

本溪西门子PLC模块代理商

产品名称	本溪西门子PLC模块代理商
公司名称	浔之漫智控技术-西门子PLC代理商
价格	.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

本溪西门子PLC模块代理商

STEP 7 V11将STEP 7老版本的“库”中的SFC、SFB、FC和FB也称为指令，分为基本指令、扩展指令、工艺和通信4大部分。原来“库”中的某些块被放入基本指令，原来的某些指令被合并为一条指令。以梯形图中的比较指令为例，原来的方框比较器改为像S7-200一样的触点形式（见下图）。可用下拉式菜单设置7种数据类型，因此几十条比较指令被合并为6条。

下面是各类基本指令的变动情况：

- 1) IEC定时器、计数器（SFB0~SFB5）被列入基本指令。
- 2) 数学运算指令可选3种数据类型（Int、Dint和Real），增加了MIN、MAX和LIMIT指令。
- 3) 移动操作增加了SFC BLKMOV、UBLKMOV和FILL指令。
- 4) 转换操作增加了SCALE（FC105）和UNSCALE（FC 106），原来的多条数据转换指令合并为一条CONV指令，转换前后的数据类型均可设置（见下图）。

5) 程序控制操作增加了OPN (打开全局数据块)、OPNI (打开背景数据块)、RE_TRIG R (复位循环周期监视时间)、STP (退出程序)、WAIT (设置等待时间)、PROTECT (更改保护等级) 指令。

6) 字逻辑运算指令将不同的数据类型的指令合并为4条指令，增加了DECO (解码)、ENCO (编码) 和SEL (选择) 指令。

7) 不同数据类型的移位、循环指令合并为4条指令。

8) 其他操作指令：原来的状态位触点指令被合并为常开和常闭触点两条指令。

新增了SET (置位位数组)、RESET (复位位数组)、SETP (在I/O 区域置位位数组)、RESETP (在I/O 区域复位位数组)、SETI (置位字节数组)、RESET I (复位字节数组)、REPL_VAL (输入替换值)、DRUM (执行顺控程序)、DRUM_X (执行顺控程序)、DCAT (离散控制定时器报警)、MCAT (电机控制定时器报警)、I MC (比较输入位与掩码位)、SMC (比较扫描矩阵)、LEAD_LAG (提前和滞后算法)、TONR_X (时间累加器)、WSR (将数据保存到移位寄存器)、SHRB (将位移动到移位寄存器)、SEG (创建7 段显示的位模式)、BCDCPL (求十进制补码)、BITSUM (统计置位位数量)。

编程时可将编程区之外的其他窗口临时关闭，V11版的梯形图比V5.4的紧凑一些

常问问题一：字符中断是什么？ 通过阅读系统手册，我们知道使用字符中断方式接收数据，接收每个字符时都会产生中断。在执行与接收字符事件相连的中断程序前，接收的字符存入SMB2寄存器中，校验状态存入SM3.0。SMB2、SM3.0都是只读的。Port0/Port1共用SMB2/SMB3。Port0对应于中断事件8。Port1对应于中断事件25。常问问题二：字符中断怎么用？ 以端口0接收字符为例

：

当CPU通过端口0接收到一个字符后，会将该字符存入接收字符缓冲区SMB2，然后进入相连接的中断程序中。 注意，对于这段程序，如果在中断程序中不作任何编程，那么当CPU接收n个字符时，中断程序将被执行n次，SMB2寄存器存储接收到的后一个字符。如上位机通过串口调试软件给CPU发

送3个字符16#AA、16#BB和16#CC，

那么与接收字符事件相连的中断程序将被执行3次，SMB2中只能保存后一个接收到的字符16#CC。

常问问题三：用SMB接收多个字符，如何编程？ 由于SMB2只能存储一个字符（一个字节），如果要想接受多个字符，则应当在CPU接收下一个字符之前，在中断程序中通过指针编程将SMB2中存储的字符移出来，以便下一次接收字符。以一个例程说明：

VD0作为地址指针指向VB100，当端口0接收到个字符，将个字符存入SMB2，进入接收中断，将SMB2中的字符复制到指针VD0指向的地址字节VB100中，指针地址加1，VD0指向下一个字节VB101。当接收第二个字符，将字符存入SMB2，进入接收中断，将SMB2中的第二个字符复制到指针VD0指向的地址字节VB101中，指针地址加1，VD0指向下一个字节VB102。当接收第三个字符时以此类推。

另外，如果需要接收n个字符就结束接收，或者需要接收n个字符后循环接收，可以在中断程

例程2，接收到5

个字符后循环接收的程序：

后要提醒您注意的是，程序中的指针不要选择累加器AC，因为累加器不能在主程序和中断程序中传递参数。常问问题四：字符中断和RCV指令之间有什么关系？ 简单地说，当RCV指令使能时，接收字符不进入SMB缓冲区。 我们可以设计个实验分析一下：

在主程序中我们既定义 [PLC](#) 执行RCV接收指令的起始结束条件（起始条件为起始字符16#AA，结束条件为结束字符16#BB），又建立字符中断事件

8，上升沿条件触发RCV指

令。上位机通过串口调试软件给CPU发送字符。 程序如下图：

状态表监控结果：

结论： 当RCV指令不触发的话，接收字符只进入SMB2缓冲区，不进入RCV指令的接收缓冲区。 当RCV指令触发后，满足CPU接收的起始和结束条件的字符存入RCV指令对应的TBL缓冲区。起始条件之前的字符既不存入RCV缓冲区，也不存入SMB2缓冲区。结束条件之后接收到的字符，存入SMB2缓冲区。 通过以上几个问题的分析，我们对于字符中断和SMB2的基本用法和特点做了介绍，相信在实际项目中会有更多灵活的应用等待我们去发掘。

STEP 7 为程序提供 KNOW_HOW_PROTECT 保护功能。如果打开使用此保护功能的块时，仅块接口参数 (IN, OUT 和 IN/OUT 参数) 和块注释可见，而无法显示程序代码、临时/静态变量和网段注释。

以下介绍如何为程序块 (FBs, FCs and DBs) 设置

KNOW_HOW_PROTECT 保护功能：

No. 步骤

- 1 打开要编辑的块，在 LAD/STL/FBD 编辑器中选择 "File > Generate source..."，生成源文件。
- 2 在打开的对话框中输入项目名称，如 "Protect_FB"。
- 3 弹出 "Generate source <名称>" 对话框。选中需要转换的块，点击箭头按钮将其移至右侧 "Blocks Selected" 窗口中。点击 OK 键进行确认后，生成 STL 源文件。关闭 LAD/STL/FBD 编辑器。

图 01

- 4 在 S7 program 的 “ Sources ” 文件夹中打开近生成的源文件。
- 5 声明部分的 "TITLE" 语句下插入 "KNOW_HOW_PROTECT" 命令。

图 02

- 6 通过菜单 "File > Save" 和 "File > Compile" 保存编译 STL 源文件，完成块的保护。

注意：只有通过 STL 源文件才能去除块的保护。如果经 “ KNOW_HOW_PROTECT ” 命令的程序或者项目中的 STL 源文件不再可用，则不能再去除对块的保护。

下表介绍如何去除块 (FBs, FCs and DBs)的 KNOW_HOW_PROTECT 保护功能：

编号	步骤
1	打开源文件
2	删除 "KNOW_HOW_PROTECT" 语句行或用双斜杠将其标记为注释
3	使用菜单命令 "File > Save" 和 "File > Compile" 保存编译该 STL 源文件。
4	至此去除对该块的保护。