻辑 1，交替周期为扫描周期。

表示

符号



算法

当 IN1=OFF，输出 OUT=OFF；

当 IN1=ON0 且 IN2=OFF，输出 OUT=ON；

当 IN1=IN2=ON，输出 OUT 交替为 ON 或 OFF，交替周期为扫描周期，即当程序第 2 次运行到该模块，输出就发生跳变。

参数描述

参数	数据类型	含义
IN1	BOOL	报警信号
IN2	BOOL	确认报警信号
OUT	BOOL	输出

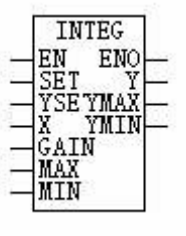
积分（限幅）模块

简介

输出是对输入的积分，并且对输出进行限幅处理，当输出值超过设定的高低限，就会被限幅，同时置相应的报警标志位。

表示

符号



算法

当 SET＝ON 时，输出 Y＝YSET，如果 Y>=MAX，YMAX=ON，否则 YMAX=OFF；如果 Y<=MIN，YMIN＝ON，否则 YMIN＝OFF。

当 SET=OFF 时，输出 Y 的值是对输入 X 积分，采样时间以 0.1ms 为单位；其中传递函数和离散化算式与积分不限幅模块相同。

如果 Y>＝MAX，则 Y=MAX，且 YMAX=ON，否则 Y 输出对输入 X 积分运算的值，YMAX＝OFF。

如果 Y<=MIN，则 Y＝MIN，且 YMIN＝ON，否则 Y 输出对输入 X 积分运算的值，YMIN＝OFF。

参数描述

参数	数据类型	含义
SET	BOOL	置位开关
YSET	SFLOAT	设定值
X	SFLOAT	输入值
GAIN	SFLOAT	增益
MAX	SFLOAT	高限
MIN	SFLOAT	低限
YMAX	BOOL	高限报警开关
YMIN	BOOL	低限报警开关
Y	SFLOAT	输出值

积分（不限幅）模块

简介

输出是对输入的积分，但并未对输出进行限幅处理，只是在输出超出限幅值时，置报警位。

表示

符号

算法

当 SET＝ON 时，输出 Y＝YSET；

当 SET=OFF 时，输出 Y 的值是对输入 X 积分，采样时间以 0.1ms 为单位：

传递函数：

$$G(S)=\frac{Y(S)}{X(S)}=\frac{GAIN}{S}$$

转化为离散化的计算公式：

$$S\times Y(S)=GAIN\times X(S)$$

$$Y(n)=Y(n-1)+GAIN\times dt\times X(n)$$

这里取输入的平均值做积分计算，上面公式变为：

$$Y(n)=Y(n-1)+GAIN\times dt\times\frac{X(n)+X(n-1)}{2}$$

以上两种情况下都对输出值进行超限判断，当输出 Y>=MAX，YMAX＝ON，否则 YMAX＝OFF；当输出 Y<=MIN，YMIN＝ON，否则 YMIN＝OFF。

参数描述

参数	数据类型	含义
SET	BOOL	置位开关
YSET	SFLOAT	设定值
X	SFLOAT	输入值
GAIN	SFLOAT	增益
MAX	SFLOAT	高限
MIN	SFLOAT	低限
YMAX	BOOL	高限报警开关
YMIN	BOOL	低限报警开关
Y	SFLOAT	输出值

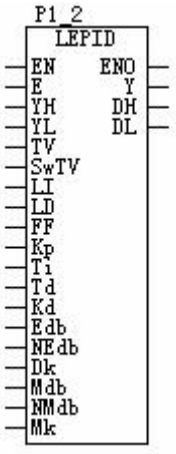
闭锁型偏差 PID 模块（LEPID）

简介

该模块提供了闭锁增减功能、抗积分饱和功能、输出跟踪功能、输出限幅功能，它的输入为设定值和测量值的偏差。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

在该模块中采用的 PID 算法公式的传递函数表达式如下：

$$Y(S)=(K_p+\frac{1}{T_i*S}+\frac{Kd*Td*S}{Td*S+1})E(s)+FF(s)$$

该模块采用增量式算法，离散化后的公式为：

$$Y(n)=K_p*e(n)+\frac{T_s}{T_i}*e(n)+Y(n-1)+\frac{Td}{Td+Ts}*Y(n-1)+\frac{Kd*Td}{Td+Ts}*[e(n)-e(n-1)]+FF(n)$$

对应的比例项的增量为： $\Delta Y_p=Y_p(n)-Y_p(n-1)=K_p*\Delta e$

对应的积分项的增量为： $\Delta Y_I=Y_I(n)-Y_I(n-1)=\frac{Ts}{Ti}*\Delta e$

对应的微分项的增量为： $\Delta Y_D=Y_D(n)-Y_D(n-1)$

所以第 n 个周期的总的 PID 计算增量值为： $\Delta Y=\Delta Y_p+\Delta Y_I+\Delta Y_D$

其中：第 n 个周期的微分项计算值 $Y_D(n)=\frac{Td}{Td+Ts}Y_D(n-1)+\frac{Kd*Td}{Td+Ts}*\Delta e$

第 n 个周期的偏差增量 $\Delta e=e(n)-e(n-1)$

参数描述

参数	数据类型	含义
E	FLOAT	设定值和测量值的偏差输入
YH	FLOAT	输出的上限
YL	FLOAT	输出的下限
TV	FLOAT	输出跟踪值
SwTV	BOOL	输出跟踪开关，当 SwTV＝ON 时，输出 Y＝TV。
LI	BOOL	闭锁增开关，当 LI＝ON 时，输出 Y 停止增加，但可以减小。
LD	BOOL	闭锁减开关，当 LD＝ON 时，输出 Y 停止减小，但可以增加。
FF	FLOAT	前馈变量
Kp	FLOAT	比例放大系数，Kp＝0.0 时，无比例项
Ti	FLOAT	积分时间常数（0.1s 为单位），Ti<=0.0 时，无积分项
Td	FLOAT	微分时间常数（0.1s 为单位），Td<=0.0 时，无微分项
Kd	FLOAT	微分器放大系数
Edb	FLOAT	积分器停止积分时的偏差值，当 E>Edb>0 时，积分停止
DK	FLOAT	积分器停止积分时 Kp 的修正值，积分停止后：Kp＝原 Kp＋Dk
Mdb	FLOAT	精控区范围 0<|E|<Mdb
Mk	FLOAT	精控区系数 1>=Mk>0
Y	FLOAT	PID 运算输出值
DH	FLOAT	PID 输出越上限标志，当 Y>YH 时，DH=ON。
DL	FLOAT	PID 输出越下限标志，当 Y<YL 时，DL=ON。

限定或模块

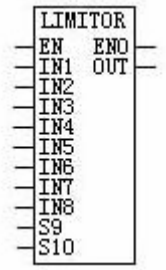
简介

该模块用来监视 8 个数字量输入，并根据 S9 和 S10 设置的状态和 8 个输入中为逻辑 1 的输入的个数来产生一个 BOOL 量输出：当逻辑 1 的输入个数等于 S9 中设定的数量且 S10 不等于 0，输出为 OUT 为 ON；当逻辑 1 的输入个数等于或大于 S9 中设定的数量且 S10 等

于 0，输出 OUT 为 ON；当逻辑 1 的输入个数小于 S9 中设定的数量，输出 OUT 为 OFF。

表示

符号



算法

先统计 8 输入量中为逻辑 1 的输入个数，记为 b；

当 b = S9 且 S10 != 0，输出 OUT=ON；

当 b>= S9 且 S10 = 0，输出 OUT=ON；

其余的情况，输出 OUT=OFF。

参数描述

参数	数据类型	含义
IN1、IN2、…、IN8	INT	8 个输入
S9	INT	选择参数
S10	INT	工作方式参数
OUT	BOOL	输出

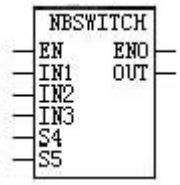
无扰动切换模块

简介

根据输入 IN3 的逻辑（ON 或 OFF）来选择两个输入 IN1、IN2 中的一个，S4 和 S5 是两个无扰动切换的限定参数，当输入 IN3 的逻辑发生变化时，使切换按照相应的限定参数平滑过渡。当 IN3=OFF，输出平衡时 OUT=IN1；当 IN3=ON，输出平衡时 OUT=IN2。

表示

符号



算法

1、当 IN3 从 OFF 切换到 ON 时，也就是输出 OUT 从 IN1 切换到 IN2，记上一次的平滑处理输出值为 LY，则有以下过渡情况：

当 LY>IN2>IN1 或者 LY<IN2<IN1 时，输出 OUT=LY－S5*(IN2－IN1)

当 LY>IN2 且 IN2<IN1 或者 LY<IN2 且 IN2>IN1 时，输出 OUT=LY+S5*(IN2－IN1)

判断是否稳定，即当 IN2>LY 且 OUT>IN2 或者 IN2<LY 且 OUT<IN2 或者 LY=IN2；则输出 OUT=IN2；

2、当 IN3 从 ON 切换到 OFF 时，也就是输出 OUT 从 IN2 切换到 IN1 时，记上一次的

平滑处理输出值为 LY，则同样有以下过渡情况：

当 LY>IN1 且 IN2>IN1 或者 LY<IN1 且 IN1>IN2，则输出 OUT=LY+S4*(IN1－IN2)

当 LY<IN1<IN2 或者 LY>IN1>IN2，则输出 OUT=LY－S4*(IN1－IN2)

判断是否稳定，即当 IN1>LY 且 OUT>IN1 或者 IN1<LY 且 OUT<IN1 或者 LY=IN1，则输出 OUT=IN1。

参数描述

参数	数据类型	含义
IN1	SFLOAT	第一输入
IN2	SFLOAT	第二输入
IN3	BOOL	输入选择开关
S4	SFLOAT	输出从 IN2 切换到 IN1 的变化率
S5	SFLOAT	输出从 IN1 切换到 IN2 的变化率
OUT	SFLOAT	输出

斜坡信号发生器模块（RAMP_GNT）

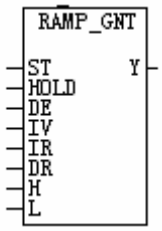
简介

该模块用来产生斜坡信号。

注意：输出下限不要大于输出上限，否则，只有当启动开关 ST 从 OFF 跳变到 ON 时，输出 Y=IV，其他情况下该模块不作任何作用便返回。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

当启动开关 ST 从 OFF 跳变为 ON 时，置输出 Y=IV。

当 ST 保持为 ON 时：

(1) 当 HOLD=ON，输出 Y 保持上个周期的值。

(2) 当 HOLD=OFF：若 DE=ON，则输出 Y 呈递增的斜坡特性，

。

若 DE=OFF，则输出 Y 呈递减的斜坡特性，　　。

其中：LY 是上个周期的输出 Y 的值，Ts 是系统的控制周期，以秒为单位。

无论是置初始值还是产生斜坡信号输出，都对输出进行限幅，若输出大于输出上限，则输出 Y=H，若输出小于输出下限，则输出 Y=L。

参数描述

参数	数据类型	含义
ST	BOOL	启动开关
HOLD	BOOL	保持开关

DE	BOOL	升降选择开关
IV	SFLOAT	初始值
IR	SFLOAT	上升速率
DR	SFLOAT	下降速率
H	SFLOAT	输出上限
L	SFLOAT	输出下限
Y	SFLOAT	输出值

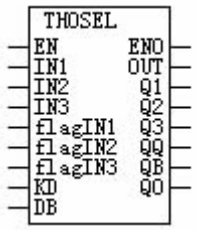
三选中模块

简介

根据工作方式参数 MD 以及两个输入的输入品质 flagIN1、flagIN2（质量码）来选择输出方式以及输出的品质。

表示

符号



算法

当工作方式参数 MD 等于 4 或 5 或 6，则输出 OUT 的值相应的等于 IN1 或 IN2 或 IN3，如果对应的输入点为坏点，则输出保持上一个周期的值不变，品质输出 QO 置为逻辑 1。

当工作方式参数 MD 等于 0、1、2、3 时，则有

- MD 等于 0 时，选择三个输入中数值居中的作为输出 OUT 的值。
- MD 等于 1 时，选择三个输入的平均值作为输出 OUT 的值。
- MD 等于 2 时，选择三个输入中数值最小的作为输出 OUT 的值。
- MD 等于 3 时，选择三个输入中数值最大的作为输出 OUT 的值。

此时，还要根据相应的输入品质来调整输出：

- 如果三个输入都是坏点，则输出保持上一个周期的值，品质输出 QO 置为逻辑 1。
- 如果有两个输入是坏点，则输出剩余的一个好点的值。
- 如果只有一个输入是坏点，则根据另外两个输入好点之间的偏差来选择输出。若两个好点之间的偏差不超过指定的偏差上限值 DB，则输出两个好点的平均值，否则输出保持上一个周期的值，品质输出 QO 置为逻辑 1。
- 如果三个输入都是好点，则根据三点之间的三对偏差是否超过 DB 作如下的选择：

i. 三对偏差都不超过 DB 时，则输出根据 MD 所选的方式输出。

ii. 只有一个偏差超过 DB 时，则选择三个输入中数值居中的值作为输出 OUT 的值。

iii. 若有两对偏差超过 DB 时，则取偏差不超限的一对输入值的平均值作为输出 OUT 的值。

iv. 若三对偏差都超过 DB 时，则输出保持上一个周期的值，品质输出 QO 置为逻辑 1。

品质输出

如果 IN1 是坏点，则相应的输出品质 Q1 置为逻辑 1；否则输出品质 Q1 置为逻辑 0。

如果 IN2 是坏点，则相应的输出品质 Q2 置为逻辑 1；否则输出品质 Q1 置为逻辑 0。

如果 IN3 是坏点，则相应的输出品质 Q3 置为逻辑 1；否则输出品质 Q1 置为逻辑 0。

IN1、IN2、IN3 中任一个为坏点时，则品质输出 QQ 为逻辑 1；反之为逻辑 0。

如果工作方式参数 MD 等于 4 或 5 或 6，且相应的 IN1 或 IN2 或 IN3 为坏点时，品质输出 QO 置为逻辑 1。

如果工作方式参数 MD 等于 4 或 5 或 6，且三个输入点都是好点时，若三点之间的三对偏差均超过 DB 时，则品质输出 QO 为逻辑 1。

如果三点之间的偏差有两对超过 DB 时，则品质输出 QB 置为逻辑 1，反之为逻辑 0。

参数描述

参数	数据类型	含义
IN1、IN2、IN3	SFLOAT 型	输入
flagIN1、flagIN2、flagIN3	INT 型	输入数据的品质
MD	INT 型	选择输出方式
DB	SFLOAT 型	偏差上限值
OUT	SFLOAT 型	输出值
Q1、Q2、Q3	BOOL 型	相应输入量的品质输出
QQ	BOOL 型	品质总报警
QB	BOOL 型	偏差报警
QO	BOOL 型	品质输出

应用

主要用于需要在三个输入中选择适当的值作为输出，选择的条件包括：输入的品质（质量码）以及输入之间的偏差。选择三个输入中的最优值作为控制参数，适用于对输入精度要求高并且对随机扰动敏感的控制过程中。

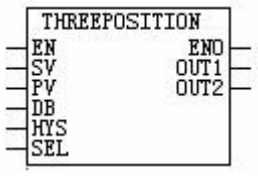
三位开关控制模块

简介

三位开/关控制模块通过两个接触输出来执行开/关动作，控制器通过比较过程变量（PV）和设定值（SV），有三种模式打开或者关闭接触输出（两个都关闭设置为中间位置）。

表示

符号



VOID THREEPOSITION（SV,PV,DB,HYS,SEL,OUT1,OUT2）

算法

当正负向选择开关 SEL=ON 时，为正向控制，根据输入值和设定值的偏差（E=PV-SV）所处的位置，执行不同的输出。具体如下：

偏差 E 增加时的操作输出状态：

（1）当 E<(-DB)时，输出 OUT1=OFF，OUT2=ON。

（2）当-(DB-HYS)≤E<DB 时，OUT1= OFF，OUT2=OFF。

（3）当时，OUT1=ON，OUT2=OFF。

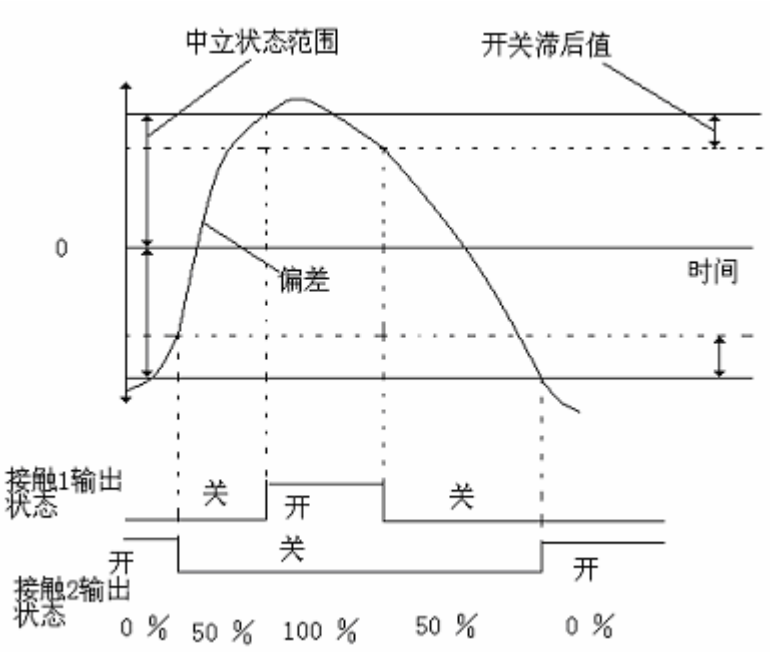
偏差 E 减小时的操作输出状态：

（1）当 E≥时，OUT1=ON，OUT2=OFF。

（2）当-DB<E≤DB-HYS，OUT1=OFF，OUT2=OFF。

（3）当 En≤-DB 时，OUT1=OFF，OUT2=ON。

原理图如下：



当正负向选择开关 SEL=OFF 时，为反向控制，根据输入值和设定值的偏差（E=PV-SV）所处的位置，执行不同的输出。具体如下：

偏差 E 增加时的操作输出状态：

（1）当 E<(-DB)时，输出 OUT1=ON，OUT2=OFF。

（2）当-(DB-HYS)≤E<DB 时，OUT1= OFF，OUT2=OFF。

（3）当时，OUT1=OFF，OUT2=ON。

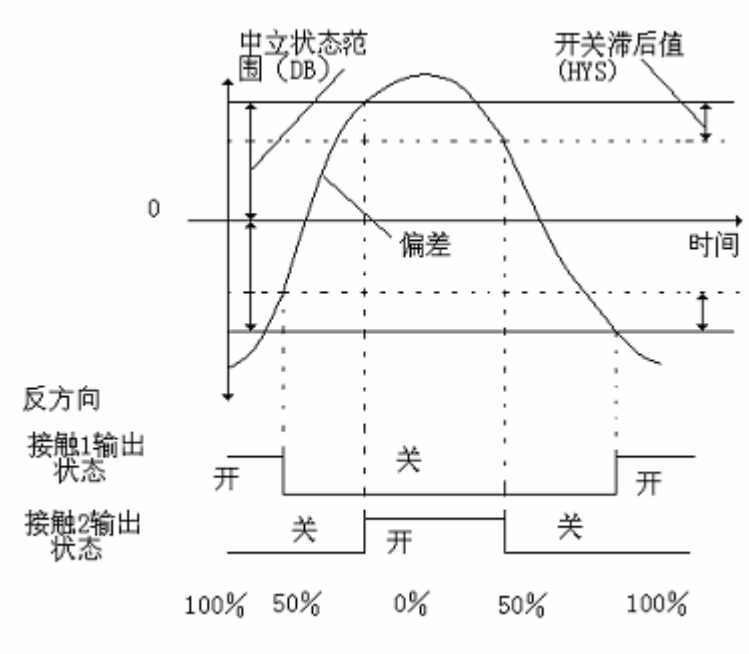
偏差 E 减小时的操作输出状态：

（1）当 E≥时，OUT1=OFF，OUT2=ON。

（2）当-DB<E≤DB-HYS 时，OUT1=OFF，OUT2=OFF。

（3）当 En≤-DB 时，OUT1=ON，OUT2=OFF。

原理图如下：



参数描述

参数	数据类型	含义
SV	SFLOAT	设定值
PV	SFLOAT	输入值
HYS	SFLOAT	开关滞后值
DB	SFLOAT	中间范围
SEL	BOOL	正负向选择开关
CPV	BOOL	操作输出值
OUT1	BOOL	接触输出开关 1
OUT2	BOOL	接触输出开关 2

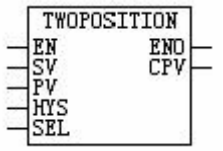
二位开关控制模块

简介

比较过程变量（PV）和设定值（SV），根据方向选择接触输出的开关特性，该模块通过一个接触输出的信号来执行开/关控制动作。

表示

符号



VOID TWOPOSITION (SV,PV,HYS,SEL,CPV)

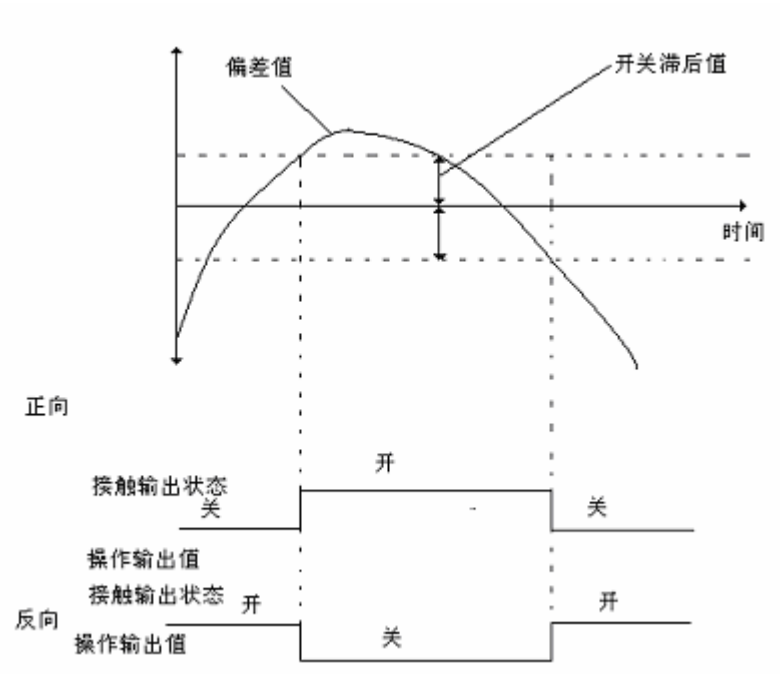
算法

当正负向选择开关 SEL=ON 时，为正向控制，如果输入值 PV 和设定值 SV 的差超过正的开关滞后值（HYS）时，输出值 CPV 为 ON；如果输入值 PV 和设定值 SV 的差小于负的开/关滞后值时，输出值 CPV 为 OFF。

当正负向选择开关 SEL=OFF 时，为反向控制，如果输入值 PV 和设定值 SV 的差超过

正的开关滞后值（HYS）时，输出值 CPV 为 OFF；如果输入值 PV 和设定值 SV 的差小于负的开/关滞后值时，输出值 CPV 为 ON。

原理图如下所示：



参数描述

参数	数据类型	含义
SV	SFLOAT	设定值
PV	SFLOAT	输入值
HYS	SFLOAT	开关滞后值
SEL	BOOL	正负向选择开关
CPV	BOOL	操作输出值

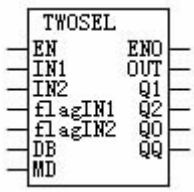
二选一模块

简介

根据工作方式参数 MD 以及两个输入的输入品质 flagIN1、flagIN2（质量码）来选择输出方式以及输出的品质。

表示

符号



算法

当工作方式参数 MD 等于 3 或 4 时，则输出 OUT 的值相应的等于 IN1 或 IN2。如果对应的输入点的品质为“坏”时(质量码为 16#0X0100 为信号可疑,质量码为 16#0X0800 为信号故障)，则品质输出 Q0 置为逻辑 1，输出保持上一个周期的值。

当工作方式参数 MD 等于 0、1、2 时，则有：

- 如果 MD=0，输出为两个输入值的平均值；
- 如果 MD=1，输出为两个输入值中的较小值；
- 如果 MD=2，输出为两个输入中的较大值；

此时，还要根据相应的输入品质来调整输出：

- 如果两个输入都是坏点，品质输出 QO 置为逻辑 1，输出保持上一个周期的值；
- 如果一个点为坏点，另一个为好点，则选择好点作为输出；
- 如果两个输入都是好点，则根据两者之间的偏差是否超过偏差设定值 DB，进行如下选择：

若偏差越限，品质输出 QO 为逻辑 1，则输出保持上一个周期的值；

若偏差没有越限，品质输出 QO 为逻辑 0，输出工作方式的值。

品质输出

- 如果输入点（IN1、IN2）为坏点，则相应的输出品质（Q1、Q2）为逻辑 1；
- 如果两个输入任何一个为坏点，则品质总输出 QQ 为逻辑 1。

参数描述

参数	数据类型	含义
IN1	SFLOAT	输入 1
IN2	SFLOAT	输入 2
flagIN1	INT	输入 1 的品质
flagIN2	INT	输入 2 的品质
MD	INT	选择输出方式
DB	SFLOAT	偏差上限值
OUT	SFLOAT	输出值
Q1	BOOL	IN1 的品质输出
Q2	BOOL	IN2 的品质输出
QQ	BOOL	品质总报警
QO	BOOL	品质输出

2. 通讯辅助函数

GETBIT 模块

简介

从输入的 DWORD 型变量中取出指定位的标号，如果为 1，则输出为 ON，如果为 0，则输出为 OFF。

表示

符号

算法

通过 1 移动指定 SERIAL 位与输入 X 按位相与，即可获得指定位的信息

参数描述

参数	数据类型	含义
X	DWORD	输入值
SERIAL	UINT	位的序号[0，31]
Q	BOOL	输出值

说明：

1.程序中对 SERIAL 判断是否大于等于 16，与原先的 XA 芯片的效率有关。

2.以下很多模块中的 SERIAL 变量，在程序中是 INT 型，在 SCControl 中模块的引脚是 UINT 型，不一致，这里没有影响，所以没统一。

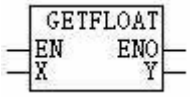
GETFLOAT 模块

简介

把传入的 DWORD 型变量 X 解释为 FLOAT 型输出。

表示

符号



参数描述

参数	数据类型	含义
X	DWORD	输入变量
Y	FLOAT	输出变量

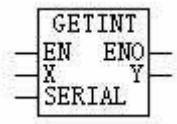
GETINT 模块

简介

从 32 位的 DWORD 型输入变量 X 的指定位置取出 16 位的 INT 型变量，当 SERIAL＝0 时，取低 16 位；当 SERIAL＝1 时，取高 16 位。

表示

符号



算法

当 SERIAL＝0 时，把输入变量和 0X0FFFF 按位与，返回 INT 型结果；

当 SERIAL＝1 时，把输入变量右移 16 位后返回 INT 型。

参数描述

参数	数据类型	含义
X	DWORD	输入值
SERIAL	UINT	序号[0,1]
Y	INT	输出值

GETMSG 模块

简介

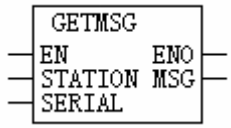
该模块用来从其他控制站取传送过来的消息，每个控制站都有一个接收缓冲区：

DWORD msg[4096]

它用来存放从网络上获取的其他控制站发过来的共享信息，其中每个控制站 512 个字节的信息。

表示

符号



算法

该模块根据输入的控制站序号和消息序号，直接从接收缓冲区中读出想要的哪个控制站的什么信息。

参数描述

参数	数据类型	含义
STATION	UINT	控制站号
SERIAL	UINT	消息序号[0,127]
MSG	DWORD	读取的消息

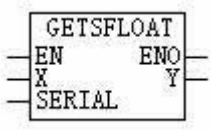
GETSFLOAT 模块

简介

该模块的功能是从输入的 32 位 DWORD 型值的指定位置取 16 位的 SFLOAT 型值， 当 SERIAL=0，取低 16 位；当 SERIAL=1，取高 16 位。

表示

符号



算法

同 GETINT 模块一样，所不同的是返回的类型是 SFLOAT 型。

参数描述

参数	数据类型	含义
X	DWORD	输入值
SERIAL	UINT	序号[0,1]
Y	SFLOAT	输出值

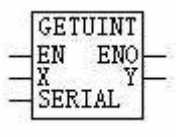
GETUINT 模块

简介

从 32 位的 DWORD 型输入变量 X 的指定位置取出 16 位的 UINT 型变量，当 SERIAL＝0 时，取低 16 位；当 SERIAL＝1 时，取高 16 位。

表示

符号



算法

当 SERIAL＝0 时，把输入变量和 0X0FFFF 按位与，返回 INT 型结果；

当 SERIAL＝1 时，把输入变量右移 16 位后返回 INT 型。

参数描述

参数	数据类型	含义
X	DWORD	输入值
SERIAL	UINT	序号[0,1]
Y	UINT	输出值

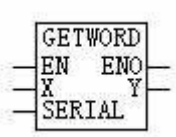
GETWORD 模块

简介

从 32 位的 DWORD 型输入变量 X 的指定位置取出 16 位的 WORD 型变量，当 SERIAL＝0 时，取低 16 位；当 SERIAL＝1 时，取高 16 位。

表示

符号



算法

当 SERIAL＝0 时，把输入变量和 0X0FFFF 按位与，返回 INT 型结果；

当 SERIAL＝1 时，把输入变量右移 16 位后返回 INT 型。

参数描述

参数	数据类型	含义
X	DWORD	输入值
SERIAL	UINT	序号[0,1]
Y	WORD	输出值

SENDMSG 模块

简介

该模块的功能是设置发送消息的个数，每个消息占 4 个字节，SIZE 定义消息的个数，消息内容放在 g_msg[ ]中，每个控制站都有共享的发送数据区：DWORD g_msg[128]，用作控制站之间的数据交互。

表示

符号

算法

该模块根据输入的消息个数，使能控制站共享信息广播，广播指定个数的消息到 SCnet II 网络，其他控制站可以收到该控制站发送的消息。

参数描述

参数	数据类型	含义
SIZE	UINT	消息个数

SETBIT 模块

简介

在输入的 X 变量的指定位置设置开关数据，其余不变，然后输出。

表示

符号

算法

当输入开关数据 Q=0 时，通过中间变量 1 左移 SERIAL 位后得到的数据的反码与输入变量 X 按位与；

当输入开关数据 Q=1 时，通过中间变量 1 左移 SERIAL 位后得到的数据与输入变量 X 按位或；

参数描述

参数	数据类型	含义
X	DWORD	输入数据
Q	BOOL	开关数据
SERIAL	UINT	位置序号[0,31]
Y	DWORD	输出变量

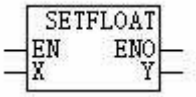
SETFLOAT 模块

简介

把输入的 FLOAT 型变量解释为 DWORD 型输出。

表示

符号



参数描述

参数	数据类型	含义
X	FLOAT	输入变量
Y	DWORD	输出变量

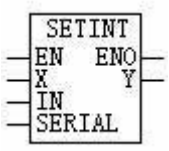
SETINT 模块

简介

在输入的 DWORD 型变量 X 指定位置设定为 INT 型输入变量 IN，其他各位不变，再赋给输出值，当 SERIAL＝0，将输入 X 的低 16 位设置为 IN，当 SERIAL＝1，将输入 X 的高 16 位设置为 IN；

表示

符号



算法

先将输入的 INT 型变量强制转换成 UINT 型：

当 SERIAL＝0 时，先将输入 X 的低 16 位清 0，然后与输入 IN 相或；

当 SERIAL＝1 时，先将输入 X 的高 16 位清 0，然后与左移 16 位后的 IN 相或；

参数描述

参数	数据类型	含义
X	DWORD	输入数据
IN	INT	输入设定值
SERIAL	UINT	位置序号[0,1]
Y	DWORD	输出变量

SETSFLOAT 模块

简介

在输入的 32 位的 DWORD 型变量 X 指定位置设定 16 位的 SFLOAT 型输入变量 SF，其他各位不变，再赋给输出值，当 SERIAL＝0，将输入 X 的低 16 位设置为 SF，当 SERIAL＝1，将输入 X 的高 16 位设置为 SF。

表示

符号

算法

参照 SETINT 模块

参数描述

参数	数据类型	含义
X	DWORD	输入数据
SF	SFLOAT	输入设定值
SERIAL	UINT	位置序号[0,1]
Y	DWORD	输出变量

SETUINT 模块

简介

该模块实现在输入的 32 位的 DWORD 型变量的指定位置设置 16 位 UINT 型变量值，其余不变。当 SERIAL=0，将输入 X 的低 16 位设置为 UINT 型输入 IN；当 SERIAL=1，将输入 X 的高 16 位设置为 UINT 型输入 IN。

表示

符号

算法

先将输入的 INT 型变量强制转换成 UINT 型：

当 SERIAL＝0 时，先将输入 X 的低 16 位清 0，然后与输入 IN 相或；

当 SERIAL＝1 时，先将输入 X 的高 16 位清 0，然后与左移 16 位后的 IN 相或；

参数描述

参数	数据类型	含义
X	DWORD	输入数据
X	UINT	输入设定值
SERIAL	UINT	位置序号[0,1]
Y	DWORD	输出变量

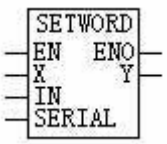
SETWORD 模块

简介

该模块实现在输入的 32 位的 DWORD 型变量的指定位置设置为 16 位的 WORD 型值，其余各位保持不变，当 SERIAL＝0，将输入 X 的低 16 位设置为 16 位 WORD 型输入 IN；当 SERIAL＝1，将输入 X 的高 16 位设置为 16 位 WORD 型输入 IN。

表示

符号



算法

先将输入的 INT 型变量强制转换成 UINT 型：

当 SERIAL＝0 时，先将输入 X 的低 16 位清 0，然后与输入 IN 相或；

当 SERIAL＝1 时，先将输入 X 的高 16 位清 0，然后与左移 16 位后的 IN 相或；

参数描述

参数	数据类型	含义
X	DWORD	输入数据
X	WORD	输入设定值
SERIAL	UINT	位置序号[0,1]
Y	DWORD	输出变量

3. 累积函数

ACCUM_TO_AISUM 模块

简介

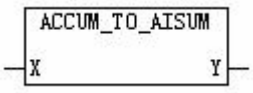
该模块的功能是把 SCControl 中的累积量转化为模拟量累积量，它实现的功能与 ACCUM_TO_SUM0 和 ACCUM_TO_SUM1 两个的模块是一样的，只是它的输出是个模拟量，而不需要用两个功能块输出一个模拟量累积量的两部分。

SCControl 中的累积量和模拟量累积量的数据类型说明详见帮助中的累积数据类型说明。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



举例

输入的 SCControl 的累积量：X＝134217719.123291，输出模拟量的累积量为：Y.SUM0＝7.123291，Y.SUM1＝8388607。

参数描述

参数	数据类型	含义
X	structAccum	输入值
Y	structAI	输出值

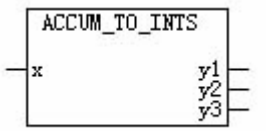
ACCUM_TO_INTS 模块

简介

该模块的功能是将累积变量结构中的 INT 型变量分别输出。

表示

符号



参数描述

参数	数据类型	含义
x	structAccum 型	输入值
y1	INT 型	输出值 （x 的小数部分）
y2	INT 型	输出值 （x 整数部分的低位）
y3	INT 型	输出值 （x 整数部分的高位）

实现方法

分别输出累积变量结构中的三个 INT 型变量

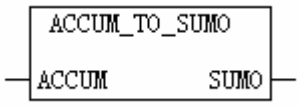
ACCUM_TO_SUM0 模块

简介

该模块的功能是将 SCControl 累积量转换成模拟量累积，输出模拟量累积的 SUM0 内容。

表示

符号



参数描述

参数	数据类型	含义
ACCUM	structAccum 型	输入值
SUM0	SFLOAT 型	输出值

实现方法

输入累积量的整数部分转换成 LONG 型，取最后四位（二进制）来表示-7～+7（SFLOAT 的整数部分），输入累积量的小数部分直接作为 SFLOAT 的小数部分。

注意

SUM0 必须连接到 AI 结构中的 SUM0 项中，否则，无意义。

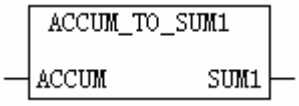
ACCUM_TO_SUM1 模块

简介

该模块的功能是将累积量转换成模拟量累积，输出模拟量累积的 SUM1 内容。

表示

符号



参数描述

参数	数据类型	含义
ACCUM	structAccum 型	第一输入值
SUM1	LONG 型	输出值

实现方法

去掉输入累积量的小数部分，整数部分转换成 LONG 型右移四位处理，得数即为所求 SUM1。

注意

SUM1 必须与 AI 结构中的 SUM1 项相连，否则，无意义。

ADD_ACCUM 模块

简介

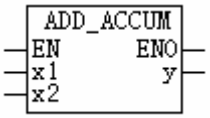
该模块的功能是将累积量相加，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

X1、X2 的量纲必须一致，否则出错。

表示

符号



公式

Y = X1 + X2

参数描述

参数	数据类型	含义
X1	structAccum 型	第一输入值
X2	structAccum 型	第二输入值
Y	structAccum 型	输出值

实现方法

将两个输入累积量的小数部分与整数部分分别以整数形式相加，其中小数部分考虑到进位，然后将得数转换成累积量的表达方式。

ADD_ACCUM_RANGE 模块

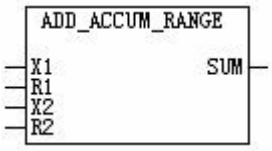
简介

该模块的功能是把两个不同量程的 SCControl 中累积量相加，输出的和按照两个输入量程中较大的量程来表示，使用时要注意两个累积量的和不能超过累积量的表示范围。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



算法

当输入量程 R1>R2 时，输出 SUM＝X1＋X2*R2/R1。

当输入量程 R2>R1 时，输出 SUM＝X1*R1/R2＋X2。

举例

输入的累积量和相应的量程为：X1＝5.599854，R1＝200，X2＝56.000000，R2＝500，输出累积的和为：SUM＝58.239746。

参数描述

参数	数据类型	含义
X1	structAccum	输入累积量 1
R1	INT	输入累积量 1 的量程
X2	structAccum	输入累积量 2
R2	INT	输入累积量 2 的量程
SUM	structAccum	输出和值

AISUM_TO_ACCUM 模块

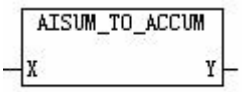
简介

该模块的功能是把模拟量的累积量转化为 SCControl 中的累积量，它实现的功能和 SUM_TO_ACCUM 模块是一样的，只是它的输入为一个模拟量，而不是模拟量累积量的两部分 SUM0 和 SUM1。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



举例

输入的模拟量的累积量为 X.SUM0＝7.123047，Y.SUM1＝8388607，输出的 SCControl 中累积量：Y＝134217719.123047。

参数描述

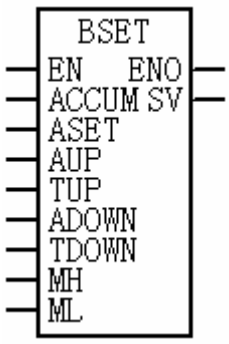
参数	数据类型	含义
X	structAI	输入值
Y	structAccum	输出值

BSET 模块

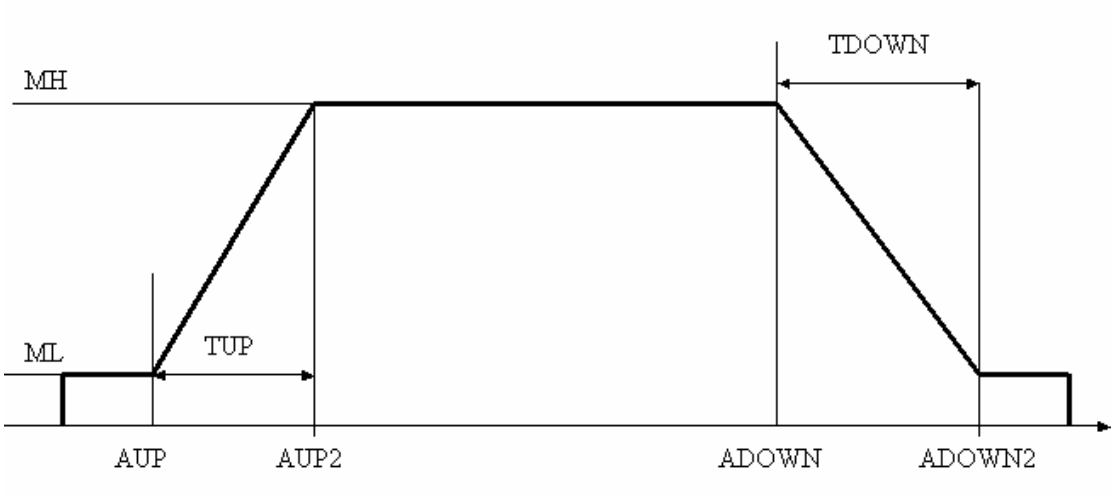
简介

该模块的功能是完成定量设定单元的输出阀位设定值的计算。

符号



SV = BSET(ACCUM,ASET,AUP,TUP,ADOWN,TDOWN,MH,ML)



说明

在定量设定的处理中，开始累积后需要将阀门开到一个阀位低限值 ML，等累积到初始上升累积值 AUP 后，输出阀位按一定速率上升直到阀位高限值 MH。等累积到初始下降值 ADOWN 后，输出阀位按一定速率下降直到阀位低限值 ML。一直累积到设定值后，将阀门关死。此模块是根据当前现场累积量 ACCUM,来作相应阀位设定的。

在上升阶段将累积 (MH + ML) * TUP / 2 。

在下降阶段将累积 (MH + ML) * TDOWN /2

参数描述

参数	数据类型	含义
ACCUM	structAccum	当前累积值
ASET	structAccum	设定累积值
AUP	FLOAT	开始上升累积值
TUP	UINT	上升时间（秒）
ADOWN	FLOAT	开始下降累积值
TDOWN	UINT	下降时间（秒）
MH	SFLOAT	阀位高限
ML	SFLOAT	阀位低限
SV	SFLOAT	阀位设定值

COMP_ACCUM 模块

简介

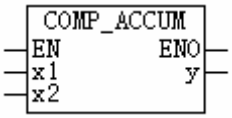
该模块的功能是比较累积量，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

X1、X2 的量纲必须一致，否则出错。

表示

符号



公式

当 X1 > X2 时　　Y = 1

当 X1 = X2 时　　Y = 0

当 X1 < X2 时　　Y =-1

参数描述

参数	数据类型	含义
X1	StructAccum	第一输入
X2	StructAccum	第二输入
Y	INT	输出

实现方法

先将两个累积量的整数部分做比较，如整数部分相等再比较小数部分，最后输出结果。

CONVERT_ACCUM 模块

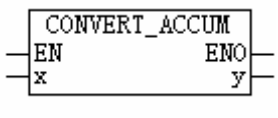
简介

该模块功能是将 structAccum 型的输入值转化为 FLOAT 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



参数描述

参数	数据类型	含义
IN	StructAccum	输入值
OUT	FLOAT	输出值

实现方法

分别把小数部分与整数部分转化成 FLOAT 型然后相加。

CONVERT_TO_ACCUM 模块

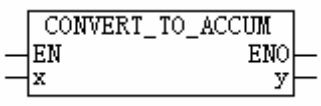
简介

该模块功能是将 FLOAT 型的输入值转化为 structAccum 型数据类型。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



参数描述

参数	数据类型	含义
IN	FLOAT	输入值

OUT	StructAccum	输出值

实现方法

先将 FLOAT 型输入值的整数部分高 16 位与低 16 位（二进制）分别存为 accum2 和 accum1，然后将小数部分乘以 4096 变成 SFLOAT 型作为累积量的小数部分记录。

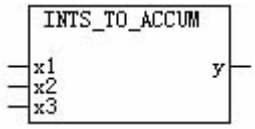
INTS_TO_ACCUM 模块

简介

该模块的功能是将三个二字节变量合并成一个累积量，以方便实现累积量的冗余。

表示

符号



参数描述

参数	数据类型	含义
x1	INT 型	输入值 （y 的小数部分）
x2	INT 型	输入值 （y 整数部分的低位）
x3	INT 型	输入值 （y 整数部分的高位）
y	structAccum 型	输出值

实现方法

将三个 INT 型输入变量分别作为累积变量结构中三个部分来组成一个累积型变量。

SUB_ACCCUM 模块

简介

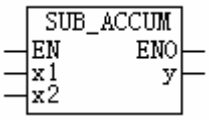
该模块的功能是将累积量相减，并将结果赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

X1、X2 的量纲必须一致，否则出错。

表示

符号



公式

Y = X1 - X2

参数描述

参数	数据类型	含义
IN1	StructAccum	被减数
IN2	StructAccum	减数
OUT	StructAccum	输出

实现方法

先将整数部分做减法，再小数部分做减法（包括借位处理），然后合成输出累积量。

SUB_ACCUM_RANGE 模块

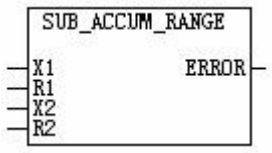
简介

该模块的功能是把输入的两个不同量程的 SCControl 累积量相减，输出的差按照两个输入量程中大的那个量程来表示。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



算法

当输入量程 R1>R2 时，输出 SUM＝X1－X2*R2/R1。

当输入量程 R2>R1 时，输出 SUM＝X1*R1/R2－X2。

举例

输入的累积量和量程分别为：X1＝366.559814，R1＝320，X2＝105.329834，R2＝5000，输出两个累积量的差为：ERROR＝-81.870117。

参数描述

参数	数据类型	含义
X1	structAccum	输入累积量 1
R1	INT	输入累积量 1 的量程
X2	structAccum	输入累积量 2
R2	INT	输入累积量 2 的量程
ERROR	structAccum	输出差值

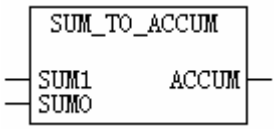
SUM_TO_ACCUM 模块

简介

该模块的功能是将模拟量累积转换成累积量。

表示

符号



参数描述

参数	数据类型	含义
SUM1	LONG 型	模拟量累积的 SUM1 内容
SUM0	SFLOAT 型	模拟量累积的 SUM0 内容
ACCUM	structAccum 型	输出值

实现方法

将输入值 SUM0 取出后 12 位（二进制）作为累积量的小数部分，SUM1 左移四位再加上 SUM0 的高四位（二进制）作为累积量的整数部分。

TOTAL_ACCUM 模块

简介

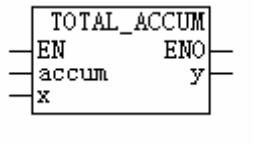
该模块的功能是进行累积，Y 与 accum 引脚用同一变量，此变量以 X 每秒的速度递增。

EN 和 ENO 能作为附加参数加以设置。

accum、x 的量纲必须一致，否则出错。

表示

符号



公式

Y = accum + X

参数描述

参数	数据类型	含义
ACCUM	structAccum	累积变量
X	SFLOAT	递增量
Y	structAccum	输出

实现方法

将累积量整数部分转成 LONG 型，小数部分与递增量相加（包括进位处理），然后合成累积量输出。

注意

该模块每秒累加一次，但必须满足以下条件：

1.主控卡的运算周期不大于 1s；

2.该模块所在程序的运行周期必须是 1Ts,即每个主控卡运算周期均调用一次该模块。

以上两个条件的任一条件不满足，可能造成该模块工作不正常！

4. 辅助计算

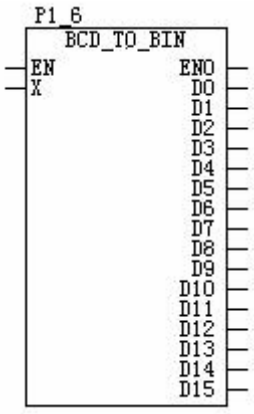
BCD 码转二进制模块

简介

该模块将输入的 4 位 BCD 码转换成 16 位二进制位输出。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

设输入的 BCD 码为 B3B2B1B0，将其各位提出转换为 4 位的二进制数表示，如 A15A14A13A12…A3A2A1A0，然后判断各个二进制位是 1 还是 0，对应的输出 ON 还是 OFF。

参数描述

参数	数据类型	含义
X	UINT	输入的 BCD 码
D0~D15	BOOL	输出二进制位状态

BCD 码转十进制模块

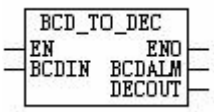
简介

该模块把一个 BYTE 型 BCD 码输入值转化为 BYTE 型十进制输出值，由于输入是 BYTE 型，所以输入的 BCD 码最大值为 99。 由于单个 BCD 码表示范围是[0，9]，所以当输入 BCD 码的个位超过 9 时，输出 BCD 码报警标志值为 0X0F（15）；当输入 BCD 码的十位超过 9 时，输出 BCD 码报警标志值为 0XF0（240）；当输入 BCD 码的个位和十位都超过 9 时，输出 BCD 码报警标志值为 0XFF（255）。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



DECOUT=BCDt_TO_DEC(BCDIN)

参数描述

参数	数据类型	含义
BCDIN	BYTE	BCD 码输入值
DECOUT	BYTE	十进制输出值
BCDALM	BYTE	BCD 码的报警标志

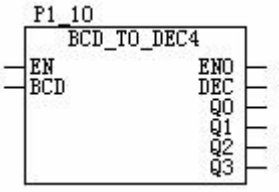
四位的 BCD 码转十进制模块

简介

该模块将输入的 4 位 BCD 码转换为十进制数。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

设输入的 BCD 码为 :B3B2B1B0,则对应的十进制输出应该为 :B3*1000+B2*100+B1*10+B0，由于 BCD 码表示的数在 0～9 之间，如果相应的 BCD 码超过 9，输出的报警标志会置 ON。例如：输入 B0>9，则置 Q0 = ON，输出的十进制对应的位为 0。

参数描述

参数	数据类型	含义
BCD	UINT	输入的 BCD 码
DEC	UINT	输出十进制数
Q0~Q3	BOOL	BCD 码超限报警位

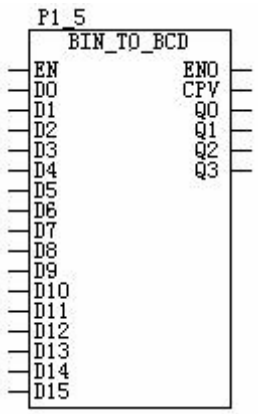
二进制转 BCD 码模块

简介

该模块把输入的 16 个开关量组成的二进制数转换为 BCD 码输出。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

设输入的 16 位二进制数为：A15A14A13A12…A3A2A1A0，将其 4 位一组的分成 4 组，每 4 位表示一个 BCD 码，分别为 B3、B2、B1、B0，则它表示的数为 B3*1000+B2*100+B1*10+B0，且由于 BCD 码表示的数在 0～9 之间，所以如果有某 4 位二进制数表示的数超过 9，则给出相应的报警，即对应的 Qi 为 ON，相应的 BCD 码输出为 0。

参数描述

参数	数据类型	含义
D0~D15	BOOL	输入的开关量
CPV	UINT	输出的 BCD 码
Q0	BOOL	D0~D3 对应的 BCD 码报警位

Q1	BOOL	D4~D7 对应的 BCD 码报警位
Q2	BOOL	D8~D11 对应的 BCD 码报警
Q3	BOOL	D12~D15 对应的 BCD 码报警

十进制转 BCD 码模块

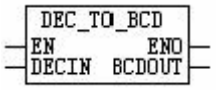
简介

该模块功能是将一个 BYTE 型输入值转化为 BCD 码输出显示，输入的十进制数是 BYTE 型的，它最大值是 255。

EN 和 ENO 能作为附加参数被加以设置。

表示

符号



BCDOUT = INT DEC_TO_BCD(DECIN)

参数描述

参数	数据类型	含义
DECIN	BYTE	十进制数输入
BCDOUT	INT	BCD 码输出

四位的十进制转 BCD 码

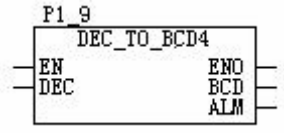
简介

将输入的无符号十进制整数转为 BCD 码。

注意：由于四位的 BCD 码表示的最大数为 9999，所以输入的十进制数不能超过 9999，否则输出的 BCD 码数置 0，并且置报警标志位为 ON。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

设输入的十进制数为 :D3D2D1D0，将其除 10 可得个位的数 D0，除 100 可得十位的数 D1，用同样的方法可得 D2、D3，然后分别用 4 个二进制位表示其中一个值，组合后输出。由于输出在 SCControl 中是以十进制显示的，所以应将输出的值转为十六进制数观察转换是否正确。

参数描述

参数	数据类型	含义
DEC	UINT	输入的十进制数
BCD	UINT	输出的 BCD 码
ALM	BOOL	输入超限报警标志

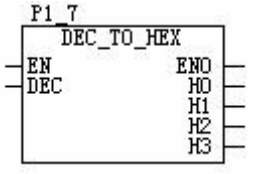
十进制转十六进制模块

简介

该模块将一个无符号的十进制整数转换为十六进制输出，由于 SCControl 不能显示字母，所以输出分成 4 个引脚显示，每个引脚显示一个十六进制数。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

从输入的无符号整数中提取 4 个 4 位的二进制数,分别赋给对应的 H0~H3 输出,即 H0~H3 表示输入的十六进制数的 4 个数码。

参数描述

参数	数据类型	含义
DEC	UINT	输入的十进制数
H0～H3	UINT	输出的十六进制 4 个位

工程量转化为无因次量模块

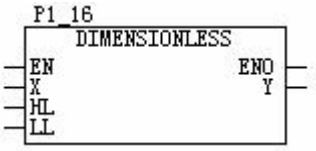
简介

该模块将输入的工程量根据设定的高低限转化为无因次量输出,输入的工程量没有硬性的限制在设定的高低限之间，以方便一些超量程使用，这一点需要用户自己注意。

注意：输入的高限必须不小于低限，否则，输出恒为 0。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

在这个模块中，输出 $Y = \frac{X - LL}{HL - LL}$。

参数描述

参数	数据类型	含义
X	FLOAT	输入的工程量
HL	LONG	设定值高限
LL	LONG	设定值低限
Y	FLOAT	输出的无因次量

无因次量转化为工程量模块

简介

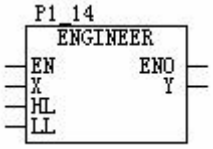
该模块将输入的无因次量根据设定的高低限，转化为工程量输出，无因次量一般在 0~1

之间，但这里不作限制，以方便某些需要超量程使用的场合，这一点需要用户自己注意。

注意：输入的高限必须不小于低限，否则，模块输出恒为 0。并且转换的精度随着输入高低限的差值增大而增加。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

在这个模块中，输出$Y = X * (HL - LL) + LL$。

参数描述

参数	数据类型	含义
X	SFLOAT	输入的无因次量
HL	LONG	设定值高限
LL	LONG	设定值低限
Y	FLOAT	输出的工程量值

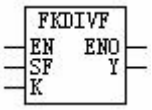
FKDIVF 模块

简介

该模块的功能是将半浮点数缩小整数倍。

表示

符号



参数描述

参数	数据类型	含义
SF	SFLOAT	需缩小的半浮点数
K	INT	缩小倍数
Y	SFLOAT	输出

实现方法

首先判断除数如果为零，那么输出 7.999756 或-8.0；两数转换成 LONG 型做除法之后，如果商超出 SFLOAT 型表示范围则用边界值表示（商大于 7.999756 则显示 7.999756，商小于-8.0 则显示-8.0）。

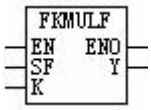
FKMULF 模块

简介

该模块的功能是将半浮点数放大整数倍。

表示

符号



参数描述

参数	数据类型	含义
SF	SFLOAT	需放大的半浮点数
K	INT	放大倍数
Y	SFLOAT	输出

实现方法

由于 SFLOAT 型变量作为 INT 型存储的，所以可以直接相乘，显示成 SFLOAT 型，如果超出 SFLOAT 表示范围则用 SFLOAT 边界值表示。(如大于 7.999756,则表示为 7.999756；如小于-8.0 则表示为-8.0）。

FKMULK 模块

简介

该模块的功能是将整数用一个半浮点数进行缩放。其中输出值 Y 四舍五入。

表示

符号

参数描述

参数	数据类型	含义
SF	SFLOAT	缩放系数
K	INT	需缩放的整数
Y	INT	输出

实现方法

将两个输入转成 LONG 型变量相乘，乘积右移 11 位，如果最后一位不为零则加一（四舍五入），再右移一位（完成乘积除以 4096 并对乘积四舍五入）。乘积作为 LONG 型变量如果超出 INT 型表示范围则取边界值（乘积大于 32767 则显示 32767，乘积小于-32768 则显示-32768）。乘积最后转成 INT 型输出。

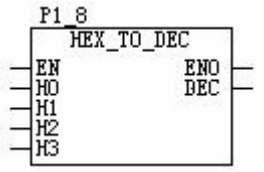
十六进制转十进制 HEX_TO_DEC

简介

该模块将输入的 4 位的十六进制数转换为十进制输出。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

将输入的 4 位十六进制数组合在一起，赋给输出值即可。

参数描述

参数	数据类型	含义
H0~H3	UINT	输入的十六进制各个位的值
DEC	UINT	输出的十进制的值

KFDIVK 模块

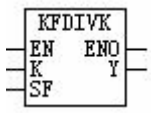
简介

该模块的功能是进行整数和半浮点数的混合除法运算。

Y = K / SF，其中输出值 Y 四舍五入。

表示

符号



参数描述

参数	数据类型	含义
K	INT	被除数
SF	SFLOAT	除数
Y	INT	输出

实现方法

首先判断除数如果为零，则输出 32767 或者-32768；被除数左移 13 位（二进制）之后两数相除，得数尾数不为零则说明有余数，加一之后再除以 2（完成被除数左移 12 位做除法，并对商进行四舍五入处理）。如果商超出 INT 型表示范围则用边界值表示（商大于 32767 则显示 32767，商小于-32768 则显示-32768）。

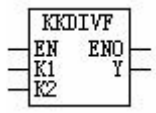
KKDIVF 模块

简介

该模块的功能是进行整数除法运算等到半浮点数。

表示

符号



参数描述

参数	数据类型	含义
K1	INT	被除数
K2	INT	除数
Y	SFLOAT	输出

实现方法

如果除数为零，则输出 7.999756 或者-8.0；如果除数不为 0，则将被除数左移 12 位（二进制）除以除数，商如果超出 SFLOAT 型表示范围则显示边界值。（商大于 7.999756 则显示 7.999756，商小于-8.0 则显示-8.0）

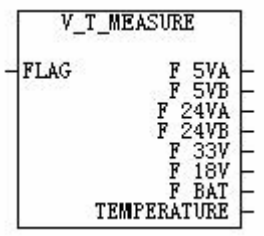
电压温度检测模块

简介

当电压温度检测标志 FLAG=ON 时，该模块输出主控卡的各种电压和主控卡环境温度值。

EN 和 ENO 能作为附加参数加以设置。

表示



参数描述

参数	数据类型	含义
FLAG	BOOL	电压温度检测标志
F_5VA	FLOAT	5VA 电压值
F_5VB	FLOAT	5VB 电压值
F_24VA	FLOAT	24VA 电压值
F_24VB	FLOAT	24VB 电压值
F_33V	FLOAT	3.3V 电压值
F_18V	FLOAT	1.8V 电压值
F_BAT	FLOAT	后备电池电压
TEMPERATURE	FLOAT	主控卡环境温度

5. 输入处理

加速度测量模块

简介

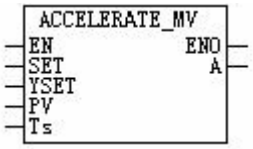
该模块使用移动平均值的方法来求指定时间 Ts 内平均加速度。

注意：指定时间间隔 Ts 以 0.1S 为单位。

EN 和 ENO 作为附加参数加以设置。

表示

符号



算法

当置位开关 SET＝ON 时，输出等于设定值 YSET，内部隐藏变量清 0。

当置位开关 SET＝OFF 时，输出 Ts 时间内的平均加速度，算法如下：

n 时刻的移动平均值：$MV[n]=\frac{V[m+1]+V[m+2]+\Lambda\ +V[n]}{n-m}$

n-1 时刻的移动平均值：$MV[n-1]=\frac{V[m]+V[m+1]+\Lambda\ +V[n-1]}{n-m}$

由上两式可得：$MV[n]=MV[n-1]+\frac{V[n]-V[m]}{n-m}$

令 V[m]=MV[n]，则有：$MV[n]=MV[n-1]+\frac{V[n]-MV[m]}{n-m}$

化简后得：$MV[n]=\frac{(n-m)*MV[n-1]+V[n]}{n-m+1}$

而此时加速度: $A=\frac{V[n]-V[m]}{(n-m)*T0}$，其中 T0 为控制周期

化简得：$A=\frac{MV[n]-MV[n-1]}{T0}$

参数描述

参数	数据类型	含义
SET	BOOL	置位开关
YSET	FLOAT	设定值
PV	FLOAT	输入值
Ts	UINT	时间间隔[以 0.1s 为单位]

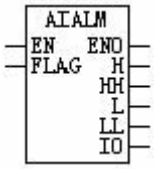
模入报警分析模块

简介

输入为模拟量质量码，输出为该质量码的报警信息。

表示

符号



算法

在主控卡控制程序里面对输入的 PV 值进行了报警处理，对高高限报警、高限报警、低限报警、低低限报警分别置质量码为：0X05，0X01，0X02，0X06，对可疑信号和故障信号，则置质量码为：0X0100，0X0800，该模块就是对这些报警信息进行分析，对比输入的质量

码与这些数值，输出分析结果。

如果输入 FLAG＝0X05，则输出 HH＝ON，表明是高高限报警。

如果输入 FLAG＝0X0100 或 FLAG＝0X0800，则输出 IO＝ON，表明是通道故障报警。

参数描述

参数	数据类型	含义
FLAG	WORD	模拟量输入的质量码
H	BOOL	高限报警
HH	BOOL	高高限报警
L	BOOL	低限报警
LL	BOOL	低低限报警
IO	BOOL	通道故障报警

泄漏报警模块

简介

该模块功能是判断当前累积输入值和上一周期的累积输入值之间的偏差是否超过泄漏设定值（LK）。当偏差超过泄漏设定值时，就置泄漏报警（LEAK）为 ON；当偏差小于泄漏设定值（LK）和滞后值（HYS）的差时，泄漏报警恢复为 OFF。主要用于锅炉流量测量和质量测量过程。

表示

符号

VOID ALM_LEAK(PV,LK,HYS,SPV,LEAK)

参数描述

参数	数据类型	含义
PV	structAccum	当前累积量输入值
LK	SFLOAT	泄漏设定值
HYS	SFLOAT	滞后值
LEAK	BOOL	泄漏报警

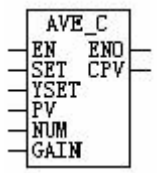
累积平均值模块

简介

该模块用来获得一段特定时间内的输入数据的平均值，其中输入数据是每个扫描周期输入一次，不到规定时间则输出上一段特定时间完成后的输出值。

表示

符号



VOID AVE_C (SET,YSET,PV,NUM,GAIN,CPV)

算法

当置位开关 SET=ON 时，输出 CPV=YSET，内部累积和计数变量清 0。

当置为开关 SET=OFF 时，输出 $CPV = Gain \times \frac{SPV}{NUM}$，其中 SPV 为 NUM 个扫描周期内输入变量的累积值。

参数描述

参数	数据类型	含义
SET	BOOL	置位开关
YSET	SFLOAT	设定值
PV	SFLOAT	过程变量输入值
NUM	UINT	扫描周期数
GAIN	SFLOAT	增益
CPV	SFLOAT	计算输出的累积平均值

过热蒸汽温压补偿模块（差压信号）

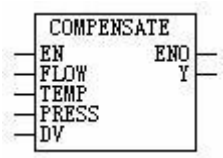
简介

该模块的功能是对过热蒸汽进行温压补偿。设计状态下的蒸汽比容—DV 可用过热蒸汽特性计算器求出。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



算法

根据蒸气的实际温度和实际压力值，查表得到其实际比容，采用下面公式进行处理，

$$y = \frac{DesignV}{\text{实际比容}} \times Flow0$$

参数描述

参数	数据类型	含义
FLOW	SFLOAT	孔板流量计测量的差压信号无因次化值，数值范围为 0~100%
TEMP	FLOAT	实际工作温度，单位为℃，范围为 260℃－800℃
PRESS	FLOAT	实际工作绝对压力，单位为 Mpa，范围为 0.5Mpa－32Mpa。
DV	FLOAT	设计比容，单位为 cm3/g
Y	SFLOAT	补偿后的流量，为无因次量，数值范围为 0~100%

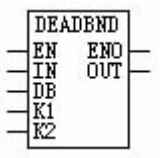
死区模块

简介

该模块设置一个死区[－DB，＋DB]，对输入进行死区处理。

表示

符号



算法

当输入 IN 在死区内，输出为 OUT=0；

当输入 IN >= DB，输出 OUT=（IN－DB）*K1；

当输入 IN <= －DB，输出 OUT=（IN＋DB）*K2；

参数描述

参数	数据类型	含义
IN	SFLOAT	输入
DB	SFLOAT	死区
K1	SFLOAT	上升比例系数
K2	SFLOAT	下降比例系数
OUT	SFLOAT	输出值

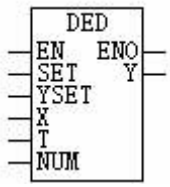
纯滞后模块

简介

输入延时一段时间后输出，其延时时间为：t=NUM*T。

表示

符号



算法

当置位开关 SET＝ON，输出 Y=YSET，通常 用于初始化。

当置位开关 SET＝OFF，输出 为输入的延时输出。

其中传递函数：$G(s) = Y(s) / X(s) = e^{-s*t}$。

参数描述

参数	数据类型	含义
SET	BOOL	置位开关
YSET	SFLOAT	设定值
X	SFLOAT	输入值
T	UINT	定时时间间隔（以 0.1 秒为单位）
NUM	UINT	定时时间间隔数目，其中 0<NUM<=50
Y	SFLOAT	输出值

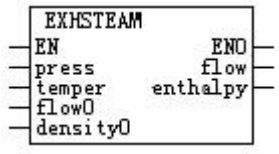
过热蒸汽流量补偿模块（流量信号）

简介

对温度范围 140-560 摄氏度，压力范围 0.2MPa-16.0MPa 内的过热蒸汽进行流量补偿。

表示

符号



算法

根据实际压力和实际温度，通过查表法得到焓值与比容 μ，而其密度 $\rho = \frac{1}{\mu}$，然后可以采用下面公式进行处理：

$$flow = \sqrt{\rho / density0} \times flow0$$

该功能块是过热蒸气进行温压补偿处理，输入 flow0 就是测量流量的无因次化值，输出 flow 为补偿后的流量无因次化值，不需要其它处理。

参数描述

参数	数据类型	含义
flow0	SFLOAT 型	表示测量蒸气流量，为无因次量，数值范围为 0－100%
press	FLOAT 型	标准大气压下所测得的相对压力，单位为 Kpa，其范围为 189.86777－15989.867777kpa。
temper	FLOAT 型	表示实际温度，单位为摄氏度，范围为 140℃－560℃
density0	FLOAT 型	蒸气设计密度，单位为 Kg/m3
flow	SFLOAT 型	补偿后的无因次化值
enthalpy	FLOAT 型	表示当前蒸气的焓值，单位为千焦每千克

折线表插值模块

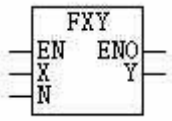
简介

在 SCControl 中，用户可以自己定义二维折线表，二维折线表的 X 轴和 Y 轴都是 10 段的，包括 11 个点，通过该模块，用户可以对输入进行线性插值处理，线性化后输出。

注意：这个模块只对二维折线表进行处理，对一维折线表没有作用。

表示

符号



算法

程序先对输入的折线表序号 N 进行越限判断，如果 N 在区间[0，63]内，则判断输入 X 的值落在 10 段中的那一段，然后在进行线性插值，具体算法如下：

当输入 X 落在 X[i]和 X[i＋1]之间，

输出：

$$Y = Y[i] + (Y[i+1] - Y[i]) \times \frac{X - X[i]}{X[i+1] - X[i]}$$

如果输入 X 小于 X[0],则输出 Y 等于 Y[0] 如果输入 X 大于 X[10]时,输出 Y 等于 Y[10]。

如果 N 不在区间[0,63]内，则输出 Y 等于 0。

参数描述

参数	数据类型	含义
X	SFLOAT	输入
N	UINT	折线表序号[0，63]
Y	SFLOAT	线性插值输出

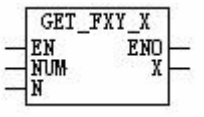
取折线表 X 值模块

简介

在 SCControl 中，用户可以自己定义二维折线表，二维折线表的 X 轴和 Y 轴都是 10 段的，包括 11 个点，通过该模块，用户可以读取 64 个折线表中指定序号 N 的折线表中 X 轴上指定序号 NUM 的坐标点的值。

表示

符号



算法

直接返回指定序号的折线表中 X 轴上指定序号的坐标点。

参数描述

参数	数据类型	含义
NUM	UINT	X 轴坐标点序号[0，10]
N	UINT	折线表序号[0，63]
X	SLFOAT	读取 X 轴上指定点的值

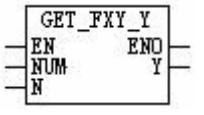
取折线表 Y 值模块

简介

在 SCControl 中，用户可以自己定义二维折线表，二维折线表的 X 轴和 Y 轴都是 10 段的，包括 11 个点，通过该模块，用户可以读取 64 个折线表中指定序号 N 的折线表中 Y 轴上指定序号 NUM 的坐标点的值。

表示

符号



算法

直接返回指定序号的折线表中 Y 轴上指定序号的坐标点。

参数描述

参数	数据类型	含义
NUM	UINT	Y 轴坐标点序号[0，10]
N	UINT	折线表序号[0，63]
Y	SLFOAT	读取 Y 轴上指定点的值

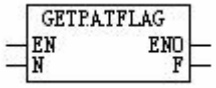
取 PAT 卡工作状态标志模块

简介

读取 PAT 第 n 个通道工作状态标志。

表示

符号



算法

取第 n 个 PAT 通道的工作状态标志。

参数描述

参数	数据类型	含义
N	INT（0<=n<64）	PAT 卡通道号
F	WORD	PAT 卡工作状态标志

取 PAT 卡 PV 值模块

简介

获取指定通道号 n 的阀位反馈值。

表示

符号

算法

直接返回第 n 个 PAT 通道的 PV 值。

参数描述

参数	数据类型	含义
N	INT	PAT 卡通道号（0<=n<64）
PV	SFLOAT	阀位反馈值

特殊操作标志读取模块

简介

读取卡件的特殊操作状态。

表示

符号

算法

用户可以通过 SETPATCON 模块来设定特殊操作命令，同样，用户如果想知道卡件的特殊状态，可以通过该模块来读取 FW342 卡回送的状态字节（Byte13），解释如下：

Byte13.0：1：硬手操增输出；　　0：无增输出；

Byte13.1：1：硬手操减输出；　　0：无减输出；

Byte13.2：1：制动状态；　　0：工作状态；

Byte13.3：1：正在自学习，　　0：自学习结束状态；

Byte13.4：1：回送自学习参数，　　0：回送用户设定参数；

Byte13.5~ Byte13.7：保留填 0。

参数描述

参数	数据类型	含义
N	INT	PAT 卡通道号（0<=n<64）
CON	INT	PAT 卡的工作状态

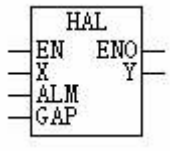
高限报警模块

简介

该模块按照设定的报警限和报警死区给出输入的一个高限报警标志。

表示

符号



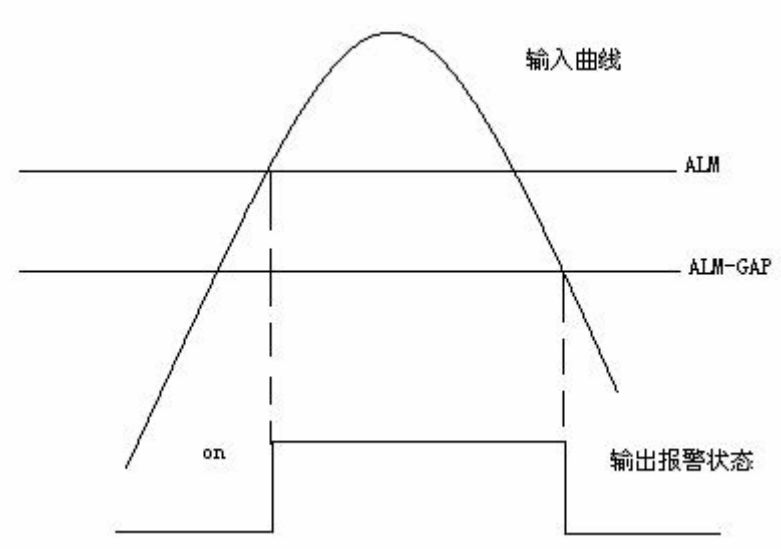
算法

当输入 X>ALM 时，输出报警开关 Y=ON，表明此时输入已经超过设定的高限了。

当输入 X<ALM－GAP 时，输出报警开关 Y=OFF，表明此时报警解除；注意：这里不是输入下降到 ALM 以下就使 Y＝OFF，使报警解除，而是有一个死区的。

当输入 X 在区间[ALM－GAP，ALM]内时，输出报警开关保持上一次的状态。

如下图所示：



参数描述

参数	数据类型	含义
X	SFLOAT	输入值

ALM	SFLOAT	高限
GAP	SFLOAT	间隙
Y	BOOL	报警开关

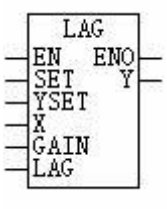
一阶滞后模块

简介

对输入信号进行一阶滞后处理，其中滞后时间 LAG 以 0.1S 为单位，就是输入 LAG=1，表示滞后时间为 0.1 秒。注意：输入 X（k）和 X（k－1）的和不得超过 SFLOAT 型的最大值 7.999756，否则会溢出，结果出错。

表示

符号



算法

当置位开关 SET=ON，输出 Y＝YSET。

当置位开关 SET＝OFF，输出 Y 是输入的一阶滞后输出，滞后时间 LAG 以 0.1S 为单位。

传递函数为：

$$G(s)=\frac{GAIN}{1+LAG\times S}$$

即

$$\frac{Y(S)}{X(S)}=\frac{GAIN}{1+LAG\times S}$$

化简后得：

$$Y(S)+LAG\times S\times Y(S)=GAIN\times X(S)$$

化到时域并差分化得：

$$Y(k)+LAG\times\frac{Y(k)-Y(k-1)}{dt}=GAIN\times X(k)$$

进一步化简得：

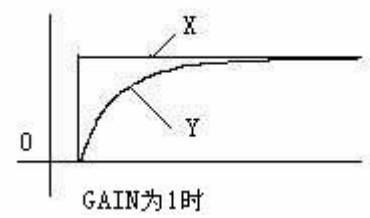
$$Y(k)=Y(k-1)+\frac{GAIN\times X(k)-Y(k-1)}{dt+LAG}\times dt$$

程序中 X(k)的地方取了 $\frac{X(k)+X(k-1)}{2}$，进行平均值滤波；

即最终公式：$Y(k)=Y(k-1)+\dfrac{GAIN\times\frac{X(k)+X(k-1)}{2}-Y(k-1)}{dt+LAG}\times dt$

传递函数的时域的阶跃响应表达式为：$Y=GAIN\times(1-e^{\frac{-t}{LAG}})$

响应曲线如下：



参数描述

参数	数据类型	含义
SET	BOOL	置位开关
YSET	SFLOAT	设定值
X	SFLOAT	输入值
GAIN	SFLOAT	增益
LAG	UINT	滞后时间常数（0.1s）
Y	SFLOAT	输出

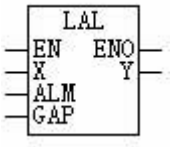
低限报警模块

简介

该模块按照设定的报警限和报警死区给出输入的一个低限报警标志。

表示

符号



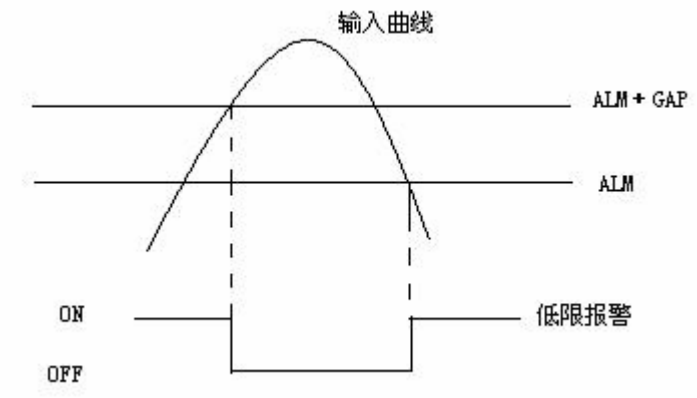
算法

当输入 X<ALM 时，则输出报警开关 Y＝ON，表明此时输入已经超过最低限了。

当输入 X>ALM+GAP 时，则输出报警开关 Y=OFF，报警解除，注意：这里不是当 X>ALM，就置报警开关 Y=OFF，使报警解除的，而是有一个死区的。

当输入 X 在区间[ALM，ALM＋GAP]中，报警开关 Y 保持上一周期的值。

如下图所示：



参数描述

参数	数据类型	含义

X	SFLOAT	输入值
ALM	SFLOAT	高限
GAP	SFLOAT	间隙
Y	BOOL	报警开关

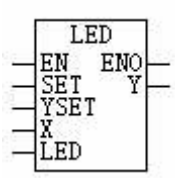
一阶超前模块

简介

对输入产生一个一阶超前输出，其中超前时间常数是以 0.1S 为单位，输入 LED＝1，就代表 0.1 秒。

表示

符号



算法

当置位开关 SET＝ON，输出 Y＝YSET。

当置位开关 SET=OFF，输出 Y 是输入一阶超前输出。

传递函数为：

$$G(S)=\frac{LED\times S}{1+LED\times S}$$

即

$$\frac{Y(S)}{X(S)}=\frac{LED\times S}{1+LED\times S}$$

化简后得：

$$Y(S)+LED\times S\times Y(S)=LED\times S\times X(S)$$

化到时域并差分化得：

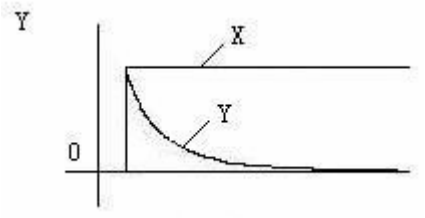
$$Y(k)+LED\times\frac{Y(k)-Y(k-1)}{dt}=LED\times\frac{X(k)-X(k-1)}{dt}$$

得最终公式：

$$Y(k)=\frac{Y(k-1)+X(k)-X(k-1)}{LED+dt}\times LED$$

传递函数在时域的阶跃响应表达式为： $Y=e^{\frac{-t}{LED}}$

响应曲线如下：



参数描述

参数	数据类型	含义
SET	BOOL	置位开关
YSET	SFLOAT	设定值
X	SFLOAT	输入值
LED	UINT	超前时间常数(0.1s)
Y	SFLOAT	输出

一阶超前滞后模块

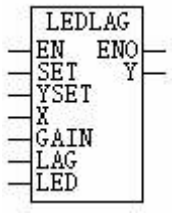
简介

该模块对输入产生一个超前滞后输出，其中超前时间常数和滞后时间常数都是以 0.1 秒为单位，该模块要求设置好恰当的超前时间常数和滞后时间常数，当 LAG>LED 时，参照下面的时域表达式可以看出此时实际上是一阶滞后输出。

当 LAG=LED 时，此时实际输出值等于实际输入值。

表示

符号



算法

当 SET=ON，输出 Y＝YSET。

当 SET＝OFF，输出 Y 是对输入 X 进行一阶超前滞后计算后的值。

其传递函数为：

$$G(s) = GAIN \times \frac{1 + LED \times S}{1 + LAG \times S}$$

即：

$$\frac{Y(S)}{X(S)} = GAIN \times \frac{1 + LED \times S}{1 + LAG \times S}$$

化简后得：

$$Y(S) + LAG \times S \times Y(S) = GAIN \times [X(S) + LED \times S \times X(S)]$$

化到时域并差分得：

$$Y(k) + LAG \times \frac{Y(k) - Y(k-1)}{dt} = GAIN \times [X(k) + LED \times \frac{X(k) - X(k-1)}{dt}]$$

最终的公式：

$$Y(k) = \frac{LAG \times Y(k-1) + [(LED + dt) \times X(k) - LED \times X(k-1)] \times GAIN}{LAG + dt}$$

传递函数在时域的阶跃响应表达式为：

$$Y = \frac{LAG - LED}{LAG} e^{\frac{-t}{LAG}}$$

参数描述

参数	数据类型	含义
SET	BOOL	置位开关
YSET	SFLOAT	设定值
X	SFLOAT	输入值
GAIN	SFLOAT	增益
LAG	UINT	滞后时间常数
LED	UINT	超前时间常数
Y	SFLOAT	输出

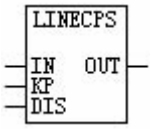
线性补偿模块

简介

根据输入的比例系数 Kp 和偏移量 DIS 对输入 IN 进行线性补偿，并输出补偿后的结果。

表示

符号



算法

根据下列公式计算补偿结果：

OUT=IN×Kp＋DIS

参数描述

参数	数据类型	含义
IN	SFLOAT	输入变量
KP	SFLOAT	比例系数
DIS	SFLOAT	偏移量
OUT	SFLOAT	输出变量

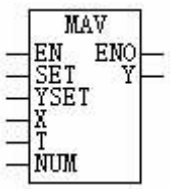
移动平均模块

简介

用来获得过去一段时间内输入的采样数据平均值，该模块输入的采样时间间隔 T 是以 0.1S 为单位的，即输入 1 就代表 0.1 秒，num 的值在[0，8]区间内取值。

表示

符号



算法

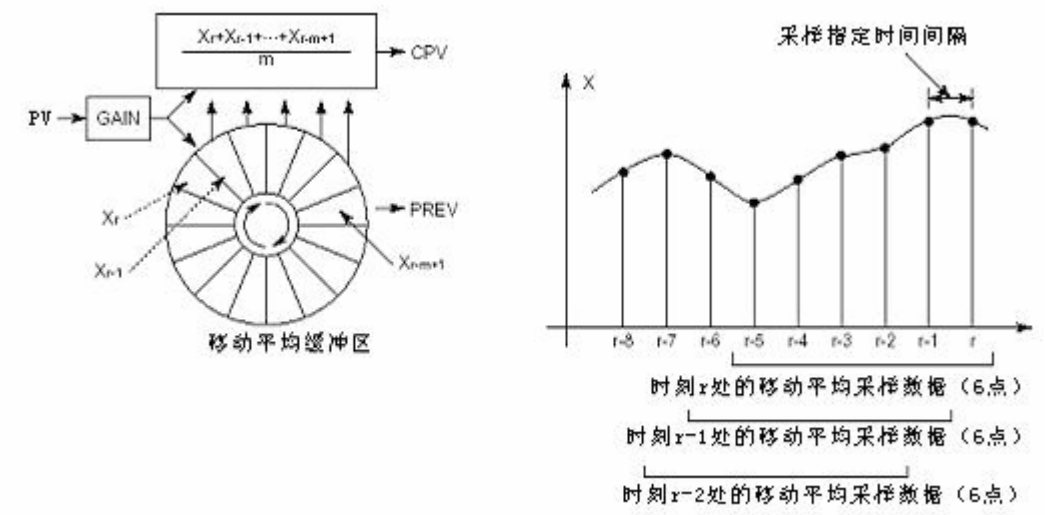
当 SET=ON，输出 Y＝YSET。

当 SET=OFF，输出 Y 是输入的移动平均值，具体算法如下：

$$CPV = GAIN * \frac{X_r + X_{r-1} + \wedge + X_{r-m+1}}{m}$$

CPV 就是输出 Y，Xi 就是输入的各个采样点的采样值，m 是采样值数量 NUM。

这种算法是对最新的一段时间内的采样数据取平均值，原理图如下：



该模块程序中开劈了 9 个输入数据的缓冲区，用来存放前几个采样时刻的采样值，所以这里采样数目 num 应该在[0，8]区间内取，当 num 大于 8 或等于 0，模块的输出直接返回输入的值，注意这对 SET=ON 和 SET=OFF 都起作用，就是说当 SET=ON，NUM=0 时，输出 Y 的值由输入 X 决定，而不是由 YSET 决定。

参数描述

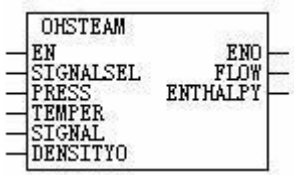
参数	数据类型	含义
SET	BOOL	置位开关
YSET	SFLOAT	设定值
X	SFLOAT	输入值
T	UINT	采样时间间隔
NUM	UINT	采样数目[0，8]
Y	SFLOAT	移动平均值输出

过热蒸汽综合补偿模块

简介

该模块必须注意 SIGNALSEL 与 SIGNAL 的关系，当 SIGNALSEL 为 OFF 时，SIGNAL 必须为差压信号，当 SIGNALSEL 为 ON 时，SIGNAL 必须为流量信号，此时补偿后的值即为实际的流量值，不需要其它处理。

表示



表达式

OHSTEAM （SIGNALSEL， PRESS， TEMPER， SIGNAL， DENSITY0， FLOW， ENTHALPY）

算法

根据实际压力和实际温度，通过查表法得到焓值与比容 μ，而其密度 $\rho = \frac{1}{\mu}$，如果为流量信号即 SIGNALSEL＝ON，采用下面公式进行处理

$$FLOW = \sqrt{\rho / DENSITY} \times SIGNAL$$

当 SIGNALSEL＝OFF 时，处理的是差压信号，其公式为：

$$FLOW = \sqrt{\rho / DENSITY0 \times SIGNAL}$$

参数描述

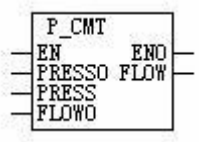
参数	数据类型	含义
SIGNALSEL	BOOL 型	信号选择开关，当它为 OFF 时，是对差压进行进行处理，当它为 ON 时，是对流量信号进行处理
PRESS	FLOAT 型	实际压力，它是在标准大气压下测得的相对压力，单位为 KPa，范围为 98.6777－15898.6777KPa
TEMPER	FLOAT 型	实际温度，单位为℃，范围为 140℃－560℃
SIGNAL	SFLOAT 型	当 SIGNALSEL 为 OFF 时必须为差压信号，当 SIGNALSEL 为 ON 时必须为流量信号。为无因次量，数值范围为 0－100%
DENSITY0	FLOAT 型	蒸汽设计密度，单位为 kg/m3
FLOW	SFLOA 型	蒸汽补偿后的值，为无因次量，数值范围为 0－100%

理想气体压力补偿模块

简介

该模块是针对那些送上来的气体流量信号已经过开方处理的情况进行温压补偿处理。

表示



表达式

FLOW = P_CMT(PRESS0，PRESS，FLOW0)

算法

$$测量实际压力值 = 压力位号量程 \times 压力无因次化值 + 压力位号量程下限$$

$$C = \frac{测量实际压力值}{设计压力值}$$

$$FLOW = FLOW0 \times \sqrt{C}$$

参数描述

参数	数据类型	含义
PRESS0	FLOAT 型	设计的压力，单位为 MPa
PRESS	structAI 型	压力补偿位号
FLOW0	SFLOAT 型	流量测量值

FLOW	SFLOAT 型	补偿后的无因次化值,数值范围为 0－100%

PAT341H 模块

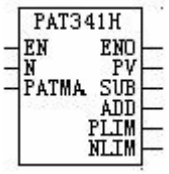
简介

该模块功能块用于对 PAT 卡（FW341）进行死区设定，并监测其报警状态。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



参数描述

参数	数据类型	含义
N	INT	PAT 通道的序号
PATMA	BOOL	死区大小选择
PV	SFLOAT	该通道的 PV 值
SUB	BOOL	是否有减脉冲输出
ADD	BOOL	是否有增脉冲输出
PLIM	BOOL	正极限报警
NLIM	BOOL	负极性报警

实现方法

当 PATMA=ON 时，即发出手动操作的指令时，PAT341H 将第 N 个 PAT 通道的死区改写成 FF；当 PATMA=OFF 时，PAT341H 将该通道死区改写为用户的设定值。同时根据其状态位确定当前所处于的状态。SUB=ON 时，表示正处于减脉冲输出阶段；ADD=ON 时，表示正处于增脉冲输出阶段；PLIM=ON 时，表示正处于正极限报警；NLIM=ON 时，表示正处于负极性报警。

PAT342H 模块

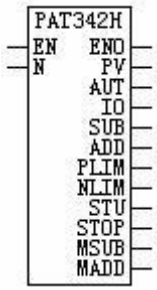
简介

该模块功能块用于 PAT 卡(FW342)中， 检测其控制电机的报警状态以及动作状态。

EN 和 ENO 能作为附加参数加以设置。

表示

符号



参数描述

参数	数据类型	含义
N	INT	PAT 通道的序号
PV	SFLOAT	该通道的 PV 值
AUTO	BOOL	电机手自动状态（ON=自动）
IO	BOOL	IO 状态是否故障（断线报警，ON=故障）
SUB	BOOL	是否有减脉冲输出（ON=有）
ADD	BOOL	是否有增脉冲输出（ON=有）
PLIM	BOOL	正极限报警（ON=报警）
NLIM	BOOL	负极限报警（ON=报警）
STU	BOOL	是否处于自学习状态（ON=自学习状态）
STOP	BOOL	是否处于制动状态（ON=制动状态）
MSUB	BOOL	是否有硬手操减输出（ON=硬手操减输出）
MADD	BOOL	是否有硬手操增输出（ON=硬手操增输出）

实现方法

读 FW342 的报警位和状态位，并：

当检测到硬手操减标识输出位由"1"到"0"的下降沿的时候将特殊操作控制字节写零；

当检测到硬手操增标识输出位由"1"到"0"的下降沿的时候将特殊操作控制字节写零；

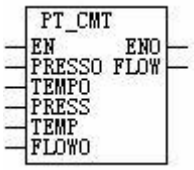
当检测到自学习标识输出位由"1"到"0"的下降沿的时候将特殊操作控制字节写零。

理想气体温压补偿模块

简介

该模块是针对那些送上来的气体流量信号已经过开方处理的情况进行温压补偿处理。

表示



表达式

FLOW = PT_CMT(PRESSO, TEMPO, PRESS, TEMP, FLOWO)

算法

$$实际工作温度值 = 温度位号量程 \times 温度无因次化值 + 温度位号量程下限$$

$$测量实际压力值 = 压力位号量程 \times 压力无因次化值 + 压力位号量程下限$$

$$C = \frac{测量实际压力值}{设计压力值} \times \frac{设计温度}{实际工作温度}$$

$$FLOW = FLOW0 \times \sqrt{C}$$

参数描述

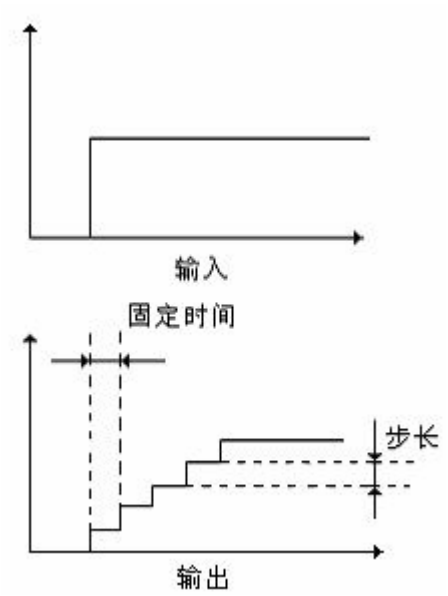
参数	数据类型	含义
PRESS0	FLOAT 型	设计的压力，单位为 MPa
TEMP0	FLOAT 型	设计温度，单位为摄氏度

PRESS	structAI 型	压力补偿位号
TEMP	structAI 型	温度补偿位号
FLOW0	SFLOAT 型	需要温压补偿的流量
FLOW	SFLOAT 型	补偿后的无因次化值，数值范围为 0－100%

斜坡模块

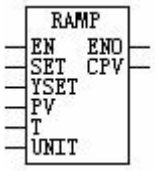
简介

当输入产生一个阶跃变化时，让输出有渐变的斜坡特性，即输出的改变变得比较平缓。如下图所示：



表示

符号



算法

当置位开关 SET=ON 时，输出 CPV=YSET，内部计时清 0。

当置位开关 SET=OFF 时，且输入 PV 不为 0 时，输出按指定的时间间隔（T）增减指定的步长值（UNIT）来达到渐变的效果，直到等于输入 PV 值。

注意：设定的时间间隔 T 以 0.1S 为单位。

参数描述

参数	数据类型	含义
SET	BOOL	置位开关
YSET	SFLOAT	设定值
PV	SFLOAT	输入值
T	UINT	特定时间设定值
UINT	SFLOAT	步长
CPV	SFLOAT	输出值

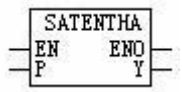
焓值计算函数模块

简介

计算压力范围是 0－15898.6777kpa 的饱和蒸汽的焓值。输入的压力单位是 KPa。

表示

符号



公式

Y = SATENTHA (P)

算法

通过查表法得到焓值。

参数描述

参数	数据类型	含义
P	FLOAT 型	实际压力,单位为 KPa，范围为 0－15898.6777KPa
Y	FLOAT 型	焓值，单位：kJ/kg

饱和蒸汽补偿模块（流量信号）

简介

该功能块是对采用孔板计量的饱和蒸气进行温压补偿处理,输入 flow0 就是测量流量的无因次化值，输出 flow 为补偿后的流量无因次化值，不需要其它处理。

表示

算法

根据实际压力，通过查表法得到焓值与比容 μ，而其密度 $\rho = \frac{1}{\mu}$，然后可以采用下面公式进行处理：

$$flow = \sqrt{\rho / density0} \times flow0$$

参数描述

参数	数据类型	含义
P	FLOAT 型	是标准大气压下测得的相对压力,单位为 KPa，范围为 0－15898.6777KPa
D	FLOAT 型	蒸汽设计密度，单位为 kg/m3
X	SFLOAT 型	表示测量蒸汽流量，为无因次量，数值范围为 0－100%
Y	SFLOAT 型	补偿后的无因次化值，数值范围为 0－100%

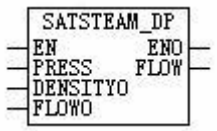
饱和蒸汽补偿模块（差压信号）

简介

该功能使用时必须注意，输入 FLOW0 为孔板测量的差压信号，补偿后的值必须经过开

方模块处理后，才是补偿后的流量值。

表示



表达式

FLOW = SATSTEAM_DP (PRESS, DENSITY0, FLOW0)

算法

根据实际压力，通过查表法得到焓值与比容 μ，而其密度 $\rho = \frac{1}{\mu}$，然后可以采用下面公式进行处理：

$$FLOW = \rho / DENSITY0 \times FLOW0$$

参数描述

参数	数据类型	含义
PRESS	FLOAT 型	是在标准大气压下测得的相对压力，单位为 KPa，范围为 0－15898.6777KPa
FLOW0	SFLOAT 型	测量蒸汽的差压信号，数值范围为 0-100%
DENSITY0	FLOAT 型	蒸汽设计密度,单位为 kg/m3
FLOW	SFLOAT 型	补偿后的无因次化值,数值范围为 0-100%

饱和蒸汽综合补偿模块

简介

该模块必须注意 SIGNALSEL 与 SIGNAL 的关系，当 SIGNALSEL 为 OFF 时，SIGANL 必须为差压信号，此时补偿后的值必须经过开方模块的处理才能得到流量的无因次化值。当 SIGNALSEL 为 ON 时，SIGANL 必须为流量信号，此时补偿后的值即为实际的流量值，不需要其它处理.

表示

表达式

FLOW = SATSTEAM_EX(SIGNALSEL, PRESS, DENSITY0, SIGNAL)

算法

根据实际压力和实际温度，通过查表法得到焓值与比容 μ，而其密度 $\rho = \frac{1}{\mu}$，如果为流量信号即 SIGNALSEL＝ON，采用下面公式进行处理

$$FLOW = \sqrt{\rho / DENSITY0} \times SINGAL$$

当 SIGNALSEL＝ON 时，处理的是差压信号，其公式为：

$$FLOW = \sqrt{\rho / DENSITY0 \times SINGAL}$$

参数描述

参数	数据类型	含义
SIGNALSEL	BOOL	信号选择开关，当它为 OFF 时，是对差压进行进行处理；当它为 ON 时，是对流量信号进行处理
PRESS	FLOAT 型	是在标准大气压下测得的相对压力，单位 KPa，范围为 0－15898.6777KPa
SIGNAL	SFLOAT 型	流量信号，当 SIGNALSEL 为 OFF 时必须为差压信号，当 SIGNALSEL 为 ON 时必须为流量信号，为无因次量，数值范围为 0-100%
DENSITY0	FLOAT 型	蒸汽设计密度，单位为 kg/m3
FLOW	SFLOAT 型	补偿后的无因次化值，数值范围为 0-100%

设折线表 X 值模块

简介

在 SCControl 中，用户可以自己定义二维折线表，二维折线表的 X 轴和 Y 轴都是 10 段的，包括 11 个点，通过该模块，用户可以设定指定序号 N 的折线表中的 X 轴上指定序号 NUM 的坐标点的值。

注意：

SCControl 中的折线表和 SCKey 中折线表已经合一，即系统总共提供了 64 个折线表，用户在 SCKey 中可以定义一维和二维折线表，SCControl 也可以定义二维折线表，他们都存放在同一片地址区内，所以两边定义的折线表序号不能相同,其他折线表相关模块也如此。

表示

符号

算法

首先判断用户输入的折线表序号和 X 轴各点的序号有没有越限，即折线表序号在区间[0，63]范围内，X 轴上各点序号在区间[0，10]之间，然后写入数据到指定序号的折线表中 X 轴上指定序号的坐标点中。

参数描述

参数	数据类型	含义
X	SFLOAT	X 轴坐标点的输入值
NUM	UINT	X 轴坐标点序号[0，10]
N	UINT	折线表序号[0，63]

设折线表 Y 值模块

简介

在 SCControl 中，用户可以自己定义二维折线表，二维折线表的 X 轴和 Y 轴都是 10 段的，包括 11 个点，通过该模块，用户可以设定指定序号 N 的折线表中 Y 轴上指定序号 NUM 的坐标点的值。

表示

符号

算法

首先判断用户输入的折线表序号和 Y 轴各点的序号有没有越限，即折线表序号在区间[0，63]范围内，Y 轴上各点序号在区间[0，10]之间，然后写入数据到指定序号的折线表 Y 轴上指定序号的坐标点中。

参数描述

参数	数据类型	含义
Y	SFLOAT	Y 轴坐标点的输入值
NUM	UINT	Y 轴坐标点序号[0,10]
N	UINT	折线表序号[0，63]

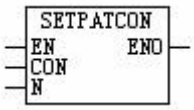
特殊操作标志设置模块

简介

设置 PAT 卡的特殊标志字节。

表示

符号



算法

void SETPATCON(WORD CON, INT N);

N：表示要操作的通道

CON：表示所进行的操作，各自代码所表示的意思为：

00： 复位命令

01: 表示制动； 02：表示启动

03： 表示自动； 04：表示手动

05： 手动增； 06：手动减

07： 启动自学习； 08：保留

09 ： 读取用户设定参数； 0A：读取自学习参数

所操作的对象为 PAT 卡组态数据区的第 13 位（PATBUF+PATLEN×n＋PAT_WR＋0xD）。

参数描述

参数	数据类型	含义
CON	要设置的工作状态参数	WORD
N	PAT 卡的通道号	INT（0=<N<64）

统计模块

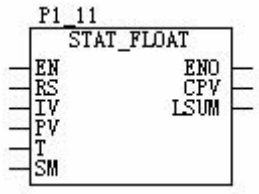
简介

该模块可以求得输入在设定的 T 时间内的累加值、平均值、最大值和最小值。

注意：当统计方式 SM 在【0，3】范围外，该模块不会做任何处理便返回。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

当复位信号 RS 从 OFF 跳变到 ON 时，输出 CPV 等于初始值 IV。

当 RS 在其他情况下：

（1）若统计方式 SM=0，选择累加方式输出，输出 T 时间内的输入数据的累加值，即 CPV(k)= CPV (k-1)+PV(k)。

（2）若统计方式 SM=1，选择平均方式输出，输出 T 时间内输入数据的平均值，即 CPV＝PV(1)+PV(2)+…+PV(k)/k。

（3）若统计方式 SM＝2，选择最大值方式输出，输出 T 时间内输入数据的最大值，即 CPV=MAX[PV(1)、PV(2)、…、PV(k)]。

（4）若统计方式 SM =3，选择最小值方式输出，输出 T 时间内输入数据的最小值，即 CPV=MIN[PV(1)、PV(2)、…、PV(k)]。

参数描述

参数	数据类型	含义
RS	BOOL	复位信号
IV	FLOAT	初始值
PV	FLOAT	输入信号
T	UINT	时间间隔（0.1S 为单位）
SM	INT	统计方式
CPV	FLOAT	输出统计值

理想气体温度补偿模块

简介

该模块是针对那些送上来的气体流量信号已经过开方处理的情况进行温压补偿处理。

表示

表达式

FLOW = T_CMT(TEMPO, TEMP, FLOWO)

算法

实际工作温度值＝温度位号量程×温度无因次化值＋温度位号量程下限

测量实际压力值＝压力位号量程×压力无因次化值＋压力位号量程下限

$$C = \frac{\text{设计温度}}{\text{实际工作温度}}$$

$$FLOW = FLOW0 \times \sqrt{C}$$

参数描述

参数	数据类型	含义
TEMP0	FLOAT 型	设计温度，单位为摄氏度
TEMP	structAI 型	温度补偿位号
FLOW0	SFLOAT 型	流量补偿位号
FLOW	SFLOAT 型	补偿后的无因次化值，数值范围为 0－100%

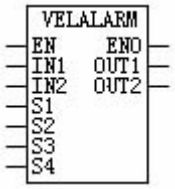
速率报警模块

简介

对输入点进行速率限制，如果超过限定值，则产生报警，工程人员可以根据报警类型采取一定的措施。该模块的输入的设定时间 S1 和 S4 都是以 0.1S 为单位的，而模块内计时变量是以主控卡采样周期为步长，递增的。

表示

符号



算法

当 IN2=ON，清报警位，输出 OUT1=OUT2=OFF；

当 IN2=OFF，进行速率报警处理。如果输入信号 IN1 在规定时间 S1 增加量超过 S2，则发生上升超速报警，OUT1=ON；在规定时间 S1 内下降量超过 S3，则发生下降超速报警，OUT2=ON。超速报警发生后开始计时，若有新的超速发生则清计数器，重新开始计时。计时至设定 S4 时，清报警。

参数描述

参数	数据类型	含义
IN1	SFLOAT	被监视点的输入
IN2	BOOL	清报警开关
S1	ULONG	报警时宽
S2	SFLOAT	增速报警限定值
S3	SFLOAT	减速报警限定值
S4	ULONG	报警回复时间
OUT1	BOOL	增速报警标志
OUT2	BOOL	减速报警标志

速度限制模块

简介

该模块的功能是输出值变化的速度进行限制。

当 SET 为 ON 时，Y=YSET；

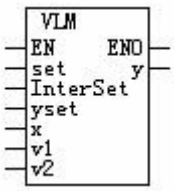
当 SET 为 OFF 时，若 X 增大，Y 按照 V1 定义的速度增大到 X；若 X 减小，Y 按照 V2 定义的速度减小到 X。

EN 和 ENO 能作为附加参数加以设置。

当 InterSet 为 ON 时，采用积分处理方法。

表示

符号



参数描述

参数	数据类型	含义
SET	BOOL	置位开关
InterSet	BOOL	积分开关
yset	SFLOAT	设定值
yset	SFLOAT	输入值
x	SFLOAT	上升速度(以秒位单位)
V2	SFLOAT	下降速度(以秒位单位)
Y	SFLOAT	输出值

实现方法

如果加了采用积分处理方法，精度比较高，对于上升和下降速度很小的情况下比较适合，但是速度比较大还是不要采用积分，以提高运行速度。

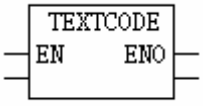
6. TEXTCODE 模块

简介

该模块的功能是使用 ST 语言来实现一段自定义程序。

表示

符号



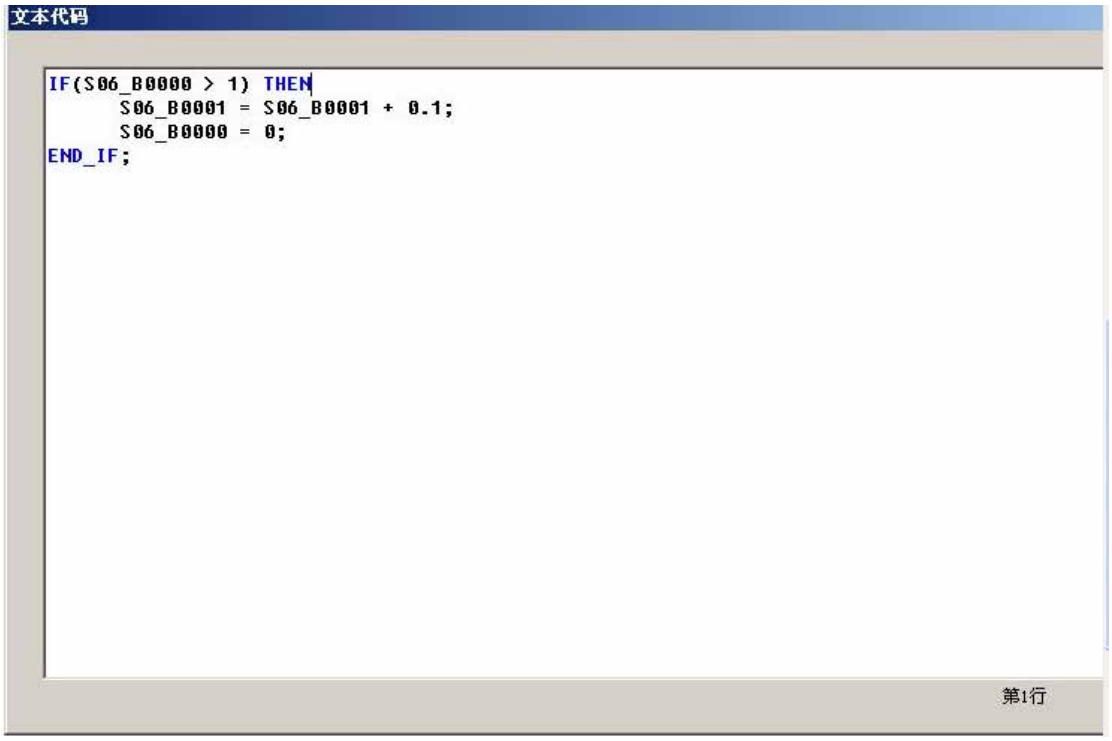
参数描述

参数	数据类型	含义
IN1	BOOL	保留
OUT1	BOOL	保留

使用方法

双击该模块便可出现下图

点击“ 代码 ”按钮，便可进入文本代码编辑区。



用户可在这个区域键入 ST 程序。它可以对系统中的各种资源、位号、变量、标准功能块、标准函数等等进行引用或操作。

用户可以在当前工程中，选择标记 TEXTCODE 代码中的变量或位号。如果在文本代码模块中（代码）查找到对应的对象（变量、位号），则在查找结果输出框中，显示相应查找到对象所在的行数。

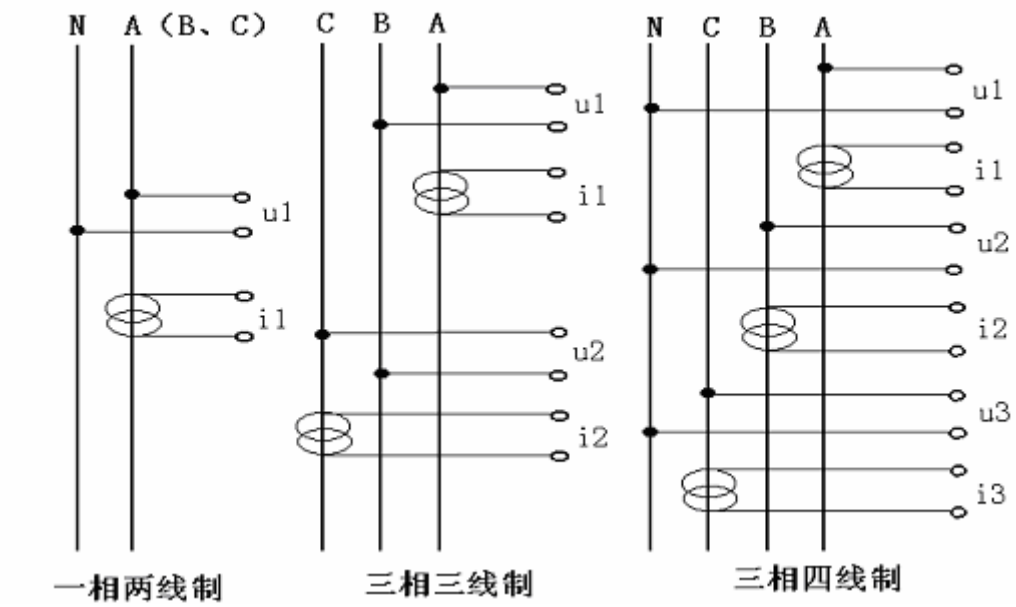
电量转换

电量处理模块处理 FW355 卡所采集的信号，它主要是根据交流电压有效值 V_{RMS}、交流电流有效值 I_{RMS} 以及有功功率 P 计算出视在功率 S、功率因数 λ、无功功率 Q、有功电能 E、无功电能 En。

视在功率 S、功率因数λ、无功功率 Q、有功电能 E、无功电能 En 是通过如下相关基本公式计算得出：

- 视在功率 $S=V_{RMS}*I_{RMS}$
- 功率因数λ= P/S= cosφ
- 无功功率 Q=S* sinφ
- 有功电能 E=P*Σt　　t 为单位累计时间
- 无功电能 E_N=Q*Σt　　t 为单位累计时间

关于用电线制



✧ 对一相二线制：u1 表示 A 相（或 B 相、或 C 相）与中性线 N 之间的电压；
i1 表示 A 相（或 B 相、或 C 相）电流；
任一根相线（或 A 相、或 B 相、或 C 相）与一根中性线组合成为**一组**一相两线制；

✧ 对三相三线制：u1、u2 分别表示 A 相与 B 相间、C 相与 B 相间的电压；
i1、i2 分别表示 A 相、C 相电流；
（注：本例以 B 相为参考，也可以 A 相或 C 相为参考）
三根相线 A 相、B 相和 C 相组合成为**一组**三相三线制；

✧ 对三相四线制：u1、u2、u3 分别表示 A 相、B 相、C 相与中性线 N 之间的电压；
i1、i2、i3 分别表示 A 相、B 相、C 相电流；
三根相线（A 相、B 相和 C 相）与一根中性线组合成为**一组**三相四线制；

- ACMETER12
- ACMETER33
- ACMETER34

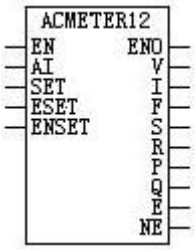
ACMETER12

简介

该模块用于对电量卡 FW355 中一相二线制的操作，计算其电压、电流、功率等值。

表示

符号



参数描述

参数	数据类型	含义
AI	structAI	电量信号位号
SET	BOOL	累积设置标志
ESET	structAccum	有功电能累积
ENSET	structAccum	无功电能累积
V	SFLOAT	交流电压有效值
I	SFLOAT	交流电流有效值
F	SFLOAT	频率
S	SFLOAT	视在功率
R	SFLOAT	功率因数
P	SFLOAT	有功功率
Q	SFLOAT	无功功率
E	structAccum	有功电能
NE	structAccum	无功电能

实现方法

当 SET=ON 时，有功电能值等于 ESET，无功电能 NE 等于 ENSET。

对一相二线制信号而言，FW355 卡件对**每组**一相二线制信号采样 4 个关联信号（电压有效值 Vrms、电流有效值 Irms、有功功率 P、频率 F），如下：

电压有效值 Vrms
电流有效值 Irms
有功功率 P
频率 F

------>计算产生 5 个信号

无功功率 Q
视在功率 S
有功电能 E
无功电能 EN
功率因数 λ

计算产生的 5 个信号为：无功功率 Q、视在功率 S、有功电能 E、无功电能 E_N、功率因数 λ，这种应用场合每个 FW355 卡最多采样**四组**一相二线制，算法如下：

S=Vrms×Irms

λ=P÷S= P÷（Vrms×Irms）

$$Q = S \times \sqrt{(1-\lambda_2)} = \sqrt{((Vrms \times Irms)2 - P2)}$$

$$E = P \times \sum t$$

$$E_N = Q \times \sum t$$

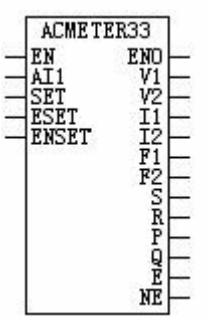
ACMETER33

简介

该模块用于对电量卡 FW355 中三相三线制的操作，计算其电压、电流、功率等值。

表示

符号



参数描述

参数	数据类型	含义
AI1	structAI	电量第一通道信号位号
SET	BOOL	累积设置标志
ESET	structAccum	有功电能累积
ENSET	structAccum	无功电能累积
V1	SFLOAT	交流电压有效值 1
V2	SFLOAT	交流电压有效值 2
I1	SFLOAT	交流电流有效值 1
I2	SFLOAT	交流电流有效值 2
F1	SFLOAT	频率 1
F2	SFLOAT	频率 2
S	SFLOAT	视在功率
R	SFLOAT	功率因数
P	SFLOAT	有功功率
Q	SFLOAT	无功功率
E	structAccum	有功电能
NE	structAccum	无功电能

实现方法

当 SET=ON 时，有功电能值等于 ESET，无功电能 NE 等于 ENSET。

对三相三线制信号而言，FW355 卡件对每组三相三线制信号采样 8 个关联信号（电压有效值 Vrms1、Vrms2，电流有效值 Irms1、Irms2，有功功率 P1、P2，频率 F1、F2），如下：

电压有效值 Vrms1、Vrms2
电流有效值 Irms1、Irms2
有功功率 P1、P2

频率 F1、F2

------>计算产生 6 个信号

有功功率 P
无功功率 Q
视在功率 S
有功电能 E
无功电能 EN
功率因数 λ

计算产生产生的 6 个信号为： 有功功率 P、无功功率 Q、视在功率 S、有功电能 E、无功电能 E_N、功率因数 λ，这种应用场合每个 FW355 卡最多采样两组三相三线制，算法如下：

S=S1+S2=V1rms×I1rms+ V2rms×I2rms

P=P1+P2

λ=P÷S= （P1+P2）÷（V1rms×I1rms+ V2rms×I2rms）

$$Q = S \times \sqrt{(1-\lambda_2)} = \sqrt{((V1rms \times I1rms + V2rms \times I2rms)2 - (P1 + P2)2)}$$

E =P×∑t

EN =Q×∑t

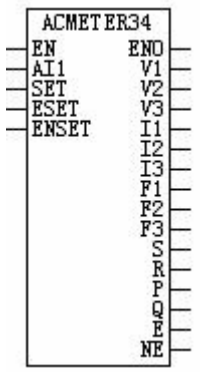
ACMETER34

简介

该模块用于对电量卡 FW355 中三相四线制的操作，计算其电压、电流、功率等值。

表示

符号



参数描述

参数	数据类型	含义
AI1	structAI	电量第一通道信号位号
SET	BOOL	累积设置标志
ESET	structAccum	有功电能累积
ENSET	structAccum	无功电能累积
V1	SFLOAT	交流电压有效值 1
V2	SFLOAT	交流电压有效值 2
V3	SFLOAT	交流电压有效值 3
I1	SFLOAT	交流电流有效值 1
I2	SFLOAT	交流电流有效值 2
I3	SFLOAT	交流电流有效值 3

F1	SFLOAT	频率 1
F2	SFLOAT	频率 2
F3	SFLOAT	频率 3
S	SFLOAT	视在功率
R	SFLOAT	功率因数
P	SFLOAT	有功功率
Q	SFLOAT	无功功率
E	structAccum	有功电能
NE	structAccum	无功电能

实现方法

当 SET=ON 时，有功电能值等于 ESET，无功电能 NE 等于 ENSET。

对三相四线制信号而言，FW355 卡件对每组三相四线制信号采样 12 个关联信号（电压有效值 Vrms1、Vrms2、Vrms3，电流有效值 Irms1、Irms2、Irms3，有功功率 P1、P2、P3，频率 F1、F2、F3），如下：

电压有效值 Vrms1、Vrms2、Vrms3
电流有效值 Irms 1、Irms 2、Irms3
有功功率 P1、P2、P3
频率 F1、F2、F3

------>计算产生 6 个信号

有功功率 P
无功功率 Q
视在功率 S
有功电能 E
无功电能 EN
功率因数 λ

计算产生产生的 6 个信号为：有功功率 P、无功功率 Q、视在功率 S、有功电能 E、无功电能 EN、功率因数 λ，这种应用场合每个 FW355 卡最多采样一组三相四线制，算法如下：

S=S1+S2+S3=V1rms×I1rms+ V2rms×I2rms+ V3rms×I3rms

P=P1+P2+P3

λ=P÷S= （P1+P2+P3）÷（V1rms×I1rms+ V2rms×I2rms+ V3rms×I3rm

$$Q = S \times \sqrt{(1-\lambda_2)} = \sqrt{((V1rms \times I1rms + V2rms \times I2rms + V3rms \times I3rms)2 - (P1 + P2 + P3)2)}$$

E =P×∑t

EN =Q×∑t

7. 信号选择模块

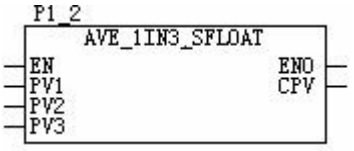
三选一信号平均选择模块（AVE_1IN3_SFLOAT）

简介

该模块输出从同一信号源获得的三个信号的平均值。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

$$\text{输出值}\ CPV = \frac{PV1 + PV2 + PV3}{3}$$

参数描述

参数	数据类型	含义
PV1	SFLOAT	输入信号 1
PV2	SFLOAT	输入信号 2
PV3	SFLOAT	输入信号 3
CPV	SFLOAT	输出值

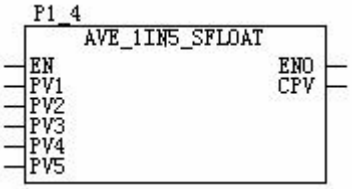
五选一信号平均选择模块（AVE_1IN5_SFLOAT）

简介

该模块输出从同一信号源获得的五个信号的平均值。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

$$\text{输出值}\ \ CPV = \frac{PV1 + PV2 + PV3 + PV4 + PV5}{5}$$

参数描述

参数	数据类型	含义
PV1	SFLOAT	输入信号 1
PV2	SFLOAT	输入信号 2
PV3	SFLOAT	输入信号 3
PV4	SFLOAT	输入信号 4
PV5	SFLOAT	输入信号 5
CPV	SFLOAT	输出值

三选一开关信号选择模块

简介

输入三个开关量信号，输出信号与输入开关信号中较多的一种保持一致，如下表所示：

信号中“ON”的个数	0	1	2	3
信号中“OFF”的个数	3	2	1	0
输出值 CPV	OFF	OFF	ON	ON

表示

符号

OUT=SEL_1IN3(PV1,PV2,PV3)

参数描述

参数	数据类型	含义
PV1	BOOL	开关量输入 1
PV2	BOOL	开关量输入 2
PV3	BOOL	开关量输入 3
OUT	BOOL	输出值

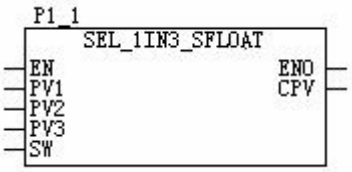
三选一模拟信号选择模块（SEL_1IN3_SFLOAT）

简介

根据选择开关的位置，选择从同一信号源获得的三个信号中的一个，赋值给输出。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

当选择开关 SW=1，选择输入信号 1 输出，即 CPV=PV1。

当选择开关 SW=2，选择输入信号 2 输出，即 CPV=PV2。

当选择开关 SW=3，选择输入信号 3 输出，即 CPV=PV3。

当选择开关 SW 不是上面的三个值，模块不做任何处理，保持上一周期值不变。

参数描述

参数	数据类型	含义
PV1	SFLOAT	输入信号 1
PV2	SFLOAT	输入信号 2
PV3	SFLOAT	输入信号 3
SW	UNIT	信号选择开关
CPV	SFLOAT	输出值

五选一开关信号选择模块

简介

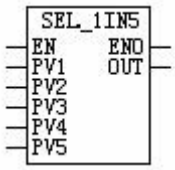
输入五个开关量信号,输出信号与输入的开关信号中较多的一种保持一致,如下表所示：

信号中“ON”的个数	0	1	2	3	4	5

信号中“OFF”的个数	5	4	3	2	1	0
输出值 CPV	OFF	OFF	OFF	ON	ON	ON

表示

符号



OUT=SEL_1IN5(PV1,PV2,PV3,PV4,PV5)

参数描述

参数	数据类型	含义
PV1	BOOL	开关量输入 1
PV2	BOOL	开关量输入 2
PV3	BOOL	开关量输入 3
PV4	BOOL	开关量输入 4
PV5	BOOL	开关量输入 5
OUT	BOOL	输出值

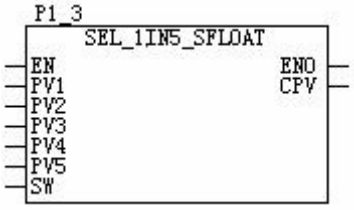
五选一模拟信号选择模块（SEL_1IN5_SFLOAT）

简介

根据选择开关的位置，选择从同一信号源获得的五个信号中的一个，赋给输出值。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

当选择开关 SW=1，选择输入信号 1 输出，即 CPV=PV1。

当选择开关 SW=2，选择输入信号 2 输出，即 CPV=PV2。

当选择开关 SW=3，选择输入信号 3 输出，即 CPV=PV3。

当选择开关 SW=4，选择输入信号 4 输出，即 CPV=PV4。

当选择开关 SW=5，选择输入信号 5 输出，即 CPV=PV5。

当选择开关 SW 不是上面的五个值，模块不做任何处理，保持上一周期值不变。

参数描述

参数	数据类型	含义
PV1	SFLOAT	输入信号 1
PV2	SFLOAT	输入信号 2

PV3	SFLOAT	输入信号 3
PV4	SFLOAT	输入信号 4
PV5	SFLOAT	输入信号 5
SW	UINT	信号选择开关
CPV	SFLOAT	输出值

三选二模块

简介

该模块有三个逻辑输入变量，输出与输入多的那种类型保持一致，如输入有两个或两个以上为 ON，则输出为 ON，反之为 OFF。

EN 和 ENO 作为附加参数加以设置。

表示

符号

算法

先计算三个逻辑输入中为 ON 的个数，当个数大于等于 2 时，输出为 ON，否则输出为 OFF。

参数描述

参数	数据类型	含义
IN1	BOOL	输入值 1
IN2	BOOL	输入值 2
IN3	BOOL	输入值
OUT	BOOL	输出值

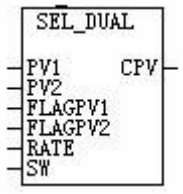
双信号选择开关（SEL_DUAL）

简介

从同一过程信号源通过两条不同的信号途径获得的两个输入信号中，通过该模块运用冗余配置自动选择一路信号输出。

EN 和 ENO 能作为附加参数加以设置。

表示



算法

当信号选择开关 SW=1 时，选择输入信号 1 输出，即 CPV=PV1。

当信号选择开关 SW=2 时，选择输入信号 2 输出，即 CPV=PV2。

当信号选择开关 SW=3 时，对两个输入信号的质量码和变化率进行判断，选择相应的信

号输出：

（1）若两个输入信号都是坏点，即它们的质量码是 0x0100（信号可疑）或 0x0800（信号故障），输出保持上一周期的值。

（2）若输入信号 1 为坏点，输入信号 2 为好点：

如果输入信号 2 的变化率没有超限，则选择输入信号 2 输出，即 CPV=PV2。

如果输入信号 2 的变化率超限，则输出保持上一个周期的值。

（3）若输入信号 2 为坏点，输入信号 1 为好点：

如果输入信号 1 的变化率没有超限，则选择输入信号 1 输出，即 CPV=PV1。

如果输入信号 1 的变化率超限，则输出保持上一个周期的值。

（4）若两个输入信号都是好点：

如果两个输入信号的变化率都超限了，则输出保持上一周期的值。

如果输入信号 2 的变化率超限了，则选择输入信号 1 输出，即 CPV=PV1。

如果输入信号 1 的变化率超限了，则选择输入信号 2 输出，即 CPV=PV2。

如果两个输入信号都没有超限，则选择输入信号 1 输出，即 CPV=PV1。

参数描述

参数	数据类型	含义
PV1	SFLOAT	输入信号 1
PV2	SFLOAT	输入信号 2
FLAGPV1	WORD	输入信号 1 质量码
FLAGPV2	WORD	输入信号 2 质量码
RATE	SFLOAT	变化率设定限（每秒的变化率）
SW	UINT	信号选择开关
CPV	SFLOAT	输出值

8. 浮点处理

浮点型死区模块

简介

该模块设置一个死区[－DB，＋DB]，对输入进行死区处理。

注意：死区参数 DB 总是设为正数。

表示

符号

算法

当输入 IN 在死区内，输出为 OUT=0；

当输入 IN >= DB，输出 OUT=（IN－DB）*K1；

当输入 IN <= －DB，输出 OUT=（IN＋DB）*K2；

参数描述

参数	数据类型	含义

IN	FLOAT	输入
DB	FLOAT	死区
K1	FLOAT	上升比例系数
K2	FLOAT	下降比例系数
OUT	FLOAT	输出值

偏差报警模块

简介

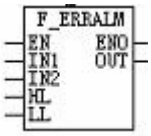
该模块的功能是判断两个输入值的偏差是否在规定的上下限内，通常输入 IN2 是设定值，输入 IN1 是需要和这个设定值比较的输入值。若两者的偏差在规定的上下限内时，输出值为 OFF；若两者偏差超出规定的上下限时，输出值为 ON。

EN 和 ENO 作为附加参数加以设置。

注意：输入的上下偏差限都用正数，即如果想设下限为-5，只要设 LL＝5 即可。

表示

符号



算法

当－LL≤IN1-IN2≤HL 时，输出 OUT＝OFF。

当 IN1－IN2>HL 或 IN1－IN2<-LL 时，输出 OUT＝ON。

参数描述

参数	数据类型	含义
IN1	FLOAT	输入值 1
IN2	FLOAT	输入值 2
HL	FLOAT	上偏差限（正值）
LL	FLOAT	下偏差限（正值）
OUT	BOOL	报警输出值

折线表处理模块

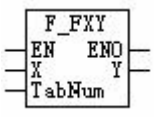
简介

该模块对输入进行二维折线表插值处理，输出插值处理后的值。

EN 和 ENO 作为附加参数加以设置。

表示

符号



算法

当输入 X 的值不大于折线表 X 坐标的最小值（即 X0）时，输出 Y 为 Y0 的值。

当输入 X 的值不小于折线表 X 坐标的最大值（即 X10）时，输出 Y 为 Y10 的值。

当输入 X 的值在折线表 X 坐标的最小值和最大值之间（即（X0,X10））时，先判断 X 落在什么区间，如当输入 X 落在 X[i]和 X[i+1]之间时，折线表按下面的插值公式对输入进行处理：

$$Y = Y[i] + \frac{Y[i+1] - Y[i]}{X[i+1] - X[i]} * (X - X[i])$$

参数描述

参数	数据类型	含义
TabNum	INT	折线表序号[0，31]
X0～X10	FLOAT	X 坐标点设定值
Y0～Y10	FLOAT	Y 坐标点设定值

FLOAT 型折线表获取模块

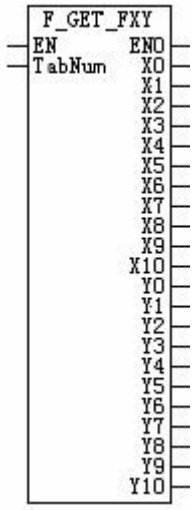
简介

该模块用来获取指定 X 和 Y 坐标的相应的折线表数据，参见上面的折线表设置模块。

EN 和 ENO 作为附加参数加以设置。

表示

符号



参数描述

参数	数据类型	含义
TabNum	INT	折线表序号[0，31]
X0～X10	FLOAT	X 坐标点设定值
Y0～Y10	FLOAT	Y 坐标点设定值

上下限报警模块

简介

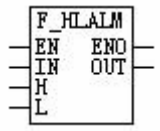
该模块的功能是判断输入值是否在规定的上下限之间，并在超限时给出相应的报警。当输入值大于上限时，输出值为 ON；当输入值小于下限时，输出值为 OFF；当输入值在上下限之间时，输出值保持上个周期的值不变。

EN 和 ENO 能作为附加参数加以设置。

注意：设定的上限值不要小于下限值，否则，该功能块不做任何处理就直接返回。

表示

符号



算法

当 IN > H, OUT = ON。

当 IN < L, OUT = OFF。

当 L≤IN≤H, OUT 保持上一周期的值不变。

参数描述

参数	数据类型	含义
IN	FLOAT	输入值
H	FLOAT	上限值
L	FLOAT	下限值
OUT	BOOL	报警输出值

浮点型积分限幅模块

简介

输出是对输入的积分，并且对输出进行限幅处理，当输出值超过设定的高低限，就会被限幅，同时置相应的报警标志位。

表示

符号

算法

当 SET＝ON 时，输出 Y＝YSET，如果 Y>=MAX，YMAX=ON，Y=MAX，否则 YMAX=OFF；如果 Y<=MIN，YMIN＝ON，Y＝MIN，否则 YMIN＝OFF。

当 SET=OFF 时，输出 Y 的值是对输入 X 积分，采样周期以 1ms 为单位；其中传递函数和离散化算式与积分不限幅模块相同。

如果 Y>＝MAX，则 Y=MAX，且 YMAX=ON，否则 Y 输出对输入 X 积分运算的值，YMAX＝OFF。

如果 Y<=MIN，则 Y＝MIN，且 YMIN＝ON，否则 Y 输出对输入 X 积分运算的值，YMIN＝OFF。

参数描述

参数	数据类型	含义
SET	BOOL	置位开关
YSET	FLOAT	设定值
X	FLOAT	输入值
GAIN	FLOAT	增益

MAX	FLOAT	高限
MIN	FLOAT	低限
YMAX	BOOL	高限报警开关
YMIN	BOOL	低限报警开关
Y	FLOAT	输出值

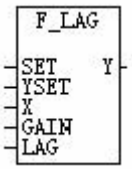
浮点型一阶滞后模块

简介

对输入信号进行一阶滞后处理，其中滞后时间常数以 0.1S 为单位，就是输入 LAG=1，表示滞后时间为 0.1 秒。

表示

符号



算法

当置位开关 SET=ON，输出 Y＝YSET。

当置位开关 SET＝OFF，输出 Y 是输入的一阶滞后输出，滞后时间 LAG 以 0.1S 为单位。

传递函数为：

$$G(S)=\frac{GAIN}{1+LAG\times 3}$$

即

$$\frac{Y(S)}{X(S)}=\frac{GAIN}{1+LAG\times 3}$$

化简后得：

$$Y(S)+LAG\times S\times Y(S)=GAIN\times X(S)$$

化到时域并差分化得：

$$Y(k)=LAG\times\frac{Y(k)-Y(k-1)}{dt}=GAIN\times X(k)$$

进一步化简得：

$$Y(k)=Y(k-1)+\frac{GAIN\times X(k)-Y(k-1)}{dt+LAG}\times dt$$

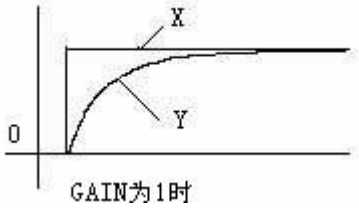
程序中 X(k)的地方取了 $\frac{X(k)+X(k-1)}{2}$，进行平均值滤波；

即最终公式：

$$Y(k)=Y(k-1)+\frac{GAIN\times\frac{X(k)+X(k-1)}{2}-Y(k-1)}{dt+LAG}\times dt$$

传递函数的时域的阶跃响应表达式为：$Y=GAIN*(1-e^{\frac{-t}{LAG}})$

响应曲线如下：



参数描述

参数	数据类型	含义
SET	BOOL	置位开关
YSET	FLOAT	设定值
X	FLOAT	输入值
GAIN	FLOAT	增益
LAG	FLOAT	滞后时间常数（0.1s）
Y	FLOAT	输出

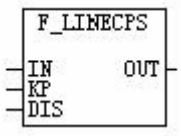
浮点型线性补偿模块

简介

根据输入的比例系数 Kp 和偏移量 DIS 对输入 IN 进行线性补偿，并输出补偿后的结果。

表示

符号



算法

根据下列公式计算补偿结果：

OUT=IN×Kp＋DIS

参数描述

参数	数据类型	含义
IN	FLOAT	输入变量
KP	FLOAT	比例系数
DIS	FLOAT	偏移量
OUT	FLOAT	输出变量

FLOAT 型折线表设置模块

简介

FLOAT 型折线表都是二维折线表，都是 10 段的，X 和 Y 的坐标点各 11 个，一个控制站最多 32 张，每张分配 128 个字节，前四个字节为折线表类型，接着存放 88 个字节的 X 和 Y 坐标值，其余的为保留字节。

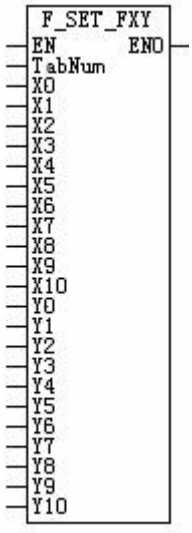
注意：输入的折线表序号在[0，31]之间，输入的 X 轴坐标点要从小到大依次赋给 X0～X10，Y 轴坐标点没有此项限制。

该模块用来设置指定序号的折线表的 X 和 Y 坐标值。

EN 和 ENO 作为附加参数加以设置。

表示

符号



参数描述

参数	数据类型	含义
TabNum	INT	折线表序号[0，31]
X0～X10	FLOAT	X 坐标点设定值（依次从小到大）
Y0～Y10	FLOAT	Y 坐标点设定值

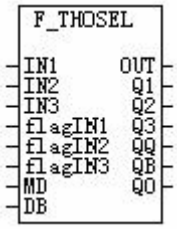
浮点型三选一模块

简介

根据工作方式参数 MD 以及两个输入的输入品质 flagIN1、flagIN2 来选择输出方式以及输出的品质。输入信号的品质码为 ON 时，认为是坏点，品质码为 OFF 时，认为是好点。

表示

符号



算法

当工作方式参数 MD 等于 4 或 5 或 6，则输出 OUT 的值相应的等于 IN1 或 IN2 或 IN3，如果对应的输入点为坏点，则输出保持上一个周期的值不变，品质输出 QO 置为逻辑 1。

当工作方式参数 MD 等于 0、1、2、3 时，则有

1) MD 等于 0 时，选择三个输入中数值居中的作为输出 OUT 的值。

2) MD 等于 1 时，选择三个输入的平均值作为输出 OUT 的值。

3) MD 等于 2 时，选择三个输入中数值最小的作为输出 OUT 的值。

4) MD 等于 3 时，选择三个输入中数值最大的作为输出 OUT 的值。

此时，还要根据相应的输入品质来调整输出：

1）如果三个输入都是坏点，则输出保持上一个周期的值，品质输出 QO 置为逻辑 1。

2）如果有两个输入是坏点，则输出剩余的一个好点的值。

3）如果只有一个输入是坏点，则根据另外两个输入好点之间的偏差来选择输出。若两个好点之间的偏差不超过指定的偏差上限值 DB，则输出两个好点的平均值，否则输出保持上一个周期的值，品质输出 QO 置为逻辑 1。

4）如果三个输入都是好点，则根据三点之间的三对偏差是否超过 DB 作如下的选择：

i. 三对偏差都不超过 DB 时，则输出根据 MD 所选的方式输出。

ii. 只有一个偏差超过 DB 时，则选择三个输入中数值居中的值作为输出 OUT 的值。

iii. 若有两对偏差超过 DB 时，则取偏差不超限的一对输入值的平均值作为输出 OUT 的值。

iv. 若三对偏差都超过 DB 时，则输出保持上一个周期的值，品质输出 QO 置为逻辑 1。

品质输出

1) 如果 IN1 是坏点，则相应的输出品质 Q1 置为逻辑 1；否则输出品质 Q1 置为逻辑 0。

2) 如果 IN2 是坏点，则相应的输出品质 Q2 置为逻辑 1；否则输出品质 Q1 置为逻辑 0。

3) 如果 IN3 是坏点，则相应的输出品质 Q3 置为逻辑 1；否则输出品质 Q1 置为逻辑 0。

4) IN1、IN2、IN3 中任一个为坏点时，则品质输出 QQ 为逻辑 1；反之为逻辑 0。

5) 如果工作方式参数 MD 等于 4 或 5 或 6，且相应的 IN1 或 IN2 或 IN3 为坏点时，品质输出 QO 置为逻辑 1。

参数描述

参数	数据类型	含义
IN1、IN2、IN3	FLOAT	输入
flagIN1、flagIN2、flagIN3	BOOL	输入数据的品质
MD	INT	选择输出方式
DB	FLOAT	偏差上限值
OUT	FLOAT	输出值
Q1、Q2、Q3	BOOL	相应输入量的品质输出
QQ	BOOL	品质总报警
QB	BOOL	偏差报警
QO	BOOL	品质输出

应用

主要用于需要在三个输入中选择适当的值作为输出，选择的条件包括：输入的品质（质量码）以及输入之间的偏差。选择三个输入中的最优值作为控制参数，适用于对输入精度要求高并且对随机扰动敏感的控制过程中。

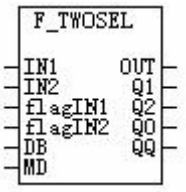
浮点型二选一模块

简介

根据工作方式参数 MD 以及两个输入的输入品质 flagIN1、flagIN2 来选择输出方式以及输出的品质。输入信号的品质码为 ON 时，认为是坏点，品质码为 OFF 时，认为是好点。

表示

符号



算法

当工作方式参数 MD 等于 3 或 4 时，则输出 OUT 的值相应的等于 IN1 或 IN2。如果对应的输入点的品质为“坏”时，则品质输出 QO 置为逻辑 1，输出保持上一个周期的值。

当工作方式参数 MD 等于 0、1、2 时，则有：

- 如果 MD=0，输出为两个输入值的平均值；
- 如果 MD=1，输出为两个输入值中的较小值；
- 如果 MD=2，输出为两个输入中的较大值；

此时，还要根据相应的输入品质来调整输出：

- 如果两个输入都是坏点，品质输出 QO 置为逻辑 1，输出保持上一个周期的值；
- 如果一个点为坏点，另一个为好点，则选择好点作为输出；
- 如果两个输入都是好点，则根据两者之间的偏差是否超过偏差设定值 DB，进行如下选择：

若偏差越限，品质输出 QO 为逻辑 1，则输出保持上一个周期的值；

若偏差没有越限，品质输出 QO 为逻辑 0，输出工作方式的值。

品质输出

如果输入点（IN1、IN2）为坏点，则相应的输出品质（Q1、Q2）为逻辑 1；

如果两个输入任何一个为坏点，则品质总输出 QQ 为逻辑 1。

参数描述

参数	数据类型	含义
IN1	FLOAT	输入 1
IN2	FLOAT	输入 2
flagIN1	BOOL	输入 1 的品质
flagIN2	BOOL	输入 2 的品质
MD	INT	选择输出方式
DB	FLOAT	偏差上限值
OUT	FLOAT	输出值
Q1	BOOL	IN1 的品质输出
Q2	BOOL	IN2 的品质输出
QQ	BOOL	品质总报警
QO	BOOL	品质输出

增减限幅模块

简介

该模块的功能是根据工作方式选择不同方式的输出，当工作在增减限幅的工作方式下时，根据命令参数 DN 和 UP 的设置，对输出值按照给定的速率进行增输出或是减输出。

EN 和 ENO 能作为附加参数加以设置。

注意：用户设定限幅上限应该大于下限，否则，该功能块不做任何处理就返回。

表示

符号

算法

当 MD＝ON,为跟踪方式,即输出等于输入（OUT=IN）。

当 MD=OFF,为增减输出方式,根据 DN 和 UP 的设置值,选择增输出还是减输出：

若 DN=OFF,UP＝ON 时,OUT＝OUT＋RATE,输出按照给定的速率 RATE 递增,直到输出被限制在上限值。

若 UP=OFF,DN＝ON 时,OUT＝OUT－RATE,输出按照给定的速率 RATE 递减,直到输出被限制在下限值。

若 UP=ON,DN=ON 时,则认为是错误命令,输出报警 ALM 为 ON,输出值保持上个周期的值。

若 UP=OFF,DN=OFF 时,输出保持上一周期的值不变,不做增减操作。

另外，不管是跟踪方式还是增减输出方式，最后都对输出值进行限幅处理：

若 OUT≥HL,输出上限限幅，OUT=HL。

若 OUT≤LL ,输出下限限幅，OUT=LL。

参数描述

参数	数据类型	含义
IN	FLOAT	输入值
UP	BOOL	增指令
DN	BOOL	减指令
MD	BOOL	工作方式
RATE	FLOAT	变化率
HL	FLOAT	上限值
LL	FLOAT	下限值
OUT	FLOAT	输出值
ALM	BOOL	错误指令报警输出

FLOAT 型速度限幅模块

简介

该模块可以通过开关 MD 来选择阶跃处理还是平滑处理。

注意：

当工作在阶跃处理时，T 设置了阶跃间隔时间，以 0.1S 为单位。

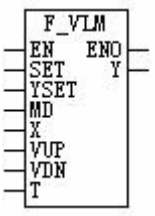
当工作在平滑处理时，T 为速度单位，以 1S 为单位，例如：若 VUP=10，T=2，则增速度为 5/S。T 的值必须大于 0，否则该功能块不会做任何处理，包括手动置值。

EN 和 ENO 作为附加参数加以设置。

注意：T 的值必须大于 0，否则，该功能块不做任何处理就直接返回。

表示

符号



算法

置位开关 SET＝ON 时，输出 Y=YSET，并对内部时间计数清 0，通常在该功能块运行前先使 SET＝OFF，YSET＝0，进行初始化。

置位开关 SET=OFF 时：

当 MD=ON 时，对输出进行阶跃处理，通过 T 设置阶跃间隔时间，以 0.1S 为单位。如果输入 X 递增，则输出 Y 按 VUP 定义的速度每隔 T 时间递增一次，直到和输入 X 相等；如果输入 X 递减，则输出 Y 按 VDN 定义的速度每隔 T 时间递增一次，直到和输入 X 相等。

当 MD=OFF 时，对输出进行平滑处理。如果输入 X 递增，则把 VUP 定义的速度转化成每个控制周期的速度，输出按照这个值每个周期递增，直到与输入 X 的值相等；如果输入 X 递减，则把 VDN 定义的速度转化成每个周期的速度，输出按照这个值每个周期递减，直到与输入 X 的值相等。

参数描述

参数	数据类型	含义
SET	BOOL	置位开关
YSET	FLOAT	设定值
MD	BOOL	工作方式参数
X	FLOAT	输入
VUP	FLOAT	增速度
UDN	FLOAT	减速度
T	ULONG	阶跃处理时：阶跃时间间隔，0.1S 为单位。平滑处理时：速度单位，1S 为单位。
Y	FLOAT	输出值

1.5.3 附加模块库

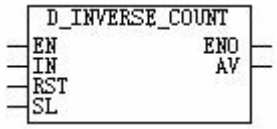
1. 特殊模块

开关量状态变化次数累积模块

简介

累积开关量出现上升沿或下降沿的次数。

表示



算法

(1) 当 RST 出现一个上升沿（从 0 到 1）时，输出 AV 清零

(2) 当 SL = 0 时，累积输入端 IN 上升沿（从 0 到 1）次数

（3） 当 SL = 1 时，累积输入端 IN 下降沿（从 1 到 0）次数

（4） 当 SL = 2 时，累积输入端 IN 变化次数（上升沿次数＋下降沿次数）

（5） 当 SL 为其他值时，效果同 SL = 0 时的状态

参数描述

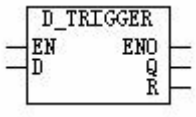
参数	数据类型	含义
IN	BOOL	待监测开关量值
RST	BOOL	复位信号
SL	UINT	统计类型
AV	FLOAT	累积结果当前值

D 触发器

简介

在输入的上升沿，给输出取反。

表示

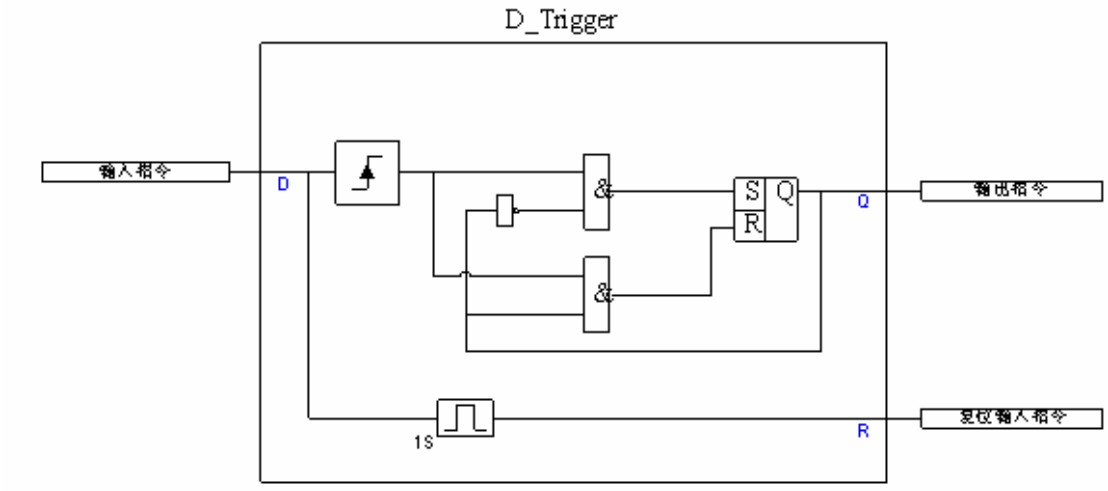


算法

（1） 注意：实际过程中 D 和 R 连接同一个位号。相当于一个常开按钮，当按钮按下（D = ON）以后过 1s 后弹起（D = OFF）。

（2） 在 D 的上升沿（D 从 OFF 跳变到 ON），Q 取反一次。同时产生脉宽为 1s 的脉冲 R。

逻辑示意图



实现D触发器的工作原理，输入指令每变化一次（上跳沿触发），输出则翻转一次。

参数描述

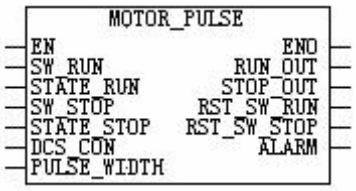
参数	数据类型	含义
D	BOOL	输入
Q	BOOL	输出
R	BOOL	复位输入

脉冲启停的二位式电机控制

简介

用于脉冲启停的普通二位式电机。当输入启动指令时产生一个指定脉宽的启动输出脉冲，当输入停止指令时产生一个指定脉宽的停止指令脉冲。同时根据输入的状态信号判断停车报警。

表示



算法

（1）注意：实际过程中启动指令 SW_RUN 和启指令复位信号 RST_SW_RUN 连接同一个位号，停止指令 SW_STOP 和停止指令复位信号 RST_SW_STOP 连接同一个位号。它相当于一个常开按钮，当按钮按下（ON）后过一段时间（PULSE_WIDTH）按钮自动弹起（OFF）。

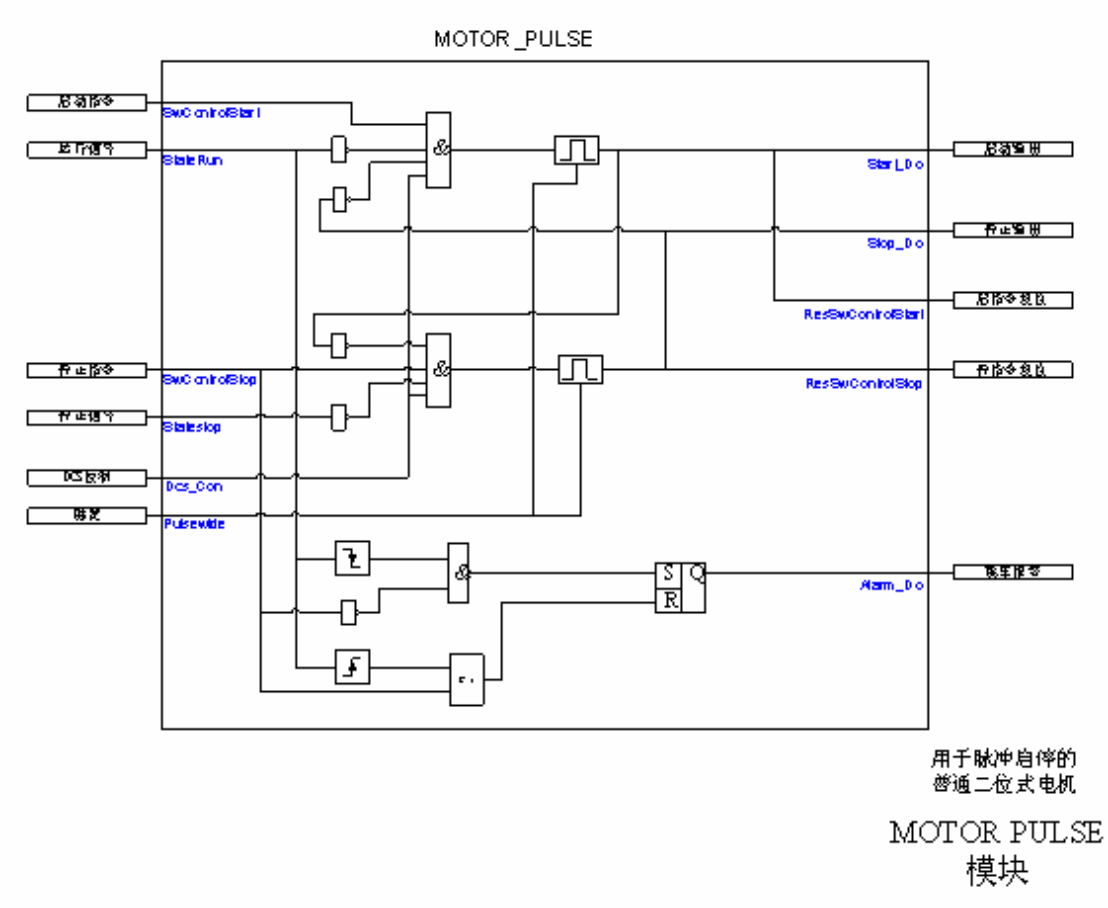
（2）在 DCS_CON = ON，以及没有运行信号（STATE_RUN= OFF）并且没有停止输出（STOP_OUT = OFF）的情况下，当启动指令 SW_RUN = ON 时，启动输出 RUN_OUT 和启指令复位信号 RST_SW_RUN 均为一个脉宽为 PULSE_WIDTH 毫秒的 ON 脉冲。也就是说，当 SW_RUN 从 OFF 跳变到 ON 的瞬间，RUN_OUT 和 RST_SW_RUN 均变为 ON，一直持续到 PULSE_WIDTH 毫秒以后,RUN_OUT 和 RST_SW_RUN 跳变为 OFF。

（3）在 DCS_CON = ON，以及没有停止信号（STATE_STOP = OFF）并且没有启动输出（RUN_OUT = OFF）的情况下，当停止指令 SW_STOP = ON 时，停止输出 STOP_OUT 和停止指令复位信号 RST_SW_STOP 均为一个脉宽为 PULSE_WIDTH 毫秒的 ON 脉冲。也就是说，当 SW_STOP 从 OFF 跳变到 ON 的瞬间，STOP_OUT 和 RST_SW_STOP 均变为 ON，一直持续到 PULSE_WIDTH 毫秒以后,STOP_OUT 和 RST_SW_STOP 跳变为 OFF。

（4）当在无停止指令（SW_STOP = OFF）的情况下，如果运行信号 STATE_RUN 从 ON 跳到 OFF 时，输出跳车报警（ALARM = ON）。

（5）当运行信号 STATE_RUN 从 OFF 跳到 ON 时，或者有停止指令（SW_STOP = ON）时，复位跳车报警（ALARM = OFF）。

逻辑示意图



参数描述

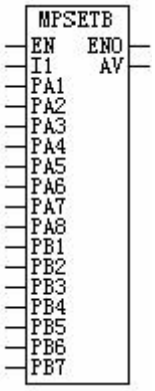
参数	数据类型	含义
SW_RUN	BOOL	启动指令
STATE_RUN	BOOL	运行指令
SW_STOP	BOOL	停止指令
STATE_STOP	BOOL	停止信号
DCS_CON	BOOL	DCS 控制
PULSE_WIDTH	ULONG	脉宽（以 1ms 位单位）
RUN_OUT	BOOL	启动输出
STOP_OUT	BOOL	停止输出
RST_SW_RUN	BOOL	启指令复位
RST_SW_STOP	BOOL	停指令复位
ALARM	BOOL	跳车报警

步参数赋值模块

简介

根据输入浮点数所处的范围确定输出值。

表示



算法

(1) 当 | I1 | ≤ PB1, AV = PA1

(2) 当 PB1 < | I1 | ≤ PB2, AV = PA2

(3) 当 PB2 < | I1 | ≤ PB3, AV = PA3

(4) 当 PB3 < | I1 | ≤ PB4, AV = PA4

(5) 当 PB4 < | I1 | ≤ PB5, AV = PA5

(6) 当 PB5 < | I1 | ≤ PB6, AV = PA6

(7) 当 PB6 < | I1 | ≤ PB7, AV = PA7

(8) 当 | I1 | > PB7, AV = PA8

(9) 注意：必须满足 0≤PB1 < PB2 < PB3 < PB4 < PB5 < PB6 < PB7，否则此功能块将不起作用

参数描述

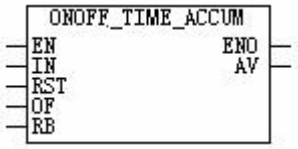
参数	数据类型	含义
I1	SFLOAT	输入
AV	SFLOAT	输出当前值
PA1～PA8	SFLOAT	可供选择的输出值
PB1～ PB7	SFLOAT	相对于每个输出值，输入值的选择范围

开关时间累积器模块

简介

对开关量的开状态或关状态的时间进行累积。

表示



算法

(1) 当 RB = 0 时，AV 输出 IN = 0 时的累积时间

(2) 当 RB = 1 时，AV 输出 IN = 1 时的累积时间

(3) 当 RST 为上升沿（从 0 到 1）时，AV = 0

(4) 当 OF = 0 时，AV 输出的时间以秒（s）为单位

(5) 当 OF = 1 时，AV 输出的时间以分钟（min）为单位

(6)　　当 OF = 2 时，AV 输出的时间以小时（h）为单位

(7)　　当 OF 为其他值时，AV 输出的时间均以秒（s）为单位

参数描述

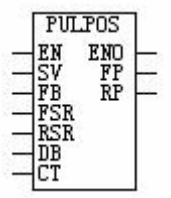
参数	数据类型	含义
IN	BOOL	待监测开关量值
RST	BOOL	复位信号
OF	UINT	输出时间格式
RB	BOOL	累积状态选择
AV	FLOAT	累积结果当前值

脉冲定位器 PULPOS

简介

脉冲定位器（Pulse Positioner）功能块比较两个输入信号 SV 和 FB，输出是持续时间正比于这两个输入偏差以及设定的行程时间的脉冲。两个输入之间的偏差被转换为一定时间的正向或反向的开关量输出。同时还提供了偏差死区和周期时间的设定。

表示



算法

(1) 设定值 SV 与反馈值 FB 之差称为偏差信号（取绝对值）。

(2) 偏差信号如果在死区范围之内（|SV － FB| ≤ DB），则不输出任何信号。

(3) 如果偏差信号|SV － FB| > DB：

a) 如果 FB 小于 SV，则产生正向输出（FP = ON）。正向输出的脉宽是由下式决定的：

$$\text{正向输出脉冲持续时间} = \frac{SV - FB}{FSR} \text{秒}$$

如果上式计算出的正向输出脉宽大于周期时间 CT，则正向输出脉宽 = CT。

b) 如果 FB 大于 SV 的量超过死区，则产生反向输出（RP = ON）。反向输出的脉宽是由下式决定的：

$$\text{反向输出脉冲持续时间} = \frac{FB - SV}{RSR} \text{秒}$$

如果上式计算出的反向输出脉宽大于周期时间 CT，则反向输出脉宽 = CT。

注意：上面的 a）、b）式计算出的脉宽值精度为 1 个控制周期。举例：假设计算出脉宽 = 1.5012 秒，则在 0.1 秒的控制周期下，实际脉宽为 1.6 秒；而在 0.5 秒的控制周期下，实际脉宽为 2 秒。

(4) 正向行程时间 FSR 表示的含义：当 SV>FB 并且偏差信号处于死区范围之外的时候每偏差 FSR，输出正向脉冲持续 1 秒。

(5) 反向行程时间 RSR 表示的含义：当 SV< FB 并且偏差信号处于死区范围之外的时候，

每偏差 RSR，输出反向脉冲持续 1 秒。

（6）周期时间 CT 设定功能块两次运算之间的时间。

参数描述

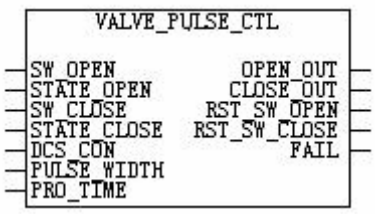
参数	数据类型	范围	含义
SV	SFLOAT	0～1	设定值
FB	SFLOAT	0～1	反馈值
FSR	SFLOAT	0～1	正向行程速率（s^{-1}）
RSR	SFLOAT	0～1	反向行程速率（s^{-1}）
DB	SFLOAT	0～1	绝对死区
CT	ULONG	0～4294967295	周期时间（以 1ms 位单位）
FP	BOOL	ON，OFF	正向脉冲输出
RP	BOOL	ON，OFF	反向脉冲输出

脉冲输出的二位式电动阀门控制

简介

用于脉冲输出的二位式电动阀门。当输入开指令时产生一个指定脉宽的开输出脉冲，当输入关指令时产生一个指定脉宽的关指令脉冲。同时根据阀门状态的反馈信号开到位和关到位以及行程时间来确定阀门是否故障。

表示



算法

（1）注意：实际过程中开指令 SW_OPEN 和开指令复位信号 RST_SW_OPEN 连接同一个位号，关指令 SW_CLOSE 和关指令复位信号 RST_SW_CLOSE 连接同一个位号。它相当于一个常开按钮，当按钮按下（ON）后过一段时间（PULSE_WIDTH）按钮自动弹起（OFF）。

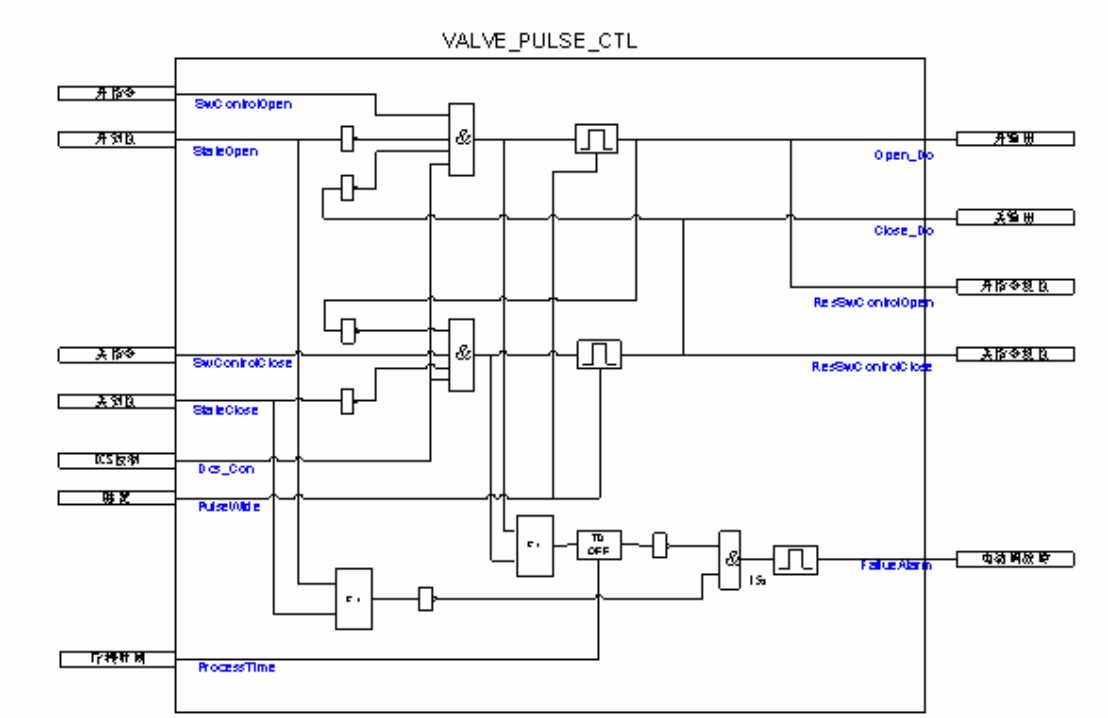
（2）在 DCS_CON = ON，以及没有开到位（STATE_OPEN = OFF）并且没有关输出（CLOSE_OUT = OFF）的情况下，当开指令 SW_OPEN = ON 时，开输出 OPEN_OUT 和开指令复位信号 RST_SW_OPEN 均为一个脉宽为 PULSE_WIDTH 毫秒的 ON 脉冲。也就是说，当 SW_OPEN 从 OFF 跳变到 ON 的瞬间，OPEN_OUT 和 RST_SW_OPEN 均变为 ON，一直持续到 PULSE_WIDTH 毫秒以后，OPEN_OUT 和 RST_SW_OPEN 跳变为 OFF。

（3）在 DCS_CON = ON，以及没有关到位（STATE_CLOSE = OFF）并且没有开输出（OPEN_OUT = OFF）的情况下，当关指令 SW_CLOSE = ON 时，关输出 CLOSE_OUT 和关指令复位信号 RST_SW_CLOSE 均为一个脉宽为 PULSE_WIDTH 毫秒的 ON 脉冲。也就是说，当 SW_CLOSE 从 OFF 跳变到 ON 的瞬间，CLOSE_OUT 和 RST_SW_CLOSE 均变为 ON，一直持续到 PULSE_WIDTH 毫秒以后，CLOSE_OUT 和 RST_SW_CLOSE 跳变为 OFF。

（4）从开输出 OPEN_OUT 的上升沿开始计时，直到收到反馈回来的开到位 STATE_OPEN 为 ON 为止，这个过程的时间如果大于等于设定的行程时间 PRO_TIME，则显示阀门故障，输出 FAIL 为一个脉宽为 15s 的脉冲。

（5）从关输出 CLOSE_OUT 的上升沿开始计时，直到收到反馈回来的关到位 STATE_CLOSE 为 ON 为止，这个过程的时间如果大于等于设定的行程时间 PRO_TIME，则显示阀门故障，输出 FAIL 为一个脉宽为 15s 的脉冲。

逻辑示意图



用于脉冲输出的
二位式电动阀
VALVE_PULSE_CTL
模块

参数描述

参数	数据类型	含义
SW_OPEN	BOOL	开指令
STATE_OPEN	BOOL	开到位
SW_CLOSE	BOOL	关指令
STATE_CLOSE	BOOL	关到位
DCS_CON	BOOL	DCS 控制
PULSE_WIDTH	ULONG	脉宽（以 1ms 为单位）
PRO_TIME	ULONG	行程时间（以 1ms 为单位）
OPEN_OUT	BOOL	开输出
CLOSE_OUT	BOOL	关输出
RST_SW_OPEN	BOOL	开指令复位（和开指令连接同一位号）
FAIL	BOOL	关指令复位（和关指令连接同一位号）

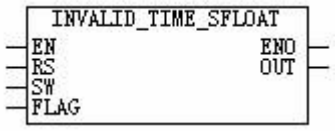
模拟量越限时间累积模块

简介

该模块对 AI 信号产生的质量码进行解析，将 AI 发生报警的时间进行累积。

表示

符号



参数描述

参数	数据类型	含义
RS	BOOL	当 RS＝ON 时对时间累积值清零
SW	UINT	当 SW＝0 时输出以秒为单位；当 SW＝1 时输出以分为单位；当 SW＝2 时输出以小时为单位，其他值则以秒为单位输出
FLAG	WORD	AI 模块质量码
OUT	FLOAT	累积时间，输出模块

2. 锅炉模块

链条炉燃烧自控模块 COAL_AND_WIND

简介

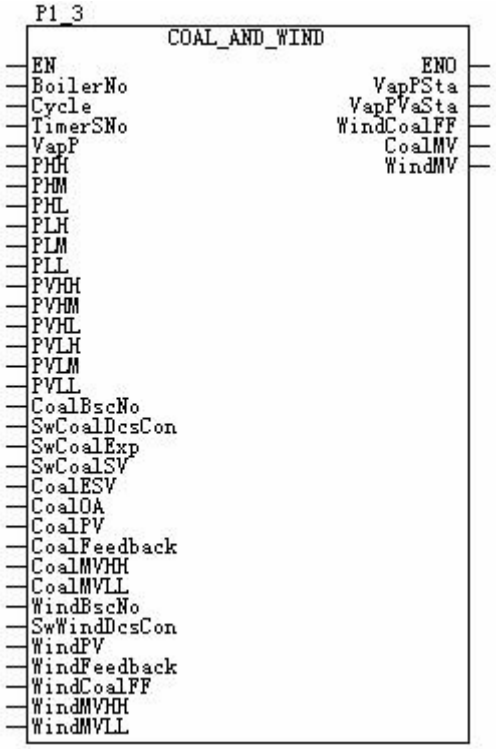
COAL_AND_WIND 功能块是链条炉给煤及鼓风机控制的专用控制模块，它融合了我公司在锅炉行业多年的工程经验。

给煤控制是保证锅炉安全运行的重要控制对象，我公司根据锅炉燃烧特点和影响汽包压力变化的因素，独创了给煤控制方案。该方案的核心是根据汽包压力状态和汽包压力变化趋势建立的 7*7（两维数组）的专家规则库，然后再根据负荷的变化趋势对专家规则库进行休整。

送风控制使燃料量与空气量相协调（风煤比），保证燃烧的经济性。结合操作经验，设定一合适的风煤比，将给煤输出与风煤比的乘积作为鼓风机的输出。

该功能块需要配合 SUPCON 链条炉燃烧控制的专用上位机软件使用，该软件主要进行专家规则库的管理，并具备对控制站专家数据的上传、下载、备份等功能。

表示



参数描述

参数	数据类型	含义
BoilerNo	UINT	控制模块序号，范围：0~7
Cycle	UINT	控制周期,单位：秒
TimerSNo	UINT	模块使用的秒定时器序号
VapP	SFLOAT	主蒸汽压力
PHH	SFLOAT	主蒸汽压力高高限
PHM	SFLOAT	主蒸汽压力高中限
PHL	SFLOAT	主蒸汽压力高低限
PLH	SFLOAT	主蒸汽压力低高限
PLM	SFLOAT	主蒸汽压力低中限
PLL	SFLOAT	主蒸汽压力低低限
PVHH	SFLOAT	主蒸汽压力变化量高高限 + 0.5
PVHM	SFLOAT	主蒸汽压力变化量高中限 + 0.5
PVHL	SFLOAT	主蒸汽压力变化量高低限 + 0.5
PVLH	SFLOAT	主蒸汽压力变化量低高限 + 0.5
PVLM	SFLOAT	主蒸汽压力变化量低中限 + 0.5
PVLL	SFLOAT	主蒸汽压力变化量低低限 + 0.5
CoalBscNo	UINT	给煤控制所使用的 BSC 回路序号
SwCoalDcsCon	BOOL	给煤控制是否采用 DCS 控制（ON：DCS 控制，OFF：硬手操）
SwCoalExp	BOOL	给煤控制是否采用专家控制方案（ON：专家控制）
SwCoalSV	BOOL	给煤控制是否外给定
CoalESV	SFLOAT	给煤量外给定值
CoalOA	SFLOAT	给煤控制输出补偿
CoalPV	SFLOAT	给煤量测量值
CoalFeedback	SFLOAT	给煤控制阀位反馈

CoalMVHH	SFLOAT	给煤控制输出上限
CoalMVLL	SFLOAT	给煤控制输出下限
WindBscNo	UINT	鼓风机控制所使用的 BSC 回路序号
SwWindDcsCon	BOOL	鼓风机控制是否采用 DCS 控制（ON：DCS 控制，OFF：硬手操）
WindPV	SFLOAT	鼓风量测量值
WindFeedback	SFLOAT	鼓风量阀位反馈
WindCoalFF	SFLOAT	风煤比系数
WindMVHH	SFLOAT	鼓风机控制输出上限
WindMVLL	SFLOAT	鼓风机控制输出下限

3. 造气模块

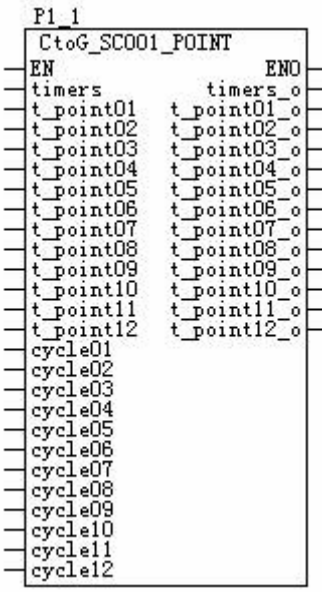
CtoG_SC001_POINT 指针赋值模块

简介

该模块是用于 12 台炉子的时间指针赋值。

EN 和 ENO 能作为附加参数加以设置。

表示



说明

(1) if(timers>0)　timers_o = 0；　　//定时器产生秒脉冲

(2) t_point01_o = t_point01_o +1；　　//1＃炉时间指针自累加

(3) if (t_point01_o>=cycle01) or (t_point_o<=0)，t_point01_o 初值被置为 0，并从 0 开始自加 1，直到等于 cycle01，然后又重新从 0 开始自加。

(4)如果 t_point01_o 值在 0 和 cycle01 之间(比如 cycle01 为 120,则取 t_point01_o＝60），t_point01_o 初值被置为 60，并从 0 开始自加 1，直到等于 cycle01，然后又重新从 0 开始自加。

(5) t_point02_o～t_point12_o 的赋值方法与 t_point01_o 相同。

参数描述（时间单位：秒）

参数	数据类型	含义	补充说明
timers	UINT	秒定时器	秒定时器，推荐位 timers[10],在不与其他程序冲突的前提

			下可以任选
t_point01～t_point12	INT	1～12＃炉时间指针	
cycle_01～cycle12	INT	1～12＃炉循环周期	
timers_o	UINT	秒定时器	
t_point01_o～t_point12_o	INT	1～12＃炉时间指针	

CtoG_SC002_INERTGAS 制气制惰切换模块

简介

该模块是用于控制 12 台炉子的制气制惰及富氧制气切换。

EN 和 ENO 能作为附加参数加以设置。

表示

说明

（1）working01~working12 任一个为 ON 时，对 oldinertgas2_ o 分别置为 ON 和 OFF，并对 inertgas_o 置值；inertgas_o 被置为 ON 和 OFF，inertgas_o 不能直接置值。

（2）如果 working01 ~ working12 任何一个为 ON（开炉时，不能进行制气/制惰切换），则

oldinertgas1_o = oldinertgas1;
oldinertgas2_o = oldinertgas2;
inertgas_o = oldinertgas2_o;

（3）如果 working01 ~ working12 任何一个为 OFF（停炉时，可以进行制气/制惰切换），则

inertgas_o = inertgas;
oldinertgas2_o = oldinertgas1_o;
oldinertgas1_o = inertgas_o;

（4）如果 inertgas 为 ON，则

gas_hy_ox01_o ～gas_hy_ox12_o 都为 OFF

（5）如果 working01 为 ON（运行时，不能进行富氧制气切换），则 gas_hy_ox01_o 保持原值不能切换。

（6）gas_hy_ox02_o ～gas_hy_ox12_o 与 working02～working12 的关系与上同。

参数描述

参数	数据类型	含义	补充说明
inertgas	BOOL	制惰指令	
oldinertgas1	BOOL	上一个周期的制惰指令	
oldinertgas2	BOOL	上二个周期的制惰指令	
gas_by_ox01～gas_by_ox12	BOOL	1～12＃炉富氧制气指令	
working01～working12	BOOL	1～12＃炉工作状态标志	
inertgas_o	BOOL	制惰指令	位号引用与第 1 输入管脚同
oldinertgas1_o	BOOL	上一个周期的制惰指令	位号应用与第 2 输入管脚同
oldinertgas2_o	BOOL	上二个周期的制惰指令	位号应用与第 3 输入管脚同
gas_by_ox01_o～gas_by_ox12_o	BOOL	1～12＃炉富氧制气指令	位号引用与第 4～15 输入管脚同

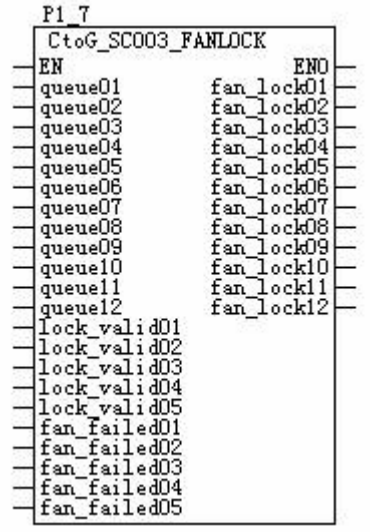
CtoG_SC003_FANLOCK 风机联锁模块

简介

该模块是用于 12 台炉子的风机联锁判断。

EN 和 ENO 能作为附加参数加以设置。

表示



说明

（1）如果 fan_failed01 与 lock_valid01 都为 ON，且

如果 queue01==1，则 fan_lock01 = ON

（2）如果 fan_failed02 与 lock_valid02 都为 ON，且

如果 queue01==2，则 fan_lock01 = ON

（3）如果 fan_failed03 与 lock_valid03 都为 ON，且

如果 queue01==3，则 fan_lock01 = ON

（4）如果 fan_failed04 与 lock_valid04 都为 ON，且

如果 queue01==4，则 fan_lock01 = ON

（5）如果 fan_failed05 与 lock_valid05 都为 ON，且

如果 queue01==5，则 fan_lock01 = ON

（6）fan_lock02～fan_lock12 的动作规律与 fan_lock01 相同。

参数描述

参数	数据类型	含义	补充说明
queue01～queue12	INT	1～12＃排队风机号	
lock_valid01～lock_valid12	BOOL	1～5＃风机联锁指令	ON 为联锁有效
an_failed0～an_failed05	BOOL	1～5＃风机停车信号	故障为 ON
fan_lock01～fan_lock12	BOOL	1～12＃炉风机联锁标志	

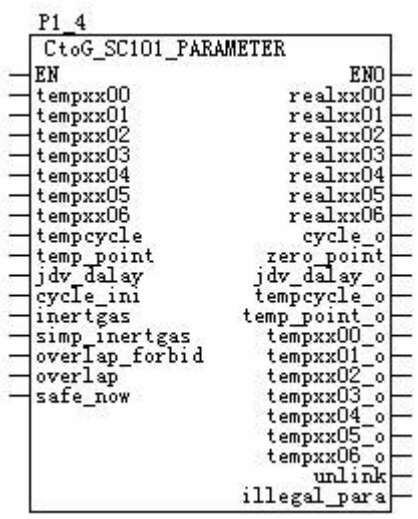
CtoG_SC101_PARAMETER 参数管理模块

简介

该模块是用于参数限幅、参数修改与传递、原始参数管理。

EN 和 ENO 能作为附加参数加以设置。

表示



说明

（1）加氮阀延迟时间有限制，*jdv_dalay＝jdv_dalay，如果 jdv_dalay>5，则*jdv_dalay＝5；如果*jdv_dalay<0，则*jdv_dalay＝0。

（2）避免重风功能：若制气、不允许重风但出现了重风，则吹风预设时间、吹净预设时间减少（不能增大），总周期增大（不能减少），其他预设时间参数可以增加或减少。

（3）预设上吹时间 tempxx01 不能小于 10 秒，否则置为 10 秒。

（4）预设二上吹时间 tempxx03 必须大于 8 秒，否则置为 8 秒。

（5）上加氮阀要早落 8 秒，即 tempxx00（预设吹风时间）－tempxx05（预设上加氮时间）－jdv_dalay（预设上加氮阀开启延时）≥8 秒，否则，tempxx05（预设上加氮时间）= tempxx00（预设吹风时间）－jdv_dalay（上加氮阀开启延时）－8 秒。

（6）tempxx06（预设吹风回收时间）不大于 tempxx00（预设吹风时间），否则 tempxx06

＝tempxx00，并且 tempxx06 不小于零，否则 tempxx06＝0。

（7）周期修改报警，当输出预设周期 tempcycle_o 不等于当前周期 cycle_o 时，unlink = ON；否则 unlink=OFF；当修改循环周期时，给出报警提示。

（8）简单制惰参数管理，当选择制惰、并采用简单制惰时（吹风、吹风回收、上吹放空三阶段制惰），预设上吹放空时间等于总周期减吹风。下吹、二上、吹净、上吹加氮四个预设参数强制为 0。

（9）当炉况安全（包括停炉），将七个预设时间参数（tempxx00－tempxx06）赋给实际时间参数，将预设循环周期赋给当前周期，预设吹净起点赋给当前吹净起点。

（10）除采用简单制惰且正处于制惰阶段之外，发生周期大于 300 或小于 110，四个基本时间参数小于等于 0，实际时间参数之和不等于实际周期等情况时。给出非法参数报警，并将原始参数赋给实际参数。原始参数为吹风 26 秒、上吹 16 秒、二上 10 秒、吹净 0 秒、下吹等于周期减去上述参数。周期等于 cycle_ini,数值由组态时确定。

参数描述

参数	数据类型	含义	补充说明
tempxx00～tempxx06	INT	预设吹风、上吹、下吹、二上、吹净、上加氮、回收时间	
tempcycle	INT	预设循环周期	
temp_point	INT	预设吹净起点	
jdv_dalay	INT	加氮阀延迟时间	
cycle_ini	INT	循环时间初始化值	直接填入原始周期时间，如 120
inertgas	BOOL	制惰指令	
simp_inertgas	BOOL	简单制惰指令	若制惰过程不包括下吹放空阶段则置为 ON，否则为 OFF。
overlap_forbid	BOOL	不许重风指令	若不允许重风，则置为 ON,否则为 OFF。
overlap	BOOL	出现重风标志	需通过其他模块进行是否重风判断
safe_now	BOOL	炉况安全标志	
realxx00～realxx06	INT	实际吹风、上吹、下吹、二上、吹净、上加氮、回收时间	
cycle_o	INT	实际循环周期	
zero_point	INT	实际吹净起点	
jdv_dalay_o	INT	加氮阀延迟时间	用于对加氮阀延迟时间进行限幅，一般在第一次引用该模块时填写。引用位号与第 10 输入管脚同
tempcycle_o	INT	预设循环周期	引用位号与第 8 输入管脚同
temp_point_o	INT	预设吹净起点	引用位号与第 9 输入管脚同
tempxx00_o~tempxx06_o	INT	预设吹风、上吹、下吹、二上、吹净、上加氮、回收时间	引用位号与第 1～7 输入管脚同
unlink	BOOL	周期不一致标志	
illegal_para	BOOL	非法参数标志	

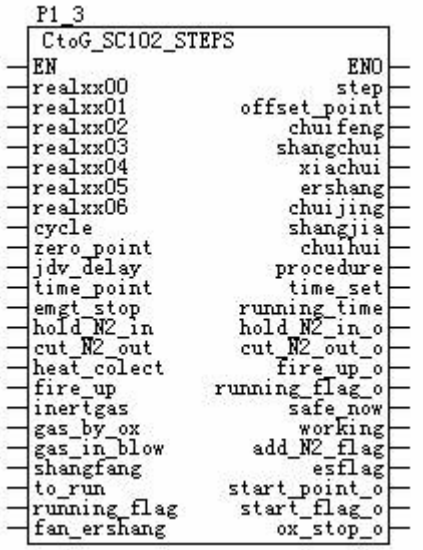
CtoG_SC102_STEPS 阶段判断模块

简介

该模块是用于预启动、预停炉及正常循环时的阶段转换。

EN 和 ENO 能作为附加参数加以设置。

表示



说明

（1）offset_point（偏移时间）＝time_point（时间指针）＋cycle（实际循环周期）－zero_point（吹净时间），offset_point 不能小于 cycle，否则 offset_point＝offset_point－cycle；offset_point 不能小于 0，否则 offset_point 置 0。

（2）常规制气循环，实现吹风、上吹、下吹、二上、吹净、上加氮、回收、送吹风气等阶段切换。

（3）升温操作运行，实现吹风、上吹放空、下吹放空、二上放空阶段切换。

（4）制惰操作运行，实现吹风、回收、上吹放空、下吹放空、二上放空的阶段切换。

（5）富氧制气操作运行，实现富氧运行阶段切换。

（6）预启动，预启动首先必须做二上或二上放空（制惰或升温时）10s,再按照吹风排队转入正常运行，富氧制气开炉先执行富氧启动 10 秒，在预启动期间操作预停炉仍然执行预启动、紧急停炉时不执行预启动程 F 序。

（7）预停车必须在吹风>=8s 后，再安全停车，在预启动期间也执行安全停炉，富氧制气时安全停炉先执行富氧停炉 10s。

（8）风机停车联锁切二上，常规制气过程中，收到风机停车联锁切二上信号时，将运行阶段切换为联锁二上，只有紧急停炉后才回复正常。

参数描述

参数	数据类型	含义	含义
realxx00～realxx07	INT	实际吹风、上吹、下吹、二上、吹净、上加氮、回收时间	
cycle	INT	实际循环周期	
zero_point	INT	吹净	
jdv_delay	INT	加氮阀延迟	
time_point	INT	时间指针	
emgt_stop	BOOL	紧急停炉指令	

hold_N2_in	BOOL	吹风回收指令	
cut_N2_out	BOOL	回收放空指令	
heat_colect	BOOL	送吹风气指令	
fire_up	BOOL	升温指令	
inertgas	BOOl	制惰指令	
gas_by_ox	BOOL	富氧制气指令	升温指令、制惰指令、富氧制气指令若同时有两个以上起作用，会自动停炉，回复正常后会首先执行预启动。
gas_in_blow	BOOL	回收制气指令	
shangfang	BOOL	初期上放指令	设置为 ON，常规制气时上吹前 10 秒放空
to_tun	BOOL	预运行标志	
running_flag	BOOL	运行标志	
fan_ershang	BOOL	风机跳二上标志	
step	INT	运行阶段	
offset_point	INT	偏移指针	
chuifeng	INT	偏移吹风起点	
shangchui	INT	偏移上吹起点	
xiachui	INT	偏移下吹起点	
ershang	INT	偏移二上起点	
chuijing	INT	偏移吹净起点	
shangjia	INT	偏移上加氮起点	
chuihui	INT	偏移吹风回收起点	
procedure	INT	内部循环阶段	
time_set	INT	当前设定时间	
running_time	INT	当前运行时间	
hole_N2_in_o	BOOL	吹风回收指令	
cut_N2_out_o	BOOL	回收放空指令	
fire_up_o	BOOL	升温指令	
running_flag_o	BOOL	运行标志	
safe_now_o	BOOL	炉况安全标志	
working	BOOL	工作状态标志	
add_N2_flag	BOOL	加氮标志	
esflag	BOOL	过渡管脚，不需连接	
start_point_o	INT	过渡管脚，不需连接	
start_flag_o	BOOL	过渡管脚，不需连接	
ox_stop_o	BOOL	过渡管脚，不需连接	

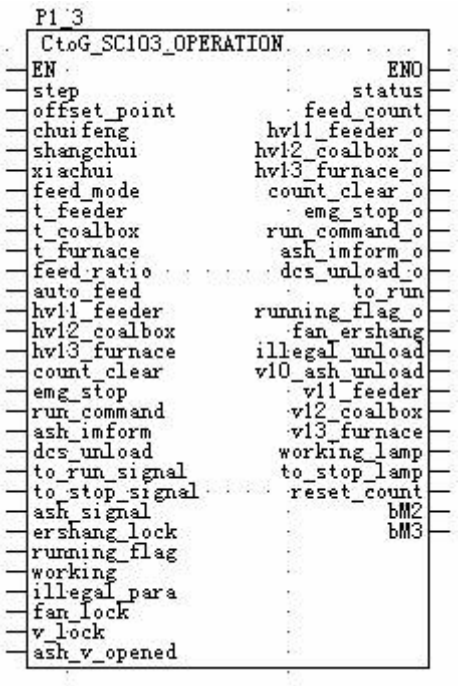
CtoG_SC103_OPERATION 操作盒模块

简介

该模块是用于加焦控制、下灰控制、联锁。

EN 和 ENO 能作为附加参数加以设置。

表示



说明

（1）几种不同加焦模式切换。手自动加焦通过第 11 输入管脚 auto_feed 切换。自动加焦模式通过第 6 输入管脚 feed_mode 选择。自动加焦模式包括：吹风同时打开、上吹同时打开、下吹打开、依次打开、非连续加焦 5 种模式，富氧制气只能选用上吹同时打开、依次打开、非连续加焦 3 种模式，带振动给料机的自动加焦机只能选用非连续加焦模式。

（2）运行灯控制。运行灯控制：如果炉子在运行，则点亮现场运行灯。预停灯控制：如果进入预停程序，则点亮现场预停灯。

（3）下灰联锁控制。下灰时必须停炉且下灰未结束不允许开炉。

（4）对现场操作的响应。炉子可以在现场启动、预停炉。当现场同时按下预启动、预停车二个按钮时，执行紧急停炉。

（5）联锁响应。非法参数与阀检信号都引起紧停，风机连锁方式选择为联锁切二上，则在风机跳车时送出联锁切二上标志。

参数描述

参数	数据类型	含义	补充说明
step	INT	运行阶段	直接与 CtoG_SC102_STEPS 阶段判断模块的第 1 输出管脚 step 相连
offset_point	INT	偏移指针	
chuifeng	INT	偏移吹风起点	
shangchui	INT	偏移上吹起点	
xiachui	INT	偏移下吹起点	
feed_mode	INT	加焦模式选择	
t_feeder	INT	给料时间设定	
t_coalbox	INT	煤箱阀开设定	
t_furnace	INT	入炉阀开设定	
feed_ratio	INT	间歇加料频率	
auto_feed	BOOL	自动加焦指令	

hv11_feeder	BOOL	给料手操指令	
hv12_coalbox	BOOL	煤箱阀手操指令	
hv13_furnace	BOOL	入炉阀手操指令	
count_clear	BOOL	加焦数清零指令	
emg_stop	BOOL	紧急停炉指令	
run_command	BOOL	开炉指令	
ash_imform	BOOL	通知下灰指令	
dcs_unload	BOOL	DCS 下灰指令	
to_run_signal	BOOL	现场预开炉指令	现场按钮指令检测
to_stop_signal	BOOL	现场预停炉指令	现场按钮指令检测
ash_signal	BOOL	现场下灰反馈指令	现场按钮指令检测
ershang_lock	BOOL	风机联锁转二上组态指令	若工艺要求风机联锁转二上，则设置为 ON，否则为 OFF
running_flag	BOOL	运行标志	
working	BOOL	工作状态标志	
illegal_para	BOOL	非法参数标志	
fan_lock	BOOL	风机联锁标志	
v_lock	BOOL	阀检联锁标志	
ash_v_opened	BOOL	灰门打开标志	灰门开启状态检测
status	INT	下灰运行阶段	
feed_count	INT	加焦手操指令	
hv11_feeder_o	BOOL	给料手操指令	
hv12_coalbox_o	BOOL	煤箱阀手操指令	
hv13_furnace_o	BOOL	入炉阀手操指令	
count_clear_o	BOOL	加焦数清零指令	
emg_stop_o	BOOL	紧急停炉指令	
run_command_o	BOOL	开炉指令	
ash_imform_o	BOOL	通知下灰指令	
dcs_unload_o	BOOL	DCS 下灰指令	
to_run	BOOL	预运行标志	
running_flag_o	BOOL	运行标志	
fan_ershang	BOOL	风机跳二上标志	
illegal_unload	BOOL	非法下灰标志	
v10_ash_unload	BOOL	下灰电磁阀信号	
v11_feeder	BOOL	给料信号	
v12_coalbox	BOOL	煤箱阀信号	
v13_furnace	BOOL	入炉阀信号	
working_lamp	BOOL	工作状态灯信号	
to_stop_lamp	BOOL	预停炉状态灯信号	
reset_count	INT	过渡管脚,不需连接	
bM2	BOOL	过渡管脚,不需连接	
bM3	BOOL	过渡管脚,不需连接	

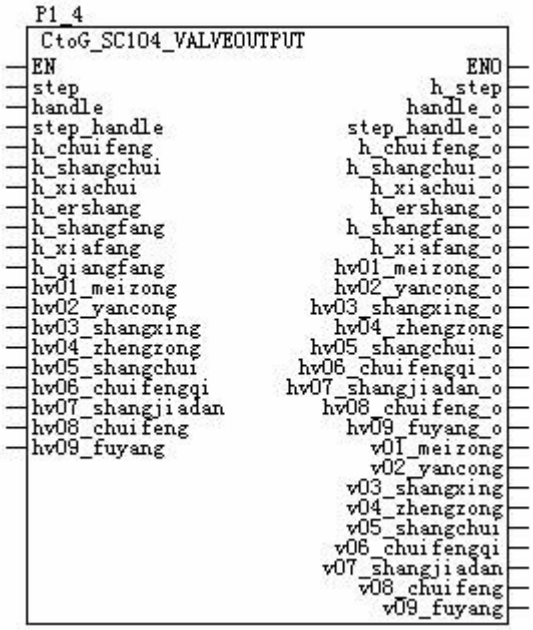
CtoG_SC104_VALVEOUTPUT 阀位输出模块

简介

该模块是用于实现手动操作、阀位输出。

EN 和 ENO 能作为附加参数加以设置。

表示



说明

(1) 阶段手操，如果 handle＝ON，step_handle＝ON 时，阶段操作可以逐个置为 ON。

(2) 单阀手操，如果 handle＝ON，step_handle＝OFF 时，单阀操作可以逐个置为 ON。

(3) 自动时手操屏蔽，如果在自动运行，则 handle＝OFF，step_handle＝OFF。

(4) 阀位输出，根据下表《间歇煤造气炉阀位组态表》实现阀位信号输出组合。

间歇煤造气炉阀位组态表

阀位 \ 阶段	0 停炉	1 吹风	2 上吹	3 下吹	4 二上	5 吹净	6 上加氮	7 吹风回收	8 送吹风气	9 上吹放空	10 下吹放空	11 吹回制气	12 富氧启动	13 富氧运行	14 富氧预停	22 联锁二上	强行放空
1 煤总阀			ON	ON	ON	ON	ON	ON				ON	ON	ON	ON	ON	
2 烟囱阀			ON	ON	ON	ON	ON	ON	ON			ON	ON	ON	ON	ON	
3 上下行阀				ON							ON						
4 蒸汽总阀			ON	ON	ON		ON			ON	ON	ON	ON	ON	ON	ON	
5 上下吹阀				ON							ON						
6 吹风气阀									ON								
7 上加氮阀							ON										
8 吹风阀		ON				ON		ON	ON			ON					
9 富氧阀														ON			

带电状态的空格缺省为OFF。ON表示继电器带电，OFF表示继电器不带电，斜线表示带电状态不定。

参数描述

参数	数据类型	含义	补充说明
step	NT	运行阶段	直接与 CtoG_SC102_STEPS 阶段判断模块的第 1 输出管脚 step 相连
handle	BOOL	手动操作指令	
step_handle	BOOL	阶段手操指令	
h_chuifeng	BOOL	吹风手操指令	
h_shangchui	BOOL	上吹手操指令	
h_xiachui	BOOL	下吹手操指令	
h_ershang	BOOL	二上手操指令	
h_shangfang	BOOL	上放手操指令	即上吹放空
h_xiafang	BOOL	下放手操指令	即下吹放空
h_qiangfang	BOOL	强放手操指令	即强行放空
hv01_meizong	BOOL	煤总阀手操指令	
hv02_yancong	BOOL	烟囱阀手操指令	
hv03_shangxing	BOOL	上下行（阀）手操指令	
hv04_zhengzong	BOOL	蒸总阀手操指令	
hv05_shangchui	BOOL	上下吹（阀）手操指令	
hv06_chuifengqi	BOOL	吹风气（阀）手操指令	
hv07_shangjiadan	BOOL	上加氮（阀）手操指令	

hv08_chuifeng	BOOL	吹风阀手操指令	
hv09_fuyang	BOOL	富氧阀手操指令	
h_step	INT	运行阶段输出	
handle_o	BOOL	手动操作指令	
step_handle_o	BOOL	阶段手操指令	
h_chuifeng_o	BOOL	吹风手操指令	
h_shangchui_o	BOOL	上吹手操指令	
h_xiachui_o	BOOL	下吹手操指令	
h_ershang_o	BOOL	二上手操指令	
h_shangfang_o	BOOL	上放手操指令	
h_xiafang_o	BOOL	下放手操指令	
hv01_meizong_o	BOOL	煤总阀手操指令	
hv02_yancong_o	BOOL	烟囱阀手操指令	
hv03_shangxing_o	BOOL	上下行（阀）手操指令	
hv04_zhengzong_o	BOOL	蒸总阀手操指令	
hv05_shangchui_o	BOOL	上下吹（阀）手操指令	
hv06_chuifengqi_o	BOOL	吹风气（阀）手操指令	
hv07_shangjiadan_o	BOOL	上加氮（阀）手操指令	
hv08_chuifeng_o	BOOL	吹风阀手操指令	
hv09_fuyang_o	BOOL	富氧阀信号输出	
v01_meizong	BOOL	煤总阀信号输出	
v02_yancong	BOOL	烟囱阀信号输出	
v03_shangxing	BOOL	上下行（阀）信号输出	
v04_zhengzong	BOOL	蒸总阀信号输出	
v05_shangchui	BOOL	上下吹（阀）信号输出	
v06_chuifengqi	BOOL	吹风气（阀）信号输出	

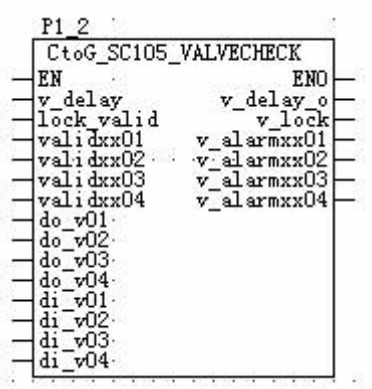
CtoG_SC105_VALVECHECK 阀位检测模块

简介

该模块是用于比较阀位输出与反馈，进行阀位检测。

EN 和 ENO 能作为附加参数加以设置。

表示



说明

（1）阀检延迟时间 v_delay 限幅为 20 秒，如果 v_delay>20，把 v_delay 置值为 20。

（2）如果 valid01=ON，且 do_valve01 与 di_valve01 状态不一致，保持时间大于等于 v_delay 则 v_alarm01 = ON，否则 v_alarm01 = OFF。

（3）如 lock_valid＝ON，如果 v_alarm01、v_alarm02、v_alarm03、v_alarm04 任何一个为 ON 则 v_lock = ON；否则 v_lock＝OFF。

参数描述

参数	数据类型	含义	补充说明
v_delay	UINT	阀检延时	特别注意数据类型为 UINT
lock_valid	BOOL	阀检联锁指令	
validxx01	BOOL	1＃阀检指令	
validxx02	BOOL	2＃阀检指令	
validxx03	BOOL	3＃阀检指令	
validxx04	BOOL	4＃阀检指令	
do_v01	BOOL	1＃阀位输出信号	
do_v02	BOOL	2＃阀位输出信号	
do_v03	BOOL	3＃阀位输出信号	
do_v04	BOOL	4＃阀位输出信号	
di_v01	BOOL	1＃阀位反馈信号	阀位关闭 反馈为 ON
di_v02	BOOL	2＃阀位反馈信号	阀位关闭反馈为 ON
di_v03	BOOL	3＃阀位反馈信号	阀位关闭反馈为 ON
di_v04	BOOL	4＃阀位反馈信号	阀位关闭反馈为 ON
v_delay_o	UINT	阀检延时	用于对阀检延迟时间进行限幅，一般在第一次引用该模块时填写。引用位号与第 1 输入管脚同
v_lock	BOOL	阀检联锁标志	
v_alarmxx01	BOOL	1＃阀检报警	
v_alarmxx02	BOOL	2＃阀检报警	
v_alarmxx03	BOOL	3＃阀检报警	

v_alarmxx04	BOOL	4＃阀检报警

4. DEH 模块

DEH 阀门伺服控制模块

简介

该模块用来驱动阀门伺服控制模块 FW346 卡的，实现 FW346 卡和主控卡之间的数据交互，所不同的是该模块主要是下发和显示阀门伺服控制模块的控制参数。

注意：

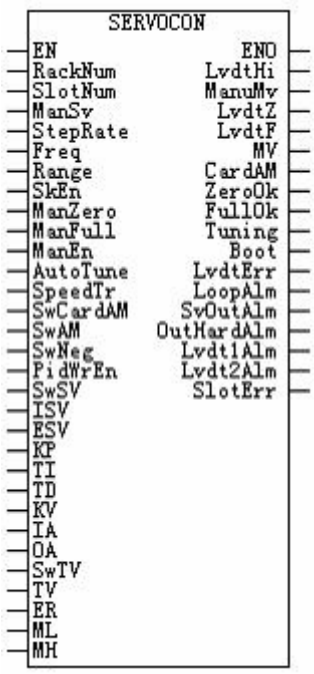
a) 使用时要指定机笼序号和槽位序号，在运行模块时会检查相应的卡件 ID 是否正确，如果组态的机笼序号和槽位序号与实际所插的卡件不一致，则置位卡件类型不匹配报警（SLOTERR），以防破坏其他卡件的数据。

b) 机笼序号和实际组态的数据转发卡序号的对应关系，如数据转发卡设为 0#和 1#，对应的机笼号为 0#， 数据转发卡设为 2#和 3#，对应的机笼号为 1#，千万不能直接把数据转发卡的序号直接当成机笼号 RACK 设置，会造成误发数据。

EN 和 ENO 作为附加参数加以设置。

表示

符号



算法

类似于 DEH 测速和超速保护模块，该模块也只是实现把主控卡下发给 FW346 卡的控制参数写到写缓冲区中，把 FW346 卡上发给主控卡的数据从读缓冲区中读出，中间不对数据进行任何处理。

参数描述

参数	数据类型	含义
RackNum	UINT	机笼序号
SlotNum	UINT	槽位序号

ManSv	UINT	DEH 阀位手动设定值
Manzero	UINT	LVDT 手动调零值
ManFull	UINT	LVDT 手动满幅值
Freq	UINT	防卡涩的震荡信号的频率值（限制在 255 内）
Range	UINT	防卡涩震荡信号的幅值（限制在 255 内）
StepRate	UINT	手操器步进值
SpeedTr	UINT	调零调幅时的速度上升时间
SwCardAM	BOOL	卡件手自动标志（OFF：自动；ON：手动）
AutoTune	BOOL	自动调零调幅启动位（OFF：停止，ON：启动）
ManEn	BOOL	LVDT 手动零幅值有效（OFF：无效，ON：有效）
SkEn	BOOL	防卡涩的震荡信号的允许位（OFF：不允许，ON：允许）
KP	UINT	比例参数
TI	UINT	积分参数
TD	UINT	微分参数
ISV	UINT	内给定值（SwSV=0）
ESV	UINT	外给定值(SwSV =1)
TV	UINT	输出跟踪量
IA	UINT	输入补偿值
OA	UINT	输出补偿值
ER	UINT	偏差报警值
ML	UINT	输出低限
MH	UINT	输出高限
KV	UINT	可变增益
SwSV	BOOL	内外给定开关（OFF：内给定，ON：外给定）
SwAM	BOOL	PID 手自动开关（OFF：手动，ON：自动）
SwNeg	BOOL	正反作用开关（OFF：正作用，ON：反作用）
SwTV	BOOL	输出跟踪开关（OFF：不跟踪，ON：跟踪）
PidWrEn	BOOL	伺服卡 PID 值下发允许（OFF：不下发，ON：下发）
LvdtHi	UINT	LVDT 高选值
ManuMv	UINT	DEH 阀位手动给定值
LvdtZ	UINT	显示 LVDT 零位值
LvdtF	UINT	显示 LVDT 满幅值
MV	UINT	DEH 阀位输出值
CardAM	BOOL	显示手自动状态（OFF：手动状态，ON：自动状态）
ZeroOk	BOOL	LVDT 零位自动调整完成（OFF：完成，ON：未完成）
FullOk	BOOL	LVDT 幅位自动整定完成（OFF：完成，ON：未完成）
Tuning	BOOL	整定进行标志（OFF：未成功或没开始，ON：成功）
ALM	BOOL	卡件启动状态标志（OFF：不在启动，ON：在启动逻辑）
LvdtErr	BOOL	LVDT1 和 LVDT2 偏差大报警（OFF：无，ON：有）
LoopAlm	BOOL	手动零调幅值错误（OFF：正确，ON：错误）
SvOutAlm	BOOL	控制给定值输出故障报警（OFF：无，ON：有）
OutHardAlm	BOOL	输出回路是否硬件故障（OFF：无，ON：有）
Lvdt1Alm	BOOL	LVDT1 输入通道硬件是否故障 (OFF：无，ON：有）
Lvdt2Alm	BOOL	LVDT2 输入通道硬件是否故障（OFF：无，ON：有）
SlotErr	BOOL	卡件类型不匹配报警（OFF：匹配，ON：不匹配）

DEH 测量及超速保护模块

简介

该模块用来驱动汽轮机转速测量及超速保护模块 FW345，实现 FW345 卡和主控卡之间的数据交互，通过该模块，主控卡可以给 FW345 卡下发参数，也可以观察 FW345 卡上送给主控卡的数据。

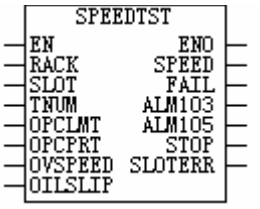
注意：

- 使用时要指定机笼序号和槽位序号,在运行模块时会检查相应的卡件 ID 是否正确，如果组态的机笼序号和槽位序号所在的卡件与实际所插的卡件 ID 不一致，则置卡件类型不匹配报警（SLOTERR）为 ON，以防破坏其他卡件的数据。
- 输入的测速齿盘齿数限制在 255 以内，超过 255 被当作 255 下发。
- 机笼序号和实际组态的数据转发卡序号的对应关系，如数据转发卡设为 0#和 1#，对应的机笼号为 0#， 数据转发卡设为 2#和 3#，对应的机笼号为 1#，千万不能直接把数据转发卡的序号直接当成机笼号 RACK 设置，会造成误发数据。

EN 和 ENO 作为附加参数加以设置。

表示

符号



算法

该模块只是实现把主控卡下发的参数存入写缓冲区，然后传送给 FW345 卡，把 FW345 卡传给主控卡的数据从读缓冲区中读出，中间不对数据进行任何处理，并且两个字节的 SFLOAT 型数据在该模块内都当作 UINT 来处理，需要事先转化好。

103%汽机报警信号：当汽机实际转速超过额定转速的 103%后，要输出报警信号。此信号为继电器输出。一般，汽机额定转速为 3000r/min。103%转速为 3090r/min。

110%汽机报警信号：当汽机实际转速超过额定转速的 110%后，要输出报警信号。此信号为继电器输出。一般，汽机额定转速为 3000r/min。110%转速为 3300r/min。

卡件类型不匹配报警 SLOTERR：当模块指定机笼号和槽位号上的卡件类型与组态不一致，置位 ON，模块直接返回，不会下发数据和接收数据，模块输出保持上次的值。

通讯故障标志 COMERR：当检测到卡件上送的状态信息为 0xFF 时，表示卡件通讯有故障，连续通讯故障 10 秒，将置 COMERR 为 ON。

参数描述

参数	数据类型	含义
RACK	UNIT	机笼序号[0,7]
SLOT	UNIT	槽位序号[0,15]
TNUM	UNIT	输入的测速齿盘的齿数
OPCLMT	BOOL	OPC 限制禁止信号(OFF:允许 103 动作,ON:禁止 103 动作)
OPCPRT	BOOL	OPC 保护禁止信号 110%(OFF:允许 110 动作,ON:禁止 110 动作)

OVSPEED	BOOL	是否机械超速试验(OFF:不允许,ON:允许)
OILSLIP	BOOL	是否油开关跳闸信号(OFF:允许,ON:禁止)
SPEED	UINT	转速
FAIL	BOOL	测速通道是否有故障(OFF:无,ON:有)
ALM103	BOOL	103%报警输出(OFF:无,ON:有)
ALM110	BOOL	110%报警输出(OFF:无,ON:有)
STOP	BOOL	紧急停车(OFF:无,ON:有)
SLOTERR	BOOL	卡件类型不匹配报警（OFF：匹配，ON：不匹配）
COMERR	BOOL	通讯故障标志位（OFF：没有故障，ON：通讯故障）

5. 智能通讯卡模块

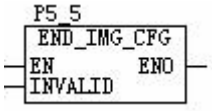
组态结束调用模块

简介

该模块需在组态结束时调用，用来结束组态配置。需放置在程序的最后。

表示

符号



说明

只需将 INVALID 引脚设置成 BYTE 型的任意数，该模块就被激活。

参数描述

参数	数据类型	含义
INVALID	BYTE	激活模块

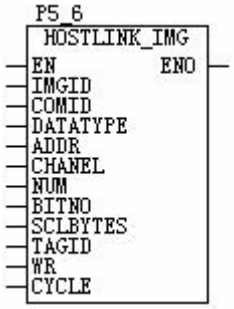
HostLink 协议调用模块

简介

该模块调用 HostLink 协议，实现使用 HostLink 协议的智能设备 I/O 点与自定义变量相互映射，从而实现对智能设备的监控作用。

表示

符号



说明

该模块描述一条使用 HOSTLINK 协议的通讯命令，将 PLC 中的数据和 DCS 中的某个自定义位号建立映射关系，使得对 PLC 的读写操作变为对 DCS 自定义位号数据的读写。每条对

PLC 的操作命令都需要定义一个协议模块建立与自定义位号的映射，它们用不同的 IMGID 号区分。系统最多提供 1000 个映射操作，即系统中各种协议的模块总和最多不超过 1000 个。

指令分为以下几种：读写 IR 寄存器（存放触点线圈状态），读写 HR 寄存器（保持寄存器，断电时存放组态），读写 LR 寄存器（连接寄存器，用来存放与其他 PLC 通讯时的操作命令），读 AR 寄存器（辅助寄存器，存放历史错误记录，日期，终端模式关键字），读 TC 状态，读写 DM 区域，读错误，读写 TC 数据，强置，取消强置。

IMGID：映射号，0-1000 中任意选择，只要不重复就可以。

COMID：串口号，为现在使用的通道号。

DATATYPE：数据类型，参照下表。

ADDR：PLC 地址。

CHANEL：对 I/O 点为操作的起始槽号，读写内部寄存器时为起始地址。

NUM：对 I/O 点时为所要连续操作的槽数（从起始槽算起），读写内部寄存器时为连续操作的地址数。

BITNO：平时都设为 0，只有在强置指令下才使用，表示需要强置哪一个 I/O 点。

SCLBYTES：表示操作对象映射自定义位号是 1：1 字节、2：2 字节、4：4 字节、8、8 字节。

TAGID：定义位号。

WR：读或写指令，1 为读，128 为写。

CYCLE：采样周期，以 ms 为单位，对写操作无效。FW248 每隔 CYCLE 对设备通讯一次，对写操作来说，该值无意义，因为只要对应的自定义变量或位号发生变化就立刻进行写动作，即写操作优先，而且每次写动作只发生一次，除非自定义位号数据又发生了变化。

HostLink 协议调用模块指令号参照表

DATATYPE 号	指令类型
1	IR
2	HR
3	AR
4	LR
5	TCStaus
6	DM
7	ERROR
8	STATUS
9	TCValue
10	FORCEDCANCEL
128＋DATATYPE	FORCEDSET
192＋DATATYPE	FORCEDRESET

注意：组态时，写 DATA 与读写 IR 最好不要同时进行，读 LR 和读 AR 最好不要同时进行，读 HR 和读 TCStaus 最好不要同时进行。

参数描述

参数	数据类型	含义
IMGID	WORD	设置映射号
COMID	BYTE	设置串口号
DATATYPE	BYTE	设置数据类型
ADDR	BYTE	智能设备地址设置
CHANEL	WORD	起始点设置
NUM	BYTE	操作数设置
BITNO	BYTE	强制设置时使用

SCLBYTES	BYTE	自定义位号类型设置
TAGID	WORD	自定义位号设置
WR	BYTE	读或写指令设置
CYCLE	WORD	采样周期设置

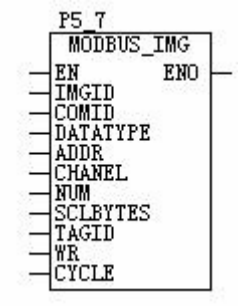
ModBus 协议调用模块

简介

该模块调用 ModBus 协议，实现使用 ModBus 协议的智能设备 I/O 点与自定义变量相互映射，从而实现对智能设备的监控作用。

表示

符号



说明

指令分为以下几种：读写开关量，读写模拟量。

SCControl 模块参数输入说明如下：

IMGID：映射号，从 0-1000 中任意选择，只要不重复就可以。

COMID：串口号，为现在使用的通道号。

DATATYPE：数据类型，1 为线圈，2 是状态，3 是输入寄存器，4 为保持寄存器。

ADDR：PLC 地址。

CHANEL：操作的起始点，由智能设备中的组态决定。

NUM：对 I/O 点时为所要连续操作的点数（从起始点算起）。

SCLBYTES：表示操作对象映射自定义位号是 1：1 字节、2：2 字节、

4：4 字节、8：8 字节。

TAGID：自定义位号。

WR：读或写指令，1 为读，128 为写。

CYCLE： 采样周期，以 ms 为单位，写操作无效。FW248 每隔 CYCLE 对设备通讯一次，对写操作来说，该值无意义，因为只要对应的自定义变量或位号发生变化就立刻进行写动作，即写操作优先，而且每次写动作只发生一次，除非自定义位号数据又发生了变化。

参数描述

参数	数据类型	含义
IMGID	WORD	设置映射号
COMID	BYTE	设置串口号
DATATYPE	BYTE	设置数据类型
ADDR	BYTE	智能设备地址设置
CHANEL	WORD	起始点设置

NUM	BYTE	操作数设置
SCLBYTES	BYTE	自定义位号类型设置
TAGID	WORD	自定义位号设置
WR	BYTE	读或写指令设置
CYCLE	WORD	采样周期设置

设置串口参数模块

简介

由于不同的协议对串口参数的设置不同，所以在调用协议模块前要先设置串口参数。

表示

符号

算法

COMID 为 1～6，表示 6 个串口中的一个；BAUD 为波特率，根据需要设置；DATALEN 为数据长度，单位是 bit，可设置为 5、6、7、8；STOPLEN 为停止位，单位是 bit，可设置为 1、2；VERIFY 为校验方式，0 是无校验，1 是偶校验，2 是奇校验，3 为置 0 校验，4 为置 1 校验。

参数描述

参数	数据类型	含义
COMID	BYTE	设置串口号
BAUD	DWORD	波特率设置
DATALEN	BYTE	数据长度设置
STOPLEN	BYTE	停止位设置
VERIFY	BYTE	校验方式设置

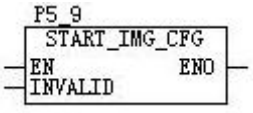
映射配置初始化模块

简介

该模块是在启动或组态编译下载前，进行初始化，需放置在程序的开始部分。

表示

符号



算法

只需将 INVALID 引脚设置成 BYTE 型的任意数，该模块就被激活。

参数描述

参数	数据类型	含义
INVALID	BYTE	激活模块

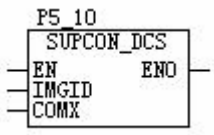
SUPCON_DCS 从机模式模块

简介

该模块是将卡件设置为智能设备的从机模式，监听智能设备命令，并作出回应。

表示

符号



说明

这是一个 MODBUS 从机协议，网关卡作为从机不断侦听外部设备发过来的读写命令，并根据命令将所需的数据发送出去。配合控制站之间的通讯操作，网关卡还可以响应外部设备读写 SCNET II 网络中其他控制站数据的命令，因此，网关卡还可以作为 DCS 系统和外部设备的一个数据接口。

SUPCON_DCS 模块的 MODBUS 数据帧头部格式：

地址	功能码	起始地址 Hi	起始地址 Lo	数量 Hi	数量 Lo
8 位	8 位	8 位	8 位	8 位	8 位

SUPCON_DCS 模块支持的功能码：

MODBUS 数据帧的功能码	DCS 中数据类型
0x01	DO 位号数据
0x02	DI 位号数据
0x03	AO 位号数据
0x04	AI 位号数据
0x05	强迫设置 DO
0x06	强迫设置 AO
0x0f	强迫设置多个 DO
0x10	强迫设置多个 AO
0x20(扩展命令)	读 1 字节自定义位号
0x21(扩展命令)	读 2 字节自定义位号
0x22(扩展命令)	读 4 字节自定义位号
0x23(扩展命令)	写 1 字节自定义位号
0x24(扩展命令)	写 2 字节自定义位号
0x25(扩展命令)	写 4 字节自定义位号

参数描述

参数	数据类型	含义
IMGID	WORD	设置映射号
IMGID	WORD	设置串口号

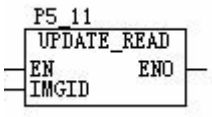
即时读更新模块

简介

该模块是即时读数据更新模块。

表示

符号



说明

所有的读智能设备数据的操作都是按照一定的采样周期(CYCLE)进行的，如果没有到达预定的周期，是不会下发读命令给设备的。但一旦执行了该功能块，此时则不受采样周期的限制，会马上进行该操作。该操作完后，指定的映射仍按照它固有的采样周期工作，除非再次执行该 UPDATE_READ 功能块。

参数描述

参数	数据类型	含义
IMGID	WORD	设置所要设置的映射号

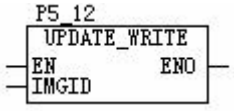
即时写更新模块

简介

该模块是即时写数据更新模块。

表示

符号



说明

正常的写操作，是在数据有变化的情况下被触发的，如果向智能设备写的数据没有变化是不会触发该操作的。执行 UPDATE_WRITE 功能块，则会无条件地马上进行 IMGID 指定的写操作。写操作完后，下次操作则仍必须检测到数据更新后才能执行，除非再次执行该 UPDATE_WRITE 功能块。

参数描述

参数	数据类型	含义
IMGID	WORD	设置所要设置的映射号

2资料版本说明

表 2-1 版本升级更改一览表

资料版本号	输出时间	更改说明
图形化编程使用手册（V1.1）		