

SIEMENS西门子 中国咸宁市智能化工控设备代理商

产品名称	SIEMENS西门子 中国咸宁市智能化工控设备代理商
公司名称	浔之漫智控技术(上海)有限公司
价格	.00/件
规格参数	西门子:代理经销商 模块:全新原装 假一罚十 德国:正品现货 实体经营
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	15801815554 15801815554

产品详情

易于使用的编程语言STEP 7 为 S7-1200 提供以下标准编程语言：

LAD（梯形图逻辑）是一种图形编程语言。它使用基于电路图（页 108）的表示法。

FBD（功能块图）是基于布尔代数（页 109）中使用的图形逻辑符号的编程语言。

SCL（结构化控制语言）是一种基于文本的gaoji编程语言（页 110）。创建代码块时，应选择该块要使用的编程语言。用户程序可以使用由任意或所有编程语言创建的代码块。梯形图（LAD）电路图的元件（如常闭触点、常开触点和线圈）相互连接构成程序段。要创建复杂运算逻辑，可插入分支以创建并行电路的逻辑。并行分支向下打开或直接连接到电源线。用户可向上终止分支。简化了编程6.2

易于使用的编程语言入门手册设备手册, 01/2015, A5E02486780-AG 109LAD

向多种功能（如数学、定时器、计数器和移动）提供“功能框”指令。STEP 7 不限制 LAD 程序段中的指令（行和列）数。说明每个 LAD 程序段都必须使用线圈或功能框指令来终止。创建 LAD 程序段时请注意以下规则：不能创建可能导致反向能流的分支。

不能创建可能导致短路的分支。6.2.2 功能块图（FBD）与 LAD 一样，FBD 也是一种图形编程语言。逻辑表示法以布尔代数中使用的图形逻辑符号为基础。要创建复杂运算的逻辑，在功能框之间插入并行分支。

算术功能和其它复杂功能可直接结合逻辑框表示。STEP 7 不限制 FBD

程序段中的指令（行和列）数。SCL 概述结构化控制语言（Structured Control Language, SCL）是用于 SIMATIC S7 CPU 的基于PASCAL 的gaoji编程语言。SCL 支持 STEP 7 的块结构。还可以将用 LAD 和 FBD 编写的程序块包括在用 SCL 编写的程序块中。SCL 指令使用标准编程运算符，例如，用（:=）表示赋值，算术功能（+ 表示相加，- 表示相减，* 表示相乘，/ 表示相除）。SCL 使用标准 PASCAL 程序控制操作，如 IFTHEN-ELSE、CASE、REPEAT-UNTIL、GOTO 和 RETURN。

SCL 编程语言的语法元素可以使用 PASCAL 的任何引用方式。许多

SCL 的其它指令（如定时器和计数器）与 LAD 和 FBD 指令匹配。由于 SCL 能像 ASCAL

一样提供条件处理、循环和嵌套控制结构，因此在 SCL 中可以比在 LAD 或 FBD

中更轻松地实现复杂的算法。将两个局部变量的和赋给一个变量"Data_block_1".Tag := #A;

为数据块变量赋值IF #A > #B THEN "C" := #A; IF-THEN 语句的条件"C" := SQRT (SQR (#A) + SQR (#B));

SQRT 指令的参数作为一种gaoji编程语言，SCL 使用标准语句实现基本任务：算术运算符可以处理各种数值数据类型。结果的数据类型取决于最高有效操作数的数据类型。例如，使用 INT 操作数和 REAL 操作数的乘法运算会产生 REAL 结果值。SCL

程序编辑器可以在创建该块时指定任何块类型（OB、FB 或 FC）以便使用 SCL 编程语言。STEP

7提供包含以下元素的 SCL 程序编辑器：用于定义代码块参数的接口部分

用于程序代码的代码部分 包含 CPU 支持的 SCL 指令的指令树可以直接在代码部分输入指令的 SCL

代码。编辑器包含用于通用代码结构和注释的按钮。要了解更复杂的指令，只需从指令树拖动SCL

指令并将其放入程序中。也可以使用任意文本编辑器创建 SCL 程序，然后将相应文件导入 STEP 7

中。SCL 代码块接口部分，可以声明下列类型的参数： Input、Output、InOut 和 Ret_Val：这些参

数定义代码块的输入变量、输出变量和返回值。执行代码块期间局部使用此处输入的变量名称。通常不会使用变量表中的全局变量名称。 Static（仅适用于 FB，上述示例适用于 FC）：代码块使用静态变量在背景数据块中存储静态中间结果。块会一直保留静态数据，直到多个周期后被覆盖。块的名称（此块将其作为多重背景调用）也存储在静态局部数据中。

Temp：这些参数是执行代码块期间使用的临时变量。

Constant：这些是为代码块指定的常数值。如果从其它代码块调用 SCL 代码块，该 SCL

代码块的参数会显示为输入或输出。功能强大的指令使编程更加轻松 提供您所期望的基本指令S7-1200

CPU 支持许多指令。这些指令在 STEP 7 的指令树中提供，分为以下几组：基本指令 扩展指令

工艺 通信指令有关所有指令的完整概述，请参见“S7-1200 Programmable Controller System Manual”。

该手册描述了许多常见指令。位逻辑指令位逻辑指令的基础是触点和线圈。触点读取位的状态，而线圈

则将操作的状态写入到位中。触点可测试位的二进制状态，结果是在接通(1)时“有能流”，在断开(0)时“没有能流”。线圈的状态反映前导逻辑的状态。如果在多个程序位置中使用地址相同的线圈，则用户

程序中最后一次运算的结果将决定输出更新期间写入物理输出中的值的状态。常开触点常闭触点在赋的位值为 1 时，常开触点将闭合 (ON)。在赋的位值为 0 时，常闭触点将闭合

(ON)。位逻辑运算的基本结构为 AND 逻辑或 OR 逻辑。以串联方式连接的触点创建 AND 逻辑程序段。

以并联方式连接的触点创建 OR

逻辑程序段。可将触点相互连接，创建用户自己的组合逻辑。如果用户指定的输入位使用存储器标识符 I（输入）或 Q（输出），则从过程映像寄存器中读取位值。控制过程中的物理触点信号会连接到 PLC

上的输入端子。CPU 扫描已连接的输入信号并更新过程映像输入寄存器中的相应状态值。通过在输入变量后加上“:P”（例如，“Motor_Start:P”或“I3.4:P”），可指定立即读取物理输入。对于立即读取，

将直接从物理输入读取位数据值，而不是从过程映像中读取。立即读取不会更新过程映像。输出线圈反向输出线圈记录以下通过输出和反向输出线圈的能流的输出结果：

如果有能流通过输出线圈，则输出位设置为 1。如果没有能流通过输出线圈，则输出线圈位设置为 0。如果有能流通过反向输出线圈，则输出位设置为 0。

如果没有能流通过反向输出线圈，则输出位设置为 1。线圈输出指令写入输出位的值。

如果用户指定的输出位使用存储器标识符 Q，则 CPU 接通或断开过程映像寄存器中的输出位，同时设置与能流状态相应的指定位。控制执行器的输出信号连接到 PLC 的输出端子。在 RUN 模式下，CPU 系统

将扫描输入信号，并根据程序逻辑处理输入状态，然后通过过程映像输出寄存器中设置新的输出状态值进行响应。在每个程序执行循环之后，CPU 都会将存储在过程映像寄存器中的新输出状态响应传送到

已连接的输出端子。通过在输出变量后加上“:P”（例如“Motor_On:P”或“Q3.4:P”），可指定立即写入物理输出。对于立即写入，会将位数据值写入到过程映像输出并直接写入到物理输出。线圈并不

局限于在程序段结尾使用。可以在 LAD 程序段的梯级中间以及触点或其它指令之间插入线圈。NOT

触点反相器(LAD)带一个反向逻辑输入的 AND 功能框 (FBD)带反向逻辑输入和输出的 AND 功能框

(FBD)LAD NOT 触点用于对能流输入的逻辑状态取反。如果没有能流流入 NOT

触点，则会有能流流出。如果有能流流入 NOT 触点，则没有能流流出。对于 FBD

编程，可从“收藏夹”(Favorites) 工具栏或指令树中拖动“取反 RLO”(Invert

RLO)工具，然后将其放置在输入或输出端以在该功能框连接器上创建逻辑反相器。AND 功能框 (FBD)

OR 功能框 (FBD) XOR 功能框 (FBD) AND 功能框的所有输入必须都为 TRUE，输出才为 TRUE。OR

功能框只要有一个输入为 TRUE，输出就为 TRUE。XOR 功能框必须有奇数个输入为

TRUE，输出才为 TRUE。在 FBD 编程中，LAD 的触点程序段由与 (&)、或 (>=1) 和异或 (x) 功能框程序

段表示，可在其中为功能框输入和输出指定位值。也可以连接到其它逻辑框，创建用户自己的逻辑组合

。在程序段中放置功能框后，可从“收藏夹”(Favorites)工具栏或指令树中拖动“插入输入”(Insert input)

工具，然后将其放置在功能框的输入侧以添加更多输入。也可以右键单击功能框输入连接器并选择“插入输入”(Insert input)。功能框输入和输出可连接到其它逻辑框，也可输入未连接输入的位地址或位符号名称。执行功能框指令时，当前输入状态会应用到二进制功能框逻辑，如果为真，功能框输出将为真。

1 对于 LAD 和 FBD：如果比较结果为 TRUE，则触点将被激活 (LAD) 或者功能框输出为 TRUE (FBD)。有关更多比较运算，请参见“S7-1200 可编程控制器系统手册”。移动操作用于将数据元素复制到新的存储器地址，并可以从一种数据类型转换为另一种数据类型。移动过程不会更改源数据。MOVE 将存储在指定地址的数据元素复制到新地址。要添加其它输出，请单击 OUT1 参数旁的图标。MOVE_BLK (可中断移动) 和 UMOVE_BLK (不可中断移动) 可将数据元素块复制到新地址。MOVE_BLK 和 UMOVE_BLK 指令具有附加的 COUNT 参数。COUNT 指定要复制的数据元素个数。每个被复制元素的字节数取决于 PLC 变量表中分配给 IN 和 OUT 参数变量名称的数据类型。

1 对于 LAD 和 FBD：在功能框名称下方单击，并从下拉菜单中选择数据类型。选择 (转换源) 数据类型之后，(转换目标) 下拉列表中将显示可能的转换项列表。2 对于 SCL：通过识别输入参数 (in) 和输出参数 (out) 的数据类型来构造转换指令。例如，DWORD_TO_REAL 将 DWord 值转换为 Real 值。将实数 (Real 或 LReal) 转换为整数。指令将实数舍入为最接近的整数值 (IEEE - 舍入为最接近值)。如果该数值刚好是两个连续整数的一半 (例如，10.5)，则指令将其取整为偶数。例如：ROUND (10.5) = 10 ROUND (11.5) = 12

对于 LAD/FBD，请单击指令框中的 ” 来选择输出的数据类型，例如 “DInt”。对于 SCL，默认输出数据类型为 DINT。要舍入为另一输出数据类型，请输入具有数据类型的显式名称的指令名称，例如 ROUND_REAL 或 ROUND_LREAL。out := TRUNC(in); 将实数 (Real 或 LReal) 转换为整数。实数的小数部分被截成零 (IEEE - 取整为零)。表格 6- 5 上取整 (CEIL) 和 Floor 指令

LAD/FBD SCL 说明 out := CEIL(in); 将实数 (Real 或 LReal) 转换为最接近的大于或等于所选实数的整数 (IEEE “向正无穷取整”)。out := FLOOR(in); 将实数 (Real 或 LReal) 转换为最接近的小于或等于所选实数的整数 (IEEE “向正无穷取整”)。CALCULATE 指令可用于创建作用于多个输入上的数学函数 (IN1, IN2, ..INn)，并根据您定义的等式在 OUT 处生成结果。首先选择数据类型。所有输入和输出的数据类型必须相同。要添加其它输入，请单击最后一个输入处的图标。

1 IN 和 OUT 参数必须具有相同的数据类型 (通过对输入参数进行隐式转换)。例如：如果 OUT 是 INT 或 REAL，则 SINT 输入值将转换为 INT 或 REAL 值单击计算器图标可打开对话框，在其中定义数学函数。输入等式作为输入 (如 IN1 和 IN2) 和操作数。单击“确定”(OK) 保存函数时，对话框会自动生成 CALCULATE 指令的输入。对话框显示一个示例，以及可根据 OUT 参数的数据类型加入的一系列指令：说明还必须为函数中的任何常量生成输入。然后会在指令 CALCULATE 的相关输入中输入该常量值。通过输入常量作为输入，可将 CALCULATE 指令复制到用户程序的其它位置，从而无需更改函数。之后，不需要修改函数，就可以更改指令输入的值或变量。当执行 CALCULATE 并成功完成计算中的所有单个运算时，ENO = 1，否则 ENO = 0。有关 CALCULATE 指令的示例，请参见“为复杂数学等式使用 CALCULATE 指令”。