

# 深圳西门子电缆一级供货商

产品名称	深圳西门子电缆一级供货商
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	品牌:西门子 型号:西门子电源线缆 产地:德国
公司地址	上海市松江区广富林路4855弄大业领地88号3楼
联系电话	13564949816 13564949816

## 产品详情

深圳西门子电缆一级供货商

深圳西门子电缆一级供货商

FB块和FC块

FB和FC区别

FB,FC块管脚定义

临时变量引起的麻烦

FB和FC区别

FB--功能块，带背景数据块 FC--功能，相当于函数

FB，FC块均相当于子程序，既可以调用其它FB，FC块，也可以被OB，FB，FC块调用。

他们之间的主要区别是：

1. FB使用背景数据块作为存储区，FC没有独立的存储区，使用全局DB或M区

2.

FB局部变量有STAT和TEMP，FC由于没有自己的存储区因此不具有STAT，TEMP本身不能设置初始值。

本质上，FB，FC的实现目的是相同的；无论何种逻辑要求，FB，FC均可实现。只是实现方式效率不同，这也和工程师个人编程习惯有关。

FB块优点：

1. 易于移植性，对于相同控制逻辑不同参数的被控对象，只要使用不同的背景DB，同一个FB块就可以方便
2. 多重背景，减少重复工作，提高效率

多次调用时，参数修改方便

4. 有独立的存储区

FC块优点：

1. 小巧灵活，对于非多次调用的程序更易理解
2. 不占用额外的存储资源

FB,FC块管脚定义

IN-----变量是外部输入的，只能被本程序块读，不能被本程序块写；

OUT-----是本程序块输出的，他可以本程序块读写，其他程序通过引脚只能读值不能写；

IN\_OUT--- 输入输出变量 本程序块和其他程序都可以读写这个引脚的值。

TEMP -----临时变量，顾名思义是暂时存储数据的变量。这些临时的数据存储在CPU工作存储区的局部数据堆栈（L堆栈）中。

STAT-----在PLC运行期间始终被存储。S7 将静态变量定义在背景数据块（仅对FB而言，FC和OB无静态变量），当被调用块运行时，能读出或修改静态变量；被调用块结束后，静态变量保留在数据块中。深圳西门子电缆一级供货商

为何定义的FB，FC块，多次调用后程序混乱？

对于，多次调用的程序块，FB块建议更换调用不同的背景DB；FC则需要确保使用的存储地址不重复，即每次调用，块中调用的地址不重复。

为何含有定时器或计数器的FB或FC单次调用ok，多次调用时定时器或计数器混乱？

对于多次调用的FB，FC，如为S7定时器，计数器，则需要IN接口中定义TIMER或Counter，每调用一次FB或FC，均赋不同的定时器或计数器号。

如为IEC定时器，计数器，则需要IN接口定义Block\_DB，每调用一次FB或FC，均赋不同的DB块给其中的IEC定时器或计数器。

临时变量引起的麻烦

临时变量可以在组织块OB、功能FC和功能块FB中使用，当块执行时它们被用来临时存储数据，一旦块执行结束，堆栈的地址将被重新分配用于其它程序块使用，此地址上的数据不会被清零，直到被其他程序块赋予新值。

需要遵循“先赋值，再使用”的原则。

因此，有常见的几种情况导致程序运行不正常：

#### 1. 某个块程序运行时好时坏，其中某个数值或多个数值偶尔不正常

此问题在于，一定遵循“先赋值，再使用”。否则，TEMP的数值在每个扫描周期开始未有明确的赋值，此地址的数值将是随机的。

#### 2. 多个块使用TEMP，单独使用任意一个都正常，无法一起正常使用深圳西门子电缆一级供货商

此问题在于，TEMP未能先赋值，再使用；程序块1的TEMP中的数值并没有清零，而是CPU运行机制调用此地址使用或直接分配给程序块2使用，导致这个TEMP地址并不为0，因此程序混乱。深圳西门子电缆一级供货商

### S7-1500 IEC定时器创建

S7-1500定时器创建有以下几种方法：

#### 1. 功能框指令直接拖入块中，自动生成定时器的背景数据块，该块位于“系统块>程序资源”中，参见图6。

自动生成定时器的背景数据块

#### 2. 功能框指令直接拖入FB块中，生成多重背景。

多重背景

功能框指令直接拖入FB、FC块中，生成参数实例，从TIA博途V14开始，参见图8。

参数实例

#### 4. 在DB块、FB的静态变量、FC和FB的INOUT变量中新建IEC\_TIMER、TP\_TIME、TON\_TIME、TOF\_TIME、TONR\_TIME（后面四个从TIA博途V11开始）类型变量，在程序中将功能框定时器指令拖入块中时，在弹出的“调用选项”页面点击“取消”按钮，之后将该建好的变量填入指定位置。对于线圈型指令，这是\*\*方法。

DB块中新建IEC\_TIMER等类型变量（LAD/FBD），如果是IEC\_TIMER等类型变量的数组，S7-1500从V2.0版本开始支持，参见图9~图11。

DB块中的定义

功能框定时器使用

线圈型定时器使用

定时器的使用

#### 2. 如何编程自复位定时器并产生脉冲？

答：正确答案参考表4，同时附上2种常见错误编程方式。

原因：S7-1500的定时器的时间更新发生在定时器功能框的Q点或ET连接变量时，或者在程序中使用背景DB（或IEC\_TIMER类型的变量）中的Q点或者ET时。即如果程序中多次使用同一背景DB的Q点，或者既使用定时器功能框的Q点或ET连接变量，又使用背景DB的Q点，以上两种情况都会造成定时器在一个扫描周期内的多次更新，可能造成定时器不能正常使用的情况。

示例正确与否

正确

错误

多次使用同一背景DB的Q点

错误

同时使用背景DB的Q点以及定时器功能框的Q点连接变量

自复位定时器示例

正确方法的流程，将程序根据指令分为两部分：

分解正确指令

阶段1.初始"DB2".脉冲=False，于是"DB2".脉冲取反为True，触发计时器开始计时，输出的"DB2".脉冲=False，状态不变；定时时间不到，则始终在阶段1；

阶段2.当定时时间到发生在 所处的位置，在TON处定时器更新，Q输出True，因此输出的"DB2".脉冲=True，等到下周期时"DB2".脉冲取反为False，导致输出的"DB2".脉冲=False，等再到下周期时就回到了阶段1；深圳西门子电缆一级供货商

阶段当定时时间到发生在 所处的位置，不影响定时器的更新，需要到下一周期才会改变输出，就回到了阶段2。阶段1.初始DY行"IEC\_Timer\_0\_DB".Q=False，触发计时器开始计时，第二行，当定时时间不到，"IEC\_Timer\_0\_DB".Q=False保持不变，输出的"DB2".脉冲=False；定时时间不到，则始终在阶段1；

阶段2.当定时时间到发生在 所处的位置，在DY行的"IEC\_Timer\_0\_DB".Q处定时器更新，更新后，"IEC\_Timer\_0\_DB".Q=True，取反为False，此时作为TON的输入，使得定时器复位，第二行的"IEC\_Timer\_0\_DB".Q=False，输出的"DB2".脉冲=False，到下周期时就回到了阶段1；

阶段当定时时间到发生在 所处的位置，在第二行的"IEC\_Timer\_0\_DB".Q处定时器更新，更新后，"IEC\_Timer\_0\_DB".Q=True，输出的"DB2".脉冲=True，等到下周期时，"IEC\_Timer\_0\_DB".Q=True，取反为False，此时作为TON的输入，使得定时器复位，第二行的"IEC\_Timer\_0\_DB".Q=False，输出的"DB2".脉冲=False，等再到下周期时就回到了阶段1；

阶段4.当定时时间到发生在所处的位置，不影响定时器的更新，需要到下一周期才会改变输出，就回到了阶段2。

从上可知无论定时器时间到发生在哪个点，定时器都可以实现自复位，但是只有当定时时间到发生在所处的位置时，"DB2".脉冲=True只保持一个周期，形成了脉冲，而在 和 时"DB2".脉冲始终为False无法形成脉冲。在正常程序中 的执行时间在整个扫描周期中占比很小，因此只有很低的概率可以实现脉冲。所以不能使用此种方法，第二个错误方法与之类似。

如何使用定时器实现JQ定时？

答：不能做到。

举例来说，定时1s的接通延时定时器，当程序扫描到定时器功能框的Q点或ET时或者扫描到背景DB（或I\_EC\_TIMER类型的变量）中的Q点或者ET时，如果时间为997ms，只会继续定时等下一次扫描，而下一次扫描到可能就是1003ms，此时定时器接通。也就是说几乎不可能正好1s接通，如果再配合计数器实现更长时间的定时，误差只会越来越大。

所以JQ定时还是推荐使用循环中断（OB30）配合计数器来实现。