

6ES7215-1HG40-0XB0维修保养

产品名称	6ES7215-1HG40-0XB0维修保养
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	品牌:西门子 型号:模块 产地:德国
公司地址	上海市松江区广富林路4855弄88号3楼
联系电话	158****1992 158****1992

产品详情

大家在学习西门子200PLC编程指令的时候肯定会遇到很多问题，比如明明指令写对了却不能正常使用，经常也搞的大家满头雾水吧。好了废话不多说现在开始为大家讲解吧！1：使用定时器加自复位做一个不断重复的计时，调用其他功能或子程序时，为何看起来工作不规律？请注意《S7-200西门子PLC 系统手册》中，关于三种定时器刷新规律的描述。按这种方法使用定时器时，定时器的置位、复位可能与程序扫描周期不配合，存在造成上述问题的机制。定时比较短的定时任务应使用“定时中断”功能，这样更为可靠。2：编了一个利用定时器的程序，在编译时已经通过，为何下载到CPU中时提示出错？这种情况往往是调用的定时器号与定时器类型不配合造成的。参见帮助的表格，如T7只能用作TONR，而不能用于TON或TOF。3：定时中断（SMB34/SMB35）长定时为255ms，如何实现更长时间的定时？可以采用T32/T96中断，长时间可到32.767s。在定时中断服务程序中对进入中断的次数进行计数，也能实现更长时间的中断延时。4：定时中断个数不够怎么办？每个定时中断服务程序不一定只能处理一项定时任务，可以把几个任务放在一个定时中断服务程序中。对于定时时间间隔不同的任务，可以计算出它们的定时长度的大公约数，以此作为定时中断的时间设置。在中断服务程序内部对中断事件进行计数，据此编程别处理不同的任务。5：使用子程序时，为何动作只能执行一次，或者某些状态不能结束？如果发生动作不能重复执行，或者状态不能结束（像锁死了一样），而这些功能都与子程序有关时，请检查是否有条件调用子程序。调用子程序的条件在上述动作执行后，或者进入某个状态后不再有效，无法再次“激活”，而脱离上述状态或复位的指令正好在子程序内，必然造成上述的现象。6：带形式参数的子程序，定义为OUT类型的变量为何会在多次调用子程序时互相干扰？那是因为定义为OUT类型的形式参数又在子程序内部参与了运算。凡是此类参数都应当定义为IN_OUT类型。7：与中断服务程序有关的计算任务，为何会偶尔得出不正确的结果？出现这种现象的原因多是在主（子）程序和中断程序之间传递数据的机制不当。中断程序可能在任何时刻执行，如果此时主程序（或子程序）正在对中断程序使用的数据进行操作，其中间结果可能带入中断程序，造成计算结果的变化；同样地，在中断程序中产生数据也对主（子）程序中的计算有类似影响。8：中断服务程序看起来没有执行？可以在中断程序中加一个测试程序段，如使用SM0.0（常为“1”）将一个输出点置位（使用Set指令），观察是否进入中断服务程序。中断程序不执行，多数原因是初始化（连接中断事件和中断程序）的问题，或者没有“开中断”。应该使用SM0.1（或沿触发）执行一次初始化，然后开中断。9：TP170、TP170 micro与S7-200相连接如何做“时钟同步”？TP170默认的时钟格式与S7-200西门子PLC时钟指令所读取的时间日期格式有所区别，读出的时钟需要改变格式才能与TP170等做时钟同步。在TP170的组态软件ProTool的在线帮助中有相关的介绍。

10：高速计数器怎样占用输入点?高速计数器根据被定义的工作模式，按需要占用CPU上的数字量输入点。每一个计数器都按其工作模式占用固定的输入点。在某个模式下没有用到的输入点，仍然可以用作普通输入点；被计数器占用的输入点（如外部复位），在用户程序中仍然访问到。

11：为什么高速计数器不能正常工作?在程序中要使用初次扫描存储器位SM0.1来调用HDEF指令，而且只能调用一次。如果用SM0.0调用或者第二次执行HDEF指令会引起运行错误，而且不能改变次执行HDEF指令时对计数器的设定。

12：对高速计数器如何寻址?为什么从SMDx中读不出当前的计数值？可以直接用HC0；HC1；HC2；HC3；HC4；HC5对不同的高速计数器进行寻址读取当前值，也可以在状态表中输入上述地址直接监视高速计数器的当前值。SMDx不存储当前值。高速计数器的计数值是一个32位的有符号整数。

13：高速计数器如何复位到0？选用带外部复位模式的高速计数器，当外部复位输入点信号有效时，高速计数器复位为0也可使用内部程序复位，即将高速计数器设定为可更新初始值，并将初始值设为0，执行HSC指令后，高速计数器即复位为0

14：高速计数器的值在复位后是复位到初始值还是“0”值?外部复位会将当前值复位到0值而不是初始值；内部复位则将当前值复位到初始值。如果你设定了可更新初始值，但在中断中未给初始值特殊寄存器赋新值，则在执行HSC指令后，它将按初始化时设定的初始值赋值。

间接寻址是指使用地址指针来存取存储器中的数据。使用前，首先将数据所在单元的内存地址放入地址指针寄存器中，然后根据此地址存取数据。S7-200

CPU中允许使用指针进行间接寻址的元器件有I、Q、V、M、S、T、C。

建立内存地址的指针为双字长度(32位)，故可以使用V、L、AC作为地址指针。必须采用双字传送指令(MOVD)将内存的某个地址移入到指针当中，以生成地址指针。指令中的操作数（内存地址）必须使用“&”符号表示内存某一位置的地址（长度为32位）。例如：

```
MOVD &VB200, AC1
```

表示将VB200在存储器的32位物理地址值送AC1。VB200是直接地址编号，&为取地址符号。将本指令中&VB200改为&VW200或VD200，指令功能不变。

在使用指针存取数据的指令中，操作数前加有*时表示该操作数为地址指针。例如：

```
MOVW *AC1, AC0
```

表示将AC1作为内存地址指针，把以AC1中内容为起始地址的内存单元的16位数据送到累加器AC0中。