

可编程控制器原理及应用

中断概念

● S7-200设置了中断功能

用于实时控制、高速处理、通信和网络等复杂和特殊的控制任务。

中断就是终止当前正在运行的程序，去执行为立即响应的信号而编制的中断服务程序。

执行完毕再返回原先被终止的程序并继续运行

● 中断源

中断源即发出中断请求的事件，又叫中断事件。

可编程控制器原理及应用

● 中断源的类型

为了便于识别，系统给每个中断源都分配一个编号，称为中断事件号。

S7-200系列可编程控制器最多有34个中断源

分为三大类：

通信中断

I/O中断

定时中断

可编程控制器原理及应用

● 中断优先级

优先级：指多个中断事件同时发出中断请求时，CPU对中断事件响应的优先次序。

● S7-200规定的中断优先由高到低依次是：

通信中断

I/O中断

定时中断

可编程控制器原理及应用

- 一个程序中总共可有128个中断

S7-200在各自的优先级组内按照先来先服务的原则为中断提供服务。

在任何时刻，只能执行一个中断程序。

一旦一个中断程序开始执行，则一直执行至完成。不能被另一个中断程序打断，即使是更高优先级的中断程序。

可编程控制器原理及应用

● 最多中断个数和溢出标志位

中断程序执行中，新的中断请求按优先级排队等候。

中断队列能保存的中断个数有限，若超出则会产生溢出。

中断队列的最多中断个数和溢出标志位如表所示。

队列	CPU 221	CPU 222	CPU 224	CPU 226	溢出标志位
通讯中断队列	4	4	4	8	SM4.0
I/O中断队列	16	16	16	16	SM4.1
定时中断队列	8	8	8	8	SM4.2

可编程控制器原理及应用

- 中断程序的标号（INT）是一个中断服务子程序的标识，说明是哪一个中断服务子程序。
- 中断事件的编号（EVNT）是CPU规定的中断事件的编号，指出是哪一个中断源。

可编程控制器原理及应用

中断指令



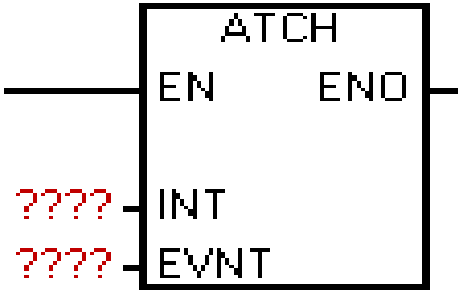
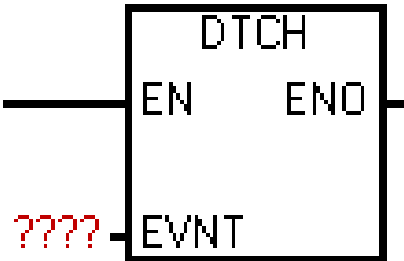
全局中断允许指令ENI (Enable Interrupt)

全局中断禁止指令DISI (Disable Interrupt)

中断连接指令ATCH (Attach Interrupt)

中断分离指令DTCH (Detach Interrupt)

注意：一个中断事件只能连接一个中断程序，但多个中断事件可以调用一个中断程序。

LAD				
STL	ENI	DISI	ATCH INT, EVNT	DTCH EVNT
说明	全局性允许所有中断事件	全局性禁止所有中断事件	将中断事件（EVNT）与中断程序号码（INT）相连接，并启用中断事件。	取消某中断事件（EVNT）与所有中断程序之间的连接，并禁用该中断事件。

可编程控制器原理及应用

(1) ATCH中断连接指令

把中断事件 (EVNT) 和中断服务子程序 (INT) 连接起来, 即把中断源和中断服务子程序对应起来 并允许这个中断事件开放

(2) DTCH中断分离指令

断开中断事件 (EVNT) 与中断服务子程序 (INT) 连接, 使中断源没有相应的中断服务子程序与其对应。该指令同时使这个中断事件禁止

(3) CRET条件中断返回指令

若满足条件, 停止中断服务子程序执行而返回主程序原处。

可编程控制器原理及应用

(4) ENI开中断指令

通过执行ENI指令全局开放中断，允许CPU响应中断请求，执行中断服务程序。

(5) DISI关中断指令

当CPU执行关中断指令（DISI）后，中断排队仍然会产生，但CPU不会响应中断请求，不去执行中断服务程序

中断程序

- 中断程序是为处理中断事件而事先编好的程序；
- 中断程序不是由程序调用，而是在中断事件发生时由操作系统调用；
- 在中断程序中不能改写其他程序使用的存储器，最好使用局部变量；
- 中断程序应实现特定的任务，应“越短越好”；
- 中断程序由中断程序号开始，以有条件返回指令（CRETI）结束；
- 在中断程序中禁止使用DISI、ENI、LSCR和END指令。

建立中断程序的方法

- 方法一：从“编辑”菜单→选择插入（Insert）→ 中断（Interrupt）；
- 方法二：从指令树，用鼠标右键单击“程序块”图标并从弹出菜单→选择插入（Insert）→ 中断（Interrupt）；
- 方法三：从“程序编辑器”窗口，从弹出菜单用鼠标右键单击插入（Insert）→ 中断（Interrupt）。

优先级分组	组内优先级	中断事件号	中断事件说明	中断事件类别
I/O中断	2	0	I0.0上升沿中断	外部输入
	3	2	I0.1上升沿中断	
	4	4	I0.2上升沿中断	
	5	6	I0.3上升沿中断	
	6	1	I0.0下降沿中断	
	7	3	I0.1下降沿中断	
	8	5	I0.2下降沿中断	
	9	7	I0.3下降沿中断	
定时中断	0	10	定时中断0（在SMB34中写入周期时间）	定时
	1	11	定时中断1（在SMB35中写入周期时间）	
	2	21	定时器T32 CT=PT中断	定时器
	3	22	定时器T96 CT=PT中断	

例题

利用定时中断功能编制一个程序，实现如下功能：

当I0.0由OFF→ON，Q0.0亮250ms，灭250ms，如此循环反复直至I0.0由ON→OFF，Q0.0停止闪烁。

Q0.0或为ON，或为OFF，取决于I0.0由ON→OFF的时刻。

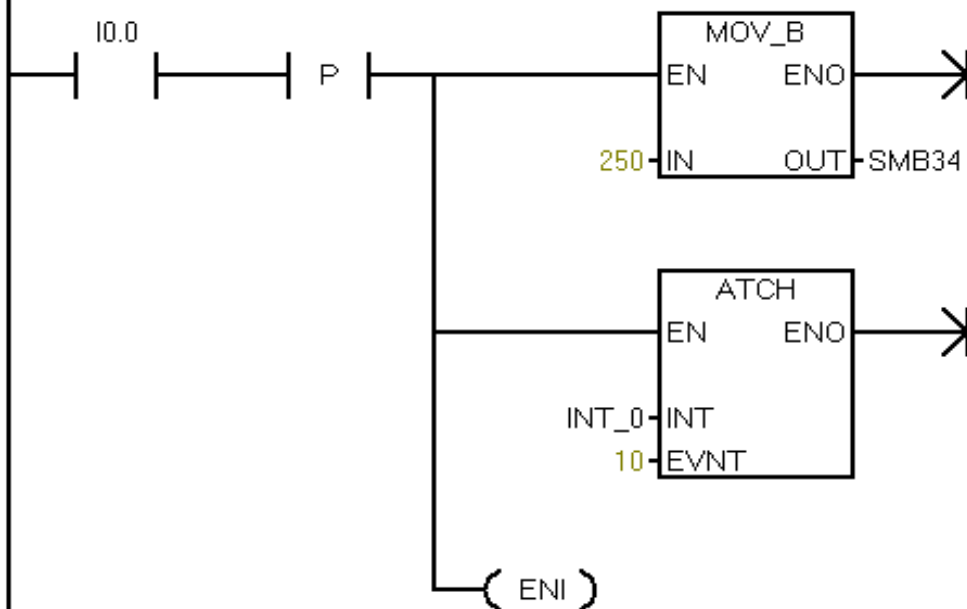
SMB34和SMB35是用于定时中断的时间间隔寄存器

可以设定定时中断0或定时中断1来完成（即将定时值送到SMB34或SMB35中），并将中断事件号（10或11）与中断服务程序连接(ATCH)起来

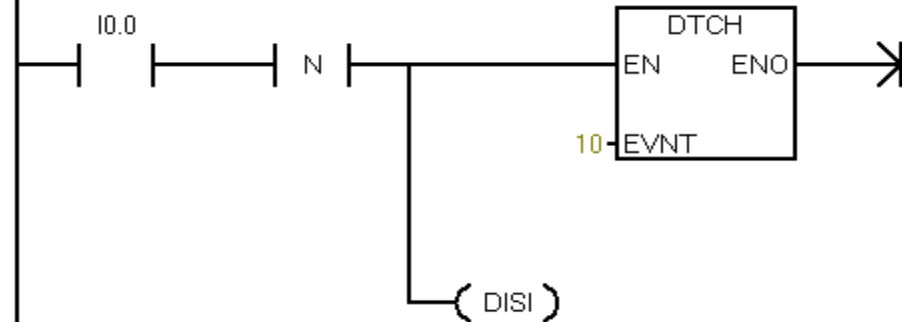
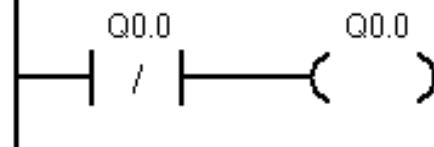
定时中断的中断号为10（定时中断0），11（定时中断1），21（定时器T32中断），22（定时器T96中断）

Network 1

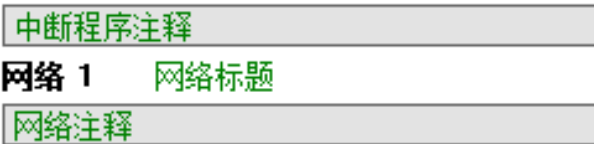
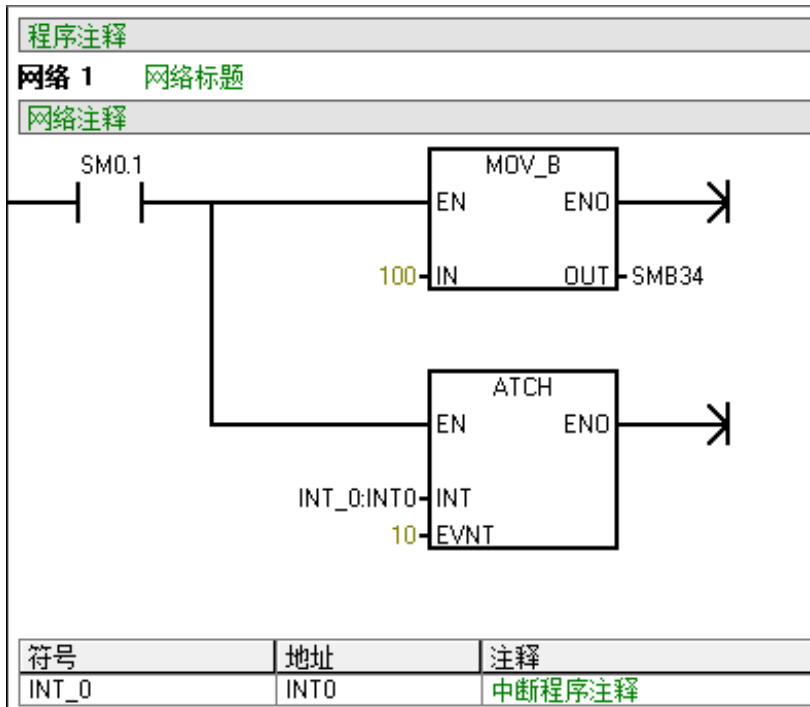
I0.0由OFF变为ON时：
 设定定时中断0的中断时间间隔为250ms
 指定产生定时中断0时执行0号中断程序
 允许全局中断

Network Comment**Network 2**

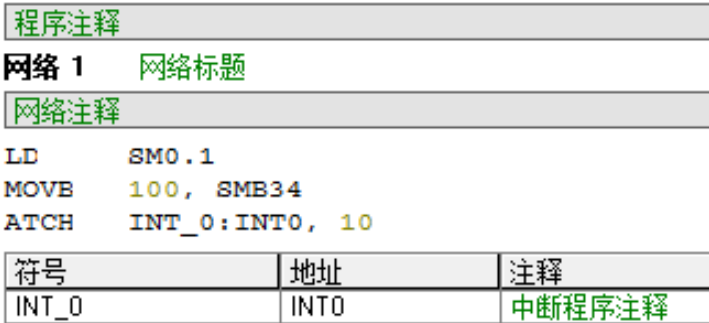
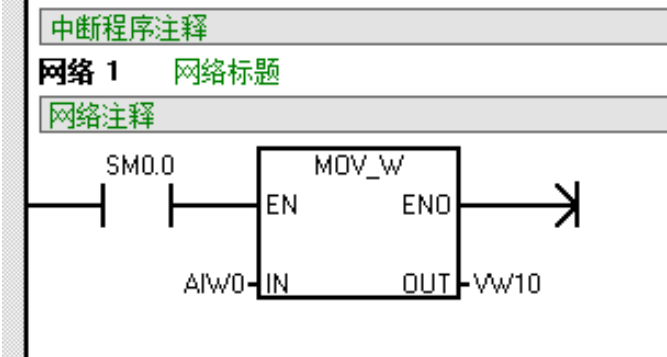
I0.0由ON变为OFF时，取消中断

**Network 1** 250ms执行一次中断，Q0.0通250s，断250s...**Network Comment**

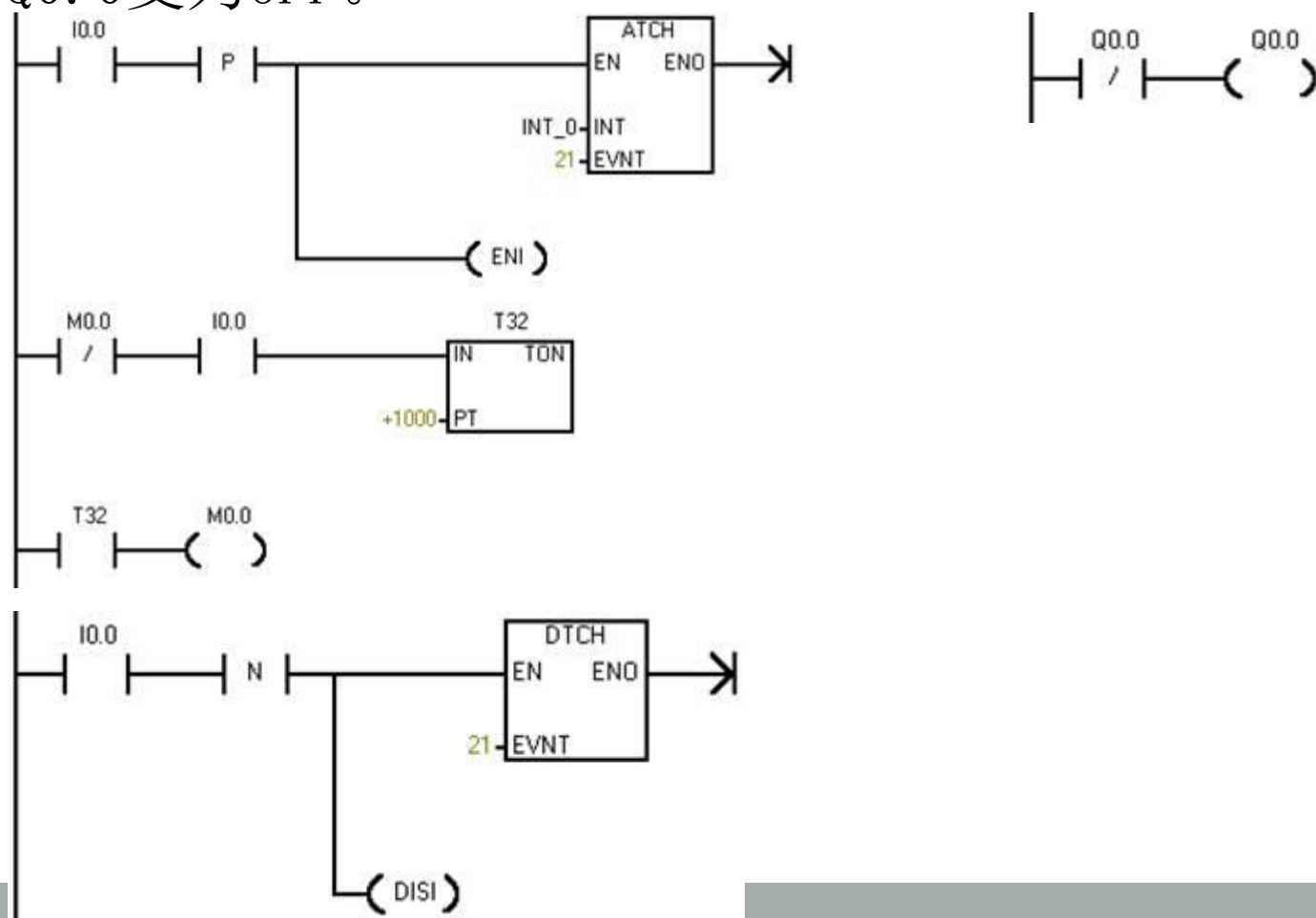
采用中断指令完成每隔0.1s将模拟量输入值AIW0采集进来，存放在VW10中。



```
LD      SM0.0
MOVW    AIW0, VW10
```



利用定时中断功能编制一个程序，实现如下功能：当I0.0由OFF→ON，Q0.0亮1s，灭1s，如此循环反复直至I0.0由ON→OFF，Q0.0变为OFF。



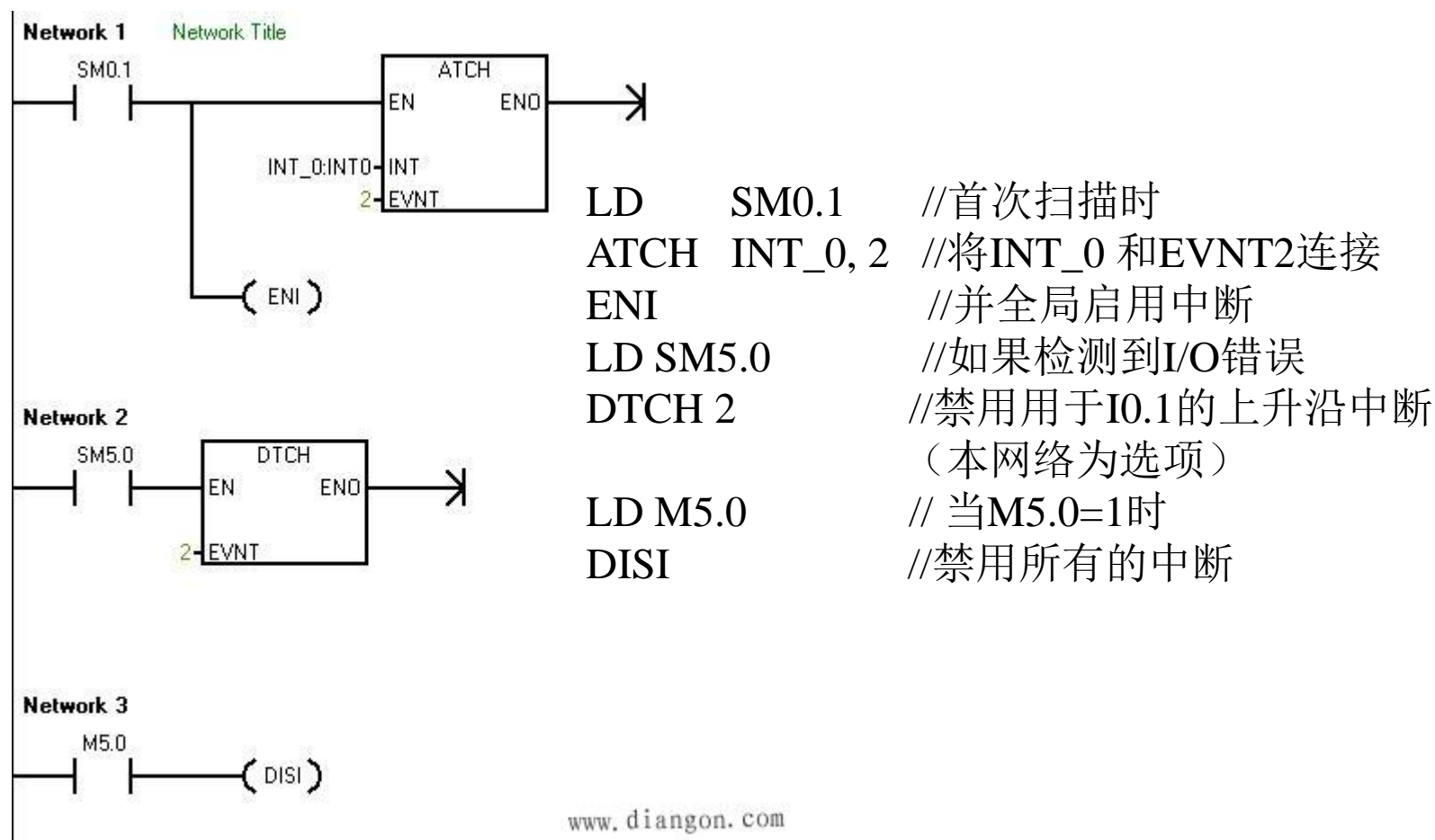
I/O口中断

假设有一个外部按钮接在了**I0.0**的端子上
当按下**I0.0**端子的那一时刻想立刻做某一件事情
首先查中断事件号，得到**0**号事件对应的就是**I0.0**按下的事件
用**ATCH** 指令，在**EVNT**参数中写入**0**,在**INT** 参数中写入**0**
即当**I0.0**闭合的那一瞬间，立刻进入**INT_0**这个中断程序去执行程序
然后再开总中断，**(ENI)**指令就是开总中断

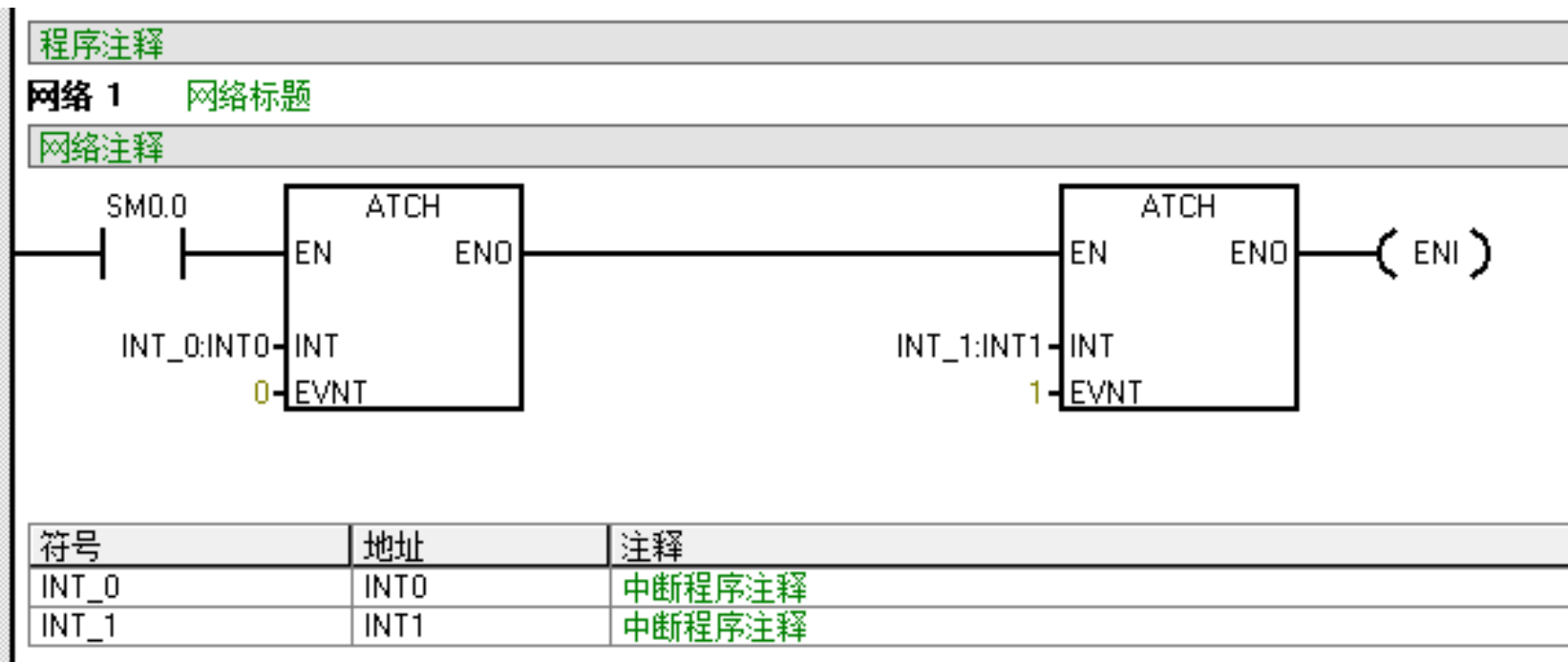
所以使用**S7-200PLC**的中断一般有**3**个步骤：

- 1步，中断连接，使用**ATCH**指令
- 2步，开总中断
- 3步，在要调用的中断程序中，编写程序

由I0.1的上升沿产生的中断事件的初始化程序

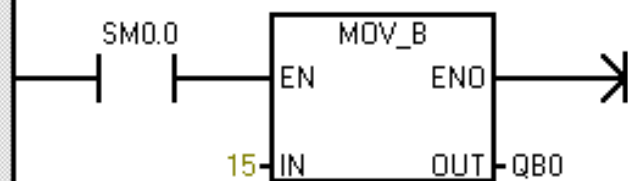


用中断指令控制输出端Q的状态。输入端I0.0接通上升沿时，Q0.0-Q0.3接通，输入端I0.0断开下降沿时，QB0=0。



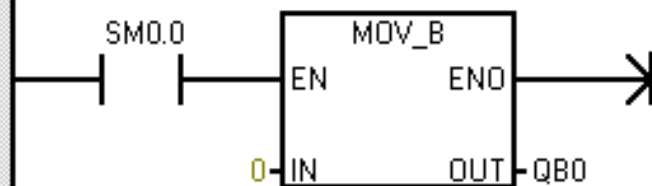
中断程序0注释

网络 1

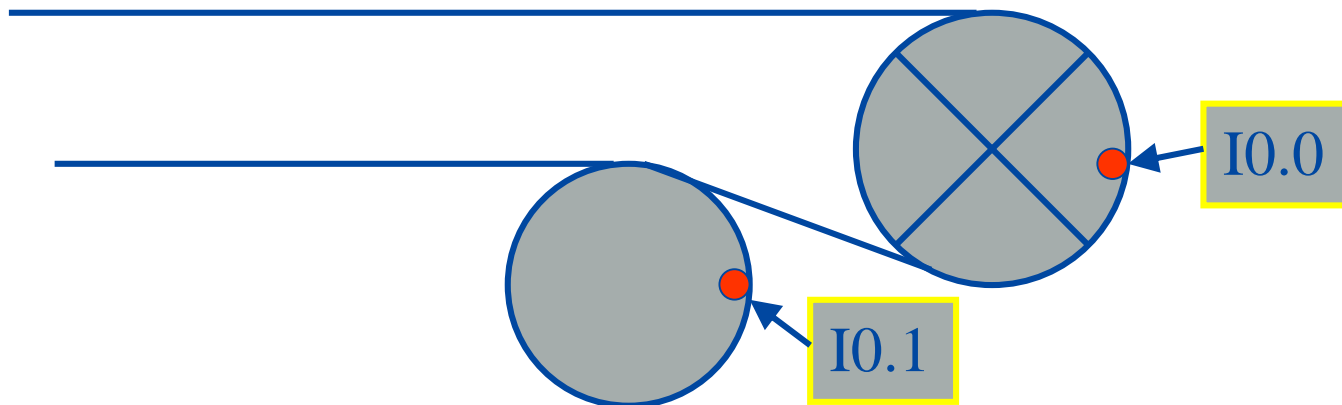


中断程序1注释

网络 1



例子:利用I/O中断测量带式输送机的转速



在带式输送机的驱动滚筒和改向滚筒上各安装一个磁铁，分别用接近开关来测量滚筒的转速。

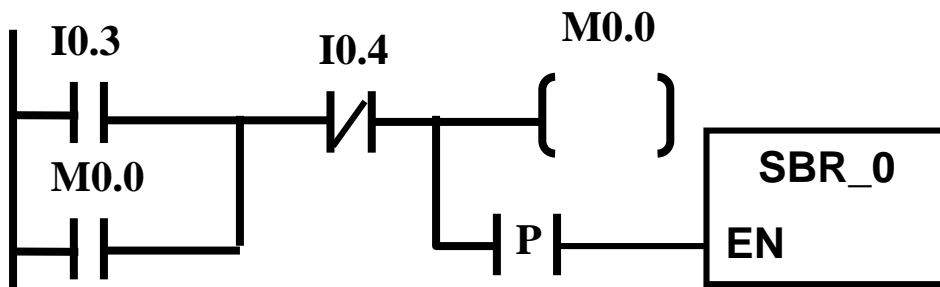
假设：驱动滚筒直径800mm，改向滚筒直径500mm
输送机带速2.0m/s。

则驱动滚筒每转时间约为1256ms。

改向滚筒每转时间约为785ms。

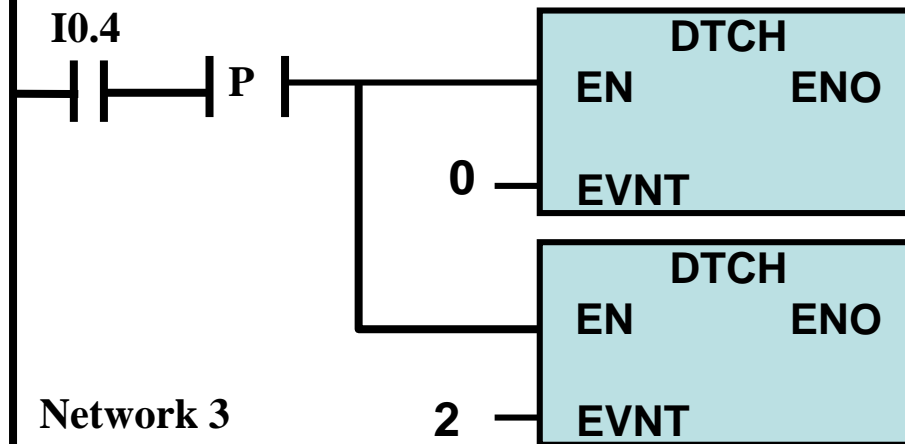
I0.3	启动按钮	I0.0	驱动滚筒PNP型接近开关
I0.4	停止按钮	I0.1	改向滚筒PNP型接近开关

Network 1



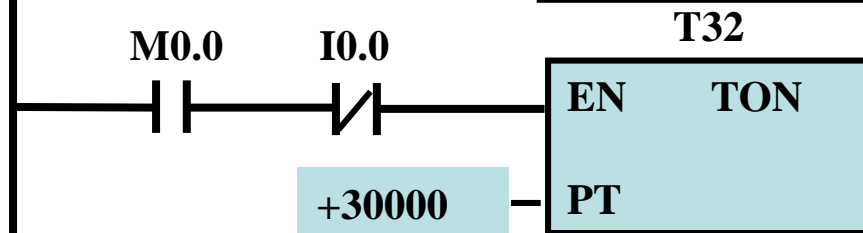
```
LD    I0.3
O     M0.0
AN    I0.4
=     M0.0
EU
CALL  SBR_0
```

Network 2



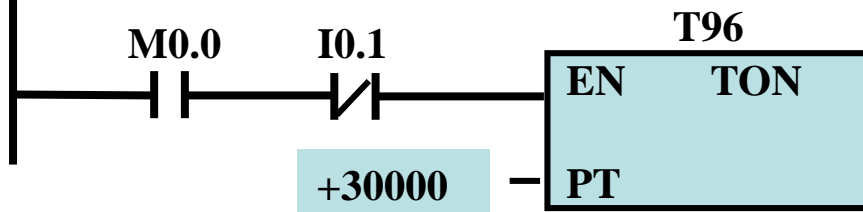
```
LD    I0.4
EU
DTCH  0
DTCH  2
```

Network 3



```
LD    M0.0
AN    I0.0
TON   T32, 30000
```

Network 4



```
LD    M0.0
AN    I0.1
TON   T96, 30000
```

网络 5

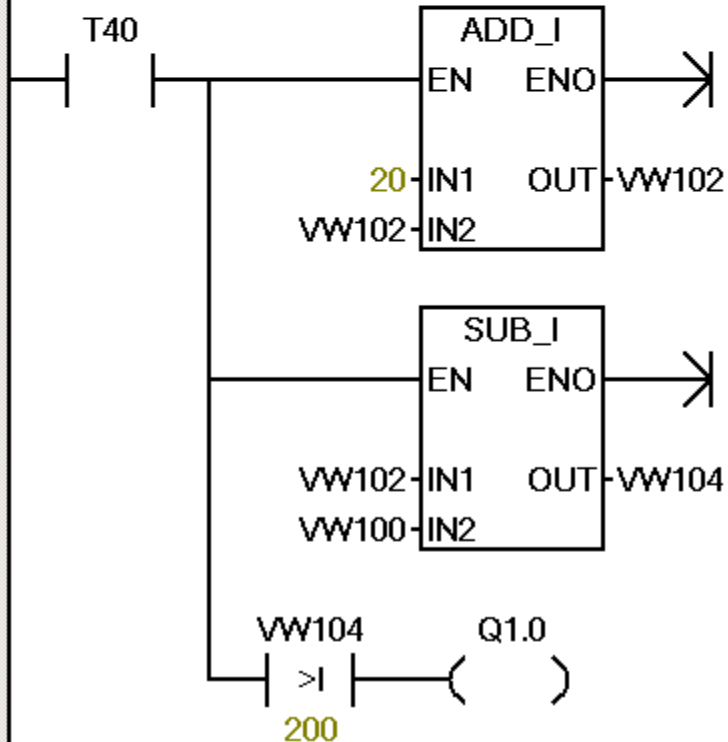
延时40秒



```
LD    M0.0
TON   T40, 400
```

网络 6

比较速度差值,超限报警Q1.0



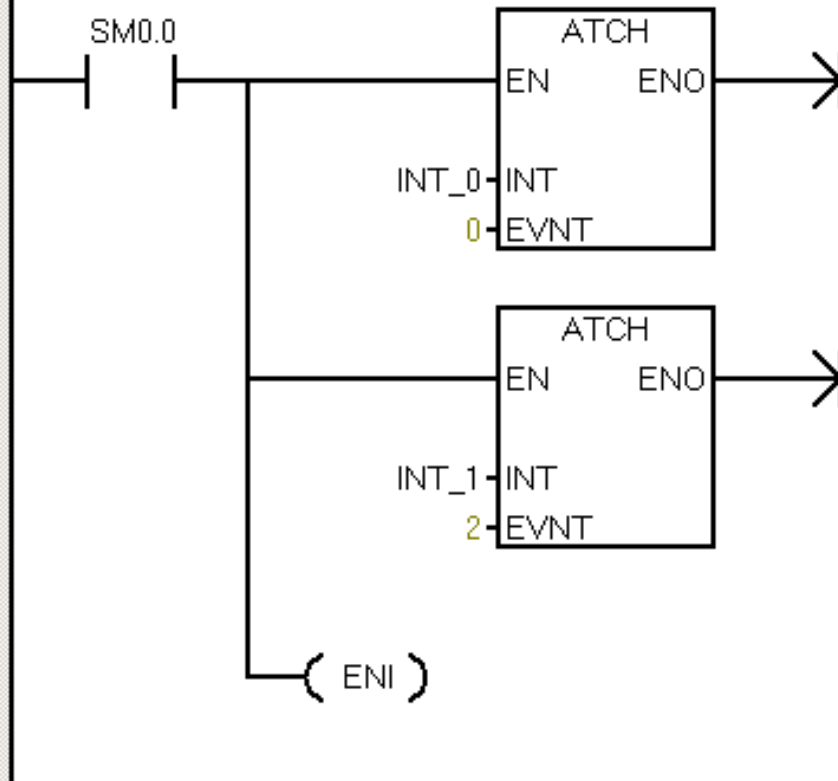
```
LD      T40
+I      20, VW102
MOVW    VW102, VW104
-I      VW100, VW104
AW>     VW104, 200
=       Q1.0
```


子程序

子例行程序注解

网络 1 网络标题

连接I0.0和I0.1的上升沿中断到中断程序,开中断



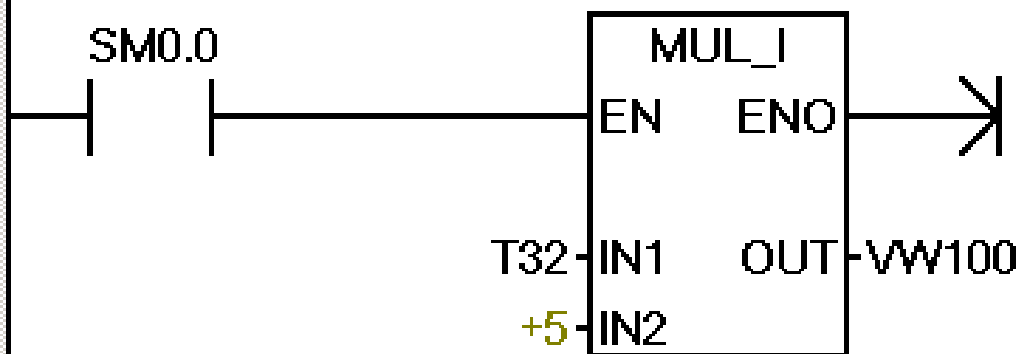
```
LD      SM0.0
ATCH    INT_0, 0
ATCH    INT_1, 2
ENI
```

中断子程序0

中断例行程序注解

网络 1 网络标题

取运行一周的时间,并进行标度转换



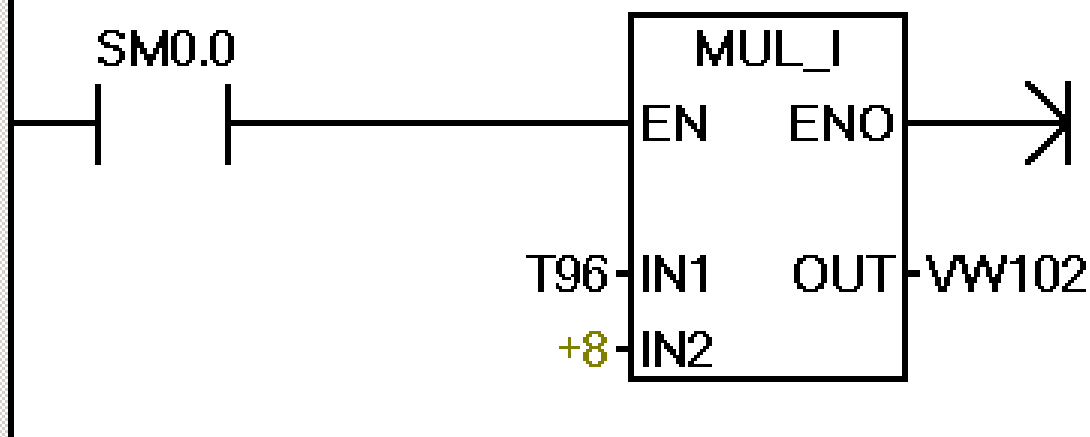
```
LD      SM0.0
MOVW    T32, VW100
*|      +5, VW100
```

中断子程序1

中断例行程序注解

网络 1 网络标题

取运行一周的时间,并进行标度转换



```
LD      SM0.0
MOVW    T96, VW102
*I      +8, VW102
```

