

## 鸡西西门子PLC代理商

产品名称	鸡西西门子PLC代理商
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	品牌:西门子 产品规格:模块式 产地:德国
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

鸡西西门子PLC代理商

计时器在OB30—OB38里呢？

是一样的。

可以在OB35里使用SD计时器，可以发现，当程序调用OB35时，计时器开始运行，把OB35执行时间和计时器时间设置大些，可以发现，只要每次在扫描的计时器触发端时，条件满足，计时器就开始运行，直到下一次扫描OB35时再扫描到此条件为止。

可以把计时器时间设置足够大，当计时未结束前把它的触发端变为0，那么其计时停止，直到再次触发。

可以得出计时器的运行只与每次扫描到它的触发端有关。扫描完触发端后，计时器的运行就与触发端无关了，直到下一次再次扫描到此触发端。

### 5. 分析程序

了解了以上的一些基本知识，咱们再来看看刚才图1中的程序。

一个CPU的扫描周期是可以计算的，根据不同的配置和数据的读取，可以计算出不同的周期，在PLC运行时，每个周期的大小也是不一样的，可以大致计算出范围，可以根据每条

语句来计算程序的执行时间，再加上相应的循环周期检测点，周期中断，访问过程映像区，通信负载等。这些时间的长短与CPU型号及使用方式有关。

使用PS307 5A，CPU315-2PN/DP (315-2EH13-0AB0 V 2.6.50)为例。以下所有时间都以此配置为标准。

我们把图1的梯型图换成语句表来分析指令执行的过程。

图10

一个CPU的扫描周期的计算可以根据以下几个过程来进行

图11

A.操作系统初始化循环时间监视

B.扫描PIO

C.扫描PII

D.执行用户程序，并执行程序中定义的操作

E.扫描周期检测点操作系统时间（周期结束时执行挂起的任务，如装载和删除块）

F.CPU返回到周期开始的时间点，并重新开始循环周期监视

在以上的步骤中都是有时间的，虽然很小，但是也占用时间。可以根据不同的硬件组态，参照

CPU Specifications手册进行计算，

为了便于计算和理解，咱们以理想状态来计算。假设CPU周期中的A，B，C，E，F的时间为固定的数值X us。

只分析程序里的"D" --用户程序中的命令执行。

程序是顺序扫描的，从Network 1—3依次进行，

以个周期开始时来分析，首先扫描Network

1中T3计时器为0，因此闭点使能，T2开始计时（0 - 8S），但此时扫描T2输出为0，

因此扫描到Network 2中T2开点不使能，扫描到T3不执行，

Network 3中T3开点不使能，M10.2为0。

到此过程 $[0.4+0.3+2.4+0.3+0.3+2.4+0.3+0.2(\text{或}0.9)] \text{ us} = 6.6 \text{ (或}7.3) \text{ us}$ 。

注意：T2一直在累加时间，相当于此时T2计时也到达6.6（或7.7）us。

然后加上刚才的时间X us，那么一个周期可以认为是 $t=X+6.6 \text{ (7.7) us}$ 。X大于7 us，可以看出语句的执行是在很短的时刻进行，所以大家在编程时常用的每个计时器都会经过若干个程序扫描周期。

因为Timer是异步的，所以T2的时间应该在一个周期里也为 $t=X+6.6 \text{ (7.7) us}$ ，那么根据上面的程序看，因为T2设置为8s，所以应该在大概 $m=8s/[X+6.6 \text{ (7.7)]us}$ 个周期时，T2执行完毕。

T2 假设情况下，T2执行完毕的时刻是在第m个周期内，

A．如果发生在Network2的T2开点之前，那么扫描到此T2开点的语句时，T2的输出变为1，执行下一条语句T2开点就会闭合，T3开始计时。

B．如果T2执行完毕的时刻是程序扫描到T2开点语句之后才发生的，那么因为后面的程序没有对T2的操作，只有在下一个m+1周期，才能检测到T2的变化。T3开始计时。

T3开始计时的前提条件是T2开点闭合，假设在第m个周期里，T3开始计时，那么同样，要经过大概m个周期左右，T3才能执行完毕，到此时，已经经过了2m个周期，因为M10.2线圈是由T3开点的闭合信号来置位的，那么现在就来分析一下什么时候可以发生此动作。

注意：在此例子程序中，在Network1-3中都有对T3的操作

T3 假设在情况下，T3执行完毕的时刻是在第2m个周期。在第2m周期内

A．如果发生在Network1的T3闭点之前，那么在程序扫描到T3闭点的时候，T3的输出值已经变为1了，闭点变为开点，T2输出变为0，往下扫描到Network2的T2开点变为0，T3的SD输出也变为0，继续扫描到Network3，T3开点为0，那么M10.2未被置位。

B．如果发生在Network1的T3闭点之后，Network3的T3开点之前，（则T2是保持为1的），在扫描到T3开点时，T3的输出值变为1，T3开点变为闭点，M10.2被置位。

C．如果发生在Network3的T3开点之后，那么在此周期内对m10.2不会产生置位，在下一周期（2m+1），T3输出值变为1了，所以在Network1里T3闭点变为开点，T2输出变为0，扫描到Network2里，T2开点变为0，导致T3输出值变为0，扫描到Network3里，T3开点变为0，因此不会对M10.2置位。在再下一周期（2m+2），扫描到Network1里T3闭点为0，使能T2重新开始计时。

从以上分析可以看出，M10.2是可以被置位的，但是在条件符合情况下，看T3中情况B的

时间大致为图10中的2, 3, 4, 5, 6操作 $Y = (0.3 + 2.4 + 0.3 + 0.3 + 2.4) = 5.7\mu s$ ，也就是图5中a时刻得在这个时间段内，这个时间极为短暂。因此我们在检测的时候很难捕捉到此信号。

我们可以在假设情况下来计算一下概率，就以现在这个例子

$$t = X + 6.6(7.7)\mu s \quad Y = 5.7\mu s \quad m = 8s/t$$

可以看出概率非常小，只有增大 $5.7\mu s$ 才能增大概率，也就是增大Y（或B）的时间

### 图12

只有SD计时器结束时刻发生在Y时间段内，那么M10.2才能被置位，也就是

Y: Network1的T3闭点之后，Network3的T3开点之前这个时间段内

如何通过试验来验证以上的理论说法呢。

可以实际通过实验来检测

在不做任何修改的情况下，上面的程序要对M10.2进行置位的条件是很难捕捉到的，在 $n \times 2^m$ 个周期也难以捕捉到，因此经过长时间运行程序，M10.2也难以发现被置位。

为了比较直观，加上了Network4，用计数器来大致评估时间。如图13

### 图13

(1) 如果在Network1的T3闭点后加上SFC47设置1ms延长此段时间，这样可以大大增加T3中情况B的时间（也就是增大Y的时间），那这样也就增加了它的概率。可以看出，在C1计算到14时，M20.0（相当与前问所述M10.2）已经被置位。如图14

### 图14

此种情况概率大约也可以计算为

$$t = X + 6.6(7.7)\mu s + 1ms \quad Y = 5.7\mu s + 1ms \quad m = 8s/t$$

可以看出概率大了很多

(2) 如果把此SFC47放在T3闭点之前，那么如T3中情况A的分析，对其程序扫描对产生B的

效果不会有任何增加。可以看出，在C1计算到999时，M20.0（相当与前问所述M10.2）还没有被置位。（补充说明：终也能被置位，在Network3后面放若干个延时块SFC47也不会增加概率即缩短被置位的时间）如图15所截图，可以看出

图15

此种情况概率没有改善，因为关键参数Y没有变化

所以概率还是为 $5.7\mu\text{s}/16\text{s}=0.00000035625$

注：以上情况经过多次检测。

由此可得出结论：

- 1、只有增大B（Network1的T3闭点之后，Network3的T3开点之前这个时间段内）的时间，才能增大置位的概率。
- 2、或者减少计时器时间，也可以在时间方面增大概率，但对编程逻辑无益处

注意：我们刚才的概率分析并不是的，只是假定的理论上情况，而且并没有考虑中断，网络结构，计时器时基（可参考Online Help）等等各种情况。

图1中的程序经过分析。

答案是：M10.2能被置位，只是概率问题

所以对于图2中的程序咱们也可以同样分析它的情况。

答案是：S\_CU计数有可能会丢数（即，不是每一次都能被记录），M6.2能被置位，只是概率问题