

## 6SL3060-4AW00-0AA0连接电缆

产品名称	6SL3060-4AW00-0AA0连接电缆
公司名称	浔之漫智控技术-西门子PLC代理商
价格	666.00/件
规格参数	
公司地址	上海市松江区石湖荡镇塔汇路755弄29号1幢一层A区213室
联系电话	15221406036

## 产品详情

### 6SL3060-4AW00-0AA0连接电缆

模拟量输入模块提供了一个数值用于标准化模拟量信号（电流，电压，电阻或温度）。这个数值代表被测量的参数（例如，以公升计量料位）。这个过程被称作标准化或缩放模拟量值。

图. 01 是标准化的例子。

图. 01

相反，使用用户程序计算过程值。这个过程值被转换成数字信号，模拟量输出模块再将其转换成模拟量信号用来驱动模拟执行器。此转换过程被称作逆标准化。以下可供下载库中的功能块被用于标准化和逆标准化模拟量处理。

图. 02 是逆标准化的例子。

图. 02

因此，y 数值总是转换结果。x 数值在不同情况下都是已知参数。

如下功能块可在下载的两个库文件中可找到。

功能块	语言	描述
skal_linear+lim_int_real [fc165]	stl	此处 x 值是整数，y 值是浮点数。
scl_scale_linear_int-to-real [fc1]	scl	
skal_linear+lim_real_int [fc166]	stl	此处x值是浮点数，y 值是整数。

scl_scale_linear_real-to-int [fc2]	scl	
skal_linear+lim_int [fc167]	stl	此处 x 值和 y 值都是整数。
scl_scale_linear_int [fc3]	scl	
skal_linear+lim_real [fc168]	stl	此处 x 值和 y 值都是浮点数。
scl_scale_linear_real [fc4]	scl	

表 01

### 注意

以上功能块不改变寄存器 ar1/ar2 的地址，并且在 fbd 和 lad 程序中提供了 eno 功能框，rlo=0 或 1 存放在该功能框的br位中。

通过输入 ymin 和 ymax，能限制计算值y（计算出的y值）到低限和高限之间的一个特定值。这限制了模拟量输入和输出量程卡的上限和下限范围。（这限制了模拟量输入和输出卡件的上溢和下溢）

y值根据一般线性方程计算： $y = ax + b$ .

由此引出以下关系： $y = (y1-y0) / (x1-x0) * (x-x0) + y0$

以下是用功能块 skal\_linear+lim\_int\_real [fc165] (integer >; real)实现线性缩放的例子

图. 03

### 例子

模拟量输入模块用来测量一个4ma 至 20ma 的电流信号。此信号在 cpu 内部被转换为 0 至27648。液位用此计算值来测量。由此可知 4ma 对应 0.0m 液位，而 20ma 对应 1.7m 液位。

输出如下参数：

p0(x0=0; y0=0.0)

p1(x1=+27648; y1=+1.7)

按照如下方式调用 fc165 功能块：

图. 04

如何使用gxworks2（与gxdeveloper基本通用）创建简单的sfc（顺序功能图）指令。下面以做一个案例来介绍。

1)打开编程软件gxworks2

创建一个新文件，选择fx2n（plc类型随意），程序语言选择sfc。\*后点击确定。

??

确定后会出现下图，其中块类型一定要选择梯形图块。然后点击执行。

继续执行，如下图

2编写初始化指令。在软件右侧输入程序，如图。\*后在转换/编译

m8002为plc运行初始化，s0-s9为初始化专用的状态，（详情参照用户手册）

4.新建数据右击图中位置，并选择新建数据，如图

新建数据出现下图，直接确定就可以。之后出现选择项，块类型选择sfc，在执行。

的图二

5编写程序框架。执行后出现下图，然后编写程序。

图中左侧，数字为1的地方，就是刚才编写的第一个指令，s0。基本不需要理会他的。

将鼠标放在2上，按键盘上的enter键出现下图

按照默认直接确定，接着按键盘上的enter键出现下图.下图中的2有问号，它代表转换条件，即当条件满足时执行下一条指令。

图中的10及状态继电器的编号，可以修改，比如改成22然后确定。如下图。

这样来回确定，就组成如下图的文件。

在上图中的8完成后，按下确定时，选择项改为jump，如下图，此时程序框架基本完成。

在上图中jump后面填写的数字表示返回哪个状态，一般情况是选择返回初始状态。

6编写\*后的程序

如上图中，2、5、8是转换条件，4、7是执行步骤，下面做出2、4来介绍。点击2处编写指令，然后转换编译，如图

点击4处，编写指令，然后转换编译，如图

????????????????

?

?其他的以此类推。当所有指令编写后，在转换所有程序，程序写完，\*后保存。??其他的以此类推。当所有指令编写后，在转换所有程序，程序写完，\*后保存。

这样一个案例做好了