编程实例概述

实际应用

本手册中描述的每个梯形图指令都会触发一个特定操作。将这些指令组合到一个程序中时，便可完成多种自动化任务。本章提供梯形图指令实际应用的以下实例：

- 控制传送带 - 使用位逻辑指令
- 检测传送带的移动方向 - 使用位逻辑指令
- 生成时钟脉冲 - 使用定时器指令
- 跟踪存储空间 - 使用计数器和比较指令
- 使用整数数学运算指令解决问题
- 设置加热烘炉的时间长度

使用的指令

助记符	程序元素目录	描述
WAND_W	字逻辑指令	(字)与运算
WOR_W	字逻辑指令	(字)或运算
--- (CD)	计数器	降值计数器线圈
--- (CU)	计数器	升值计数器线圈
---(R)	位逻辑指令	重置线圈
---(S)	位逻辑指令	置位线圈
---(P)	位逻辑指令	RLO上升沿检测
ADD_I	浮点指令	整数加
DIV_I	浮点指令	整数除
MUL_I	浮点指令	整数乘
CMP <=I, CMP >=I	比较	比较整数
杹| |杹	位逻辑指令	常开触点
杹| / |杹	位逻辑指令	常闭触点
杹( )	位逻辑指令	输出线圈
---(JMPN)	跳转	若非则跳转
---(RET)	程序控制	返回
MOVE	传送	分配值
---(SE)	定时器	扩展脉冲定时器线圈

实例：整型数学运算指令

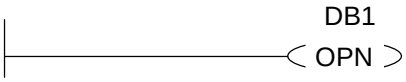
解决数学问题

实例程序显示了如何使用三个整数数学运算指令来产生与下列方程式相同的结果：

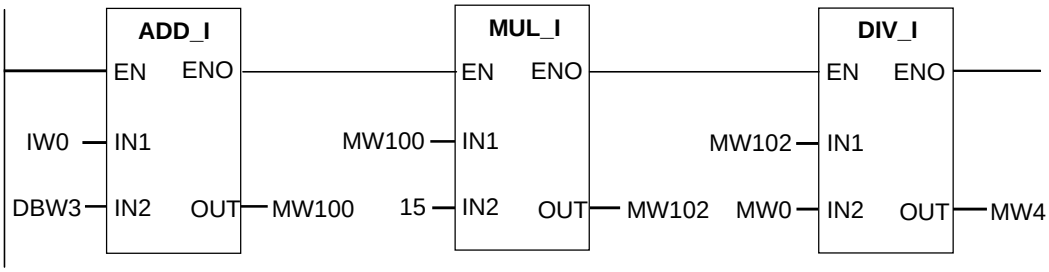
MW4 = ((IW0 + DBW3) x 15) / MW0

梯形图程序

程序段1：打开数据块DB1。



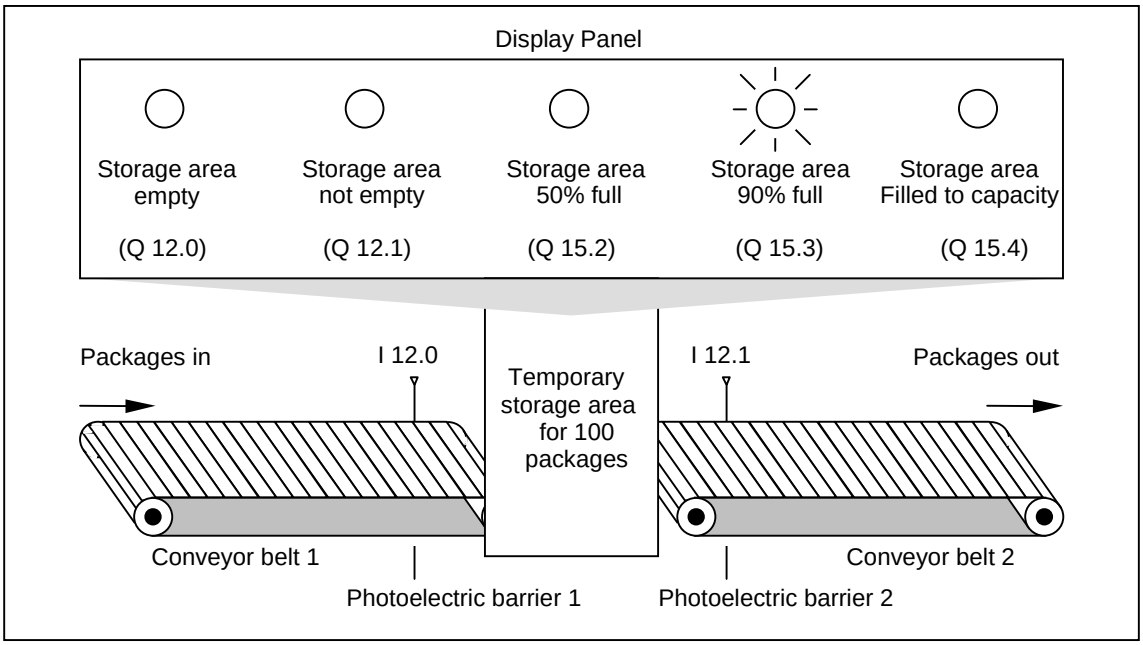
程序段2：输入字IW0加到共享数据字DBW3(必须定义和打开数据块)，总和被载入存储器字MW100。然后，MW100乘以15，结果存储到存储器字MW102中。MW102除以MW0，结果存储到MW4中。



实例：计数器和比较指令

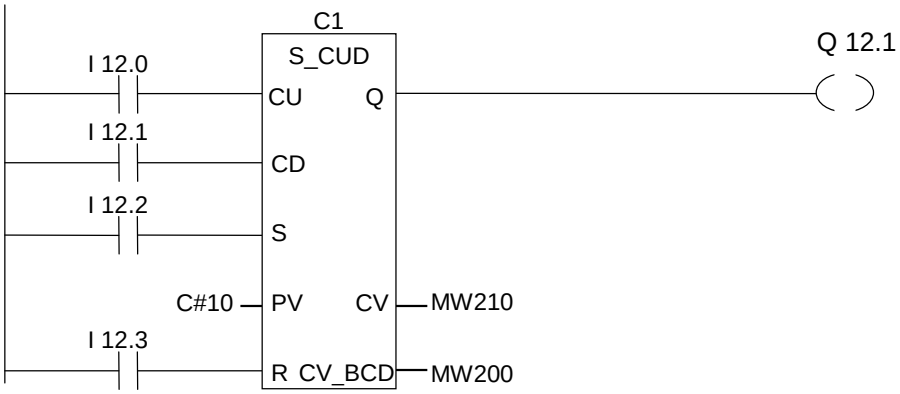
带计数器和比较器的存储区域

下图显示了具有两个传送带且在传送带之间有临时存储区域的系统。传送带1将包裹传送到存储区域。存储区域附近的传送带1末端的光电屏障确定向存储区域传送的包裹数量。传送带2会将包裹从临时存储区域传输到装载码头，而卡车在此将包裹发送给客户。存储区域附近的传送带2末端的光电屏障确定离开存储区域而转向装载码头的包裹数量。带五个指示灯的显示面板将指示临时存储区域的填充量。



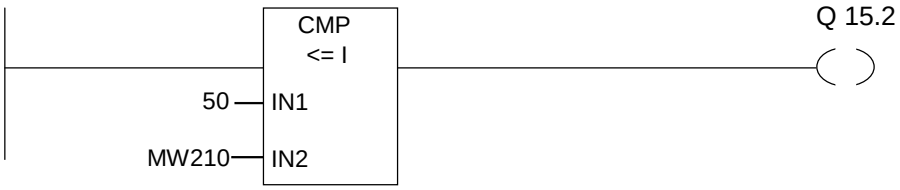
激活显示面板上的指示灯的梯形图程序

程序段1：计数器C1对输入CU处每次从"0"到"1"的信号改变都进行正计数，而对输入CD处每次从"0"到"1"的信号改变都进行倒计数。对于输入S处从"0"到"1"的信号改变，计数器值被设置为值PV。输入R处从"0"到"1"的信号改变将计数器值复位为"0"。MW200包含C1的当前计数器值。Q12.1指示"存储区域非空"。

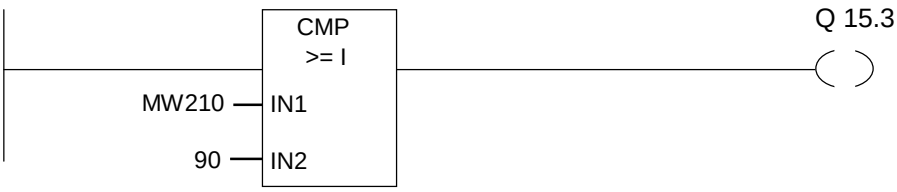


程序段2：Q12.0表明"存储区域为空"。

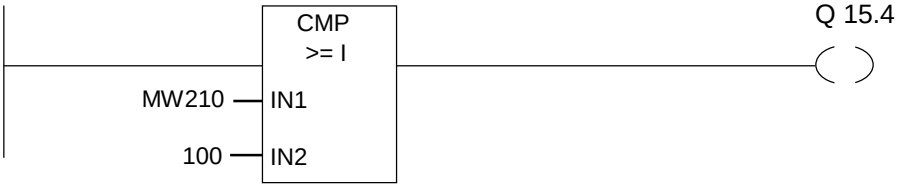
程序段3：如果50小于等于计数器值(换句话说，如果当前计数器值大于等于50)，则表示"存储区域50%满"的指示灯变亮。



程序段4：程序段4：如果计数器值大于或等于90，则表示"存储区域90%满"的指示灯变亮。



程序段5：如果计数器值大于或等于100，则表示"存储区域满"的指示灯变亮。



实例：定时器指令

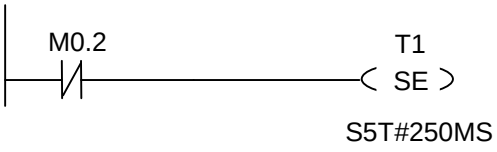
时钟脉冲发生器

当需要生成定期重复的信号时，可使用时钟脉冲发生器或闪烁继电器。时钟脉冲发生器在控制指示灯闪烁的信号系统中很常见。

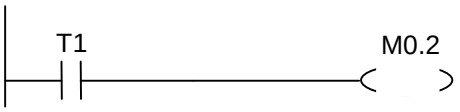
当使用S7-300时，您可用特殊组织块中的时间处理功能来执行时钟脉冲发生器功能。但下列梯形图程序中显示的实例说明的是使用定时器功能产生时钟脉冲。实例程序显示如何通过使用定时器实现任意的时钟脉冲发生器。

产生时钟脉冲(脉冲占空比1:1)的梯形图程序

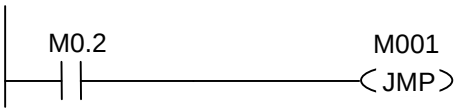
程序段1：如果定时器T1的信号状态为0，将时间值250毫秒载入T1，并将T1作为扩展脉冲定时器启动。



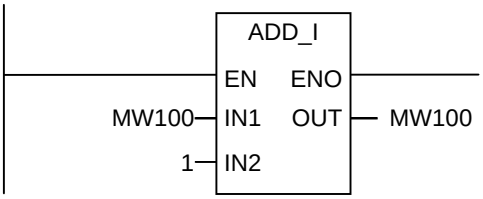
程序段2：该定时器的状态临时保存在一个辅助存储器符号中。



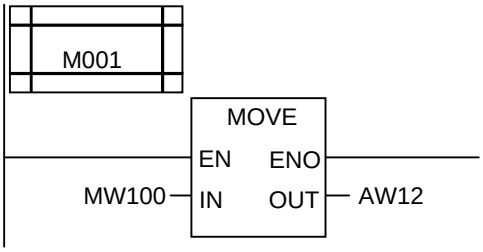
程序段3：如果定时器T1的信号状态为1，则跳转至跳转标签M001。



程序段4：定时器T1超时后，存储器字100增加1。

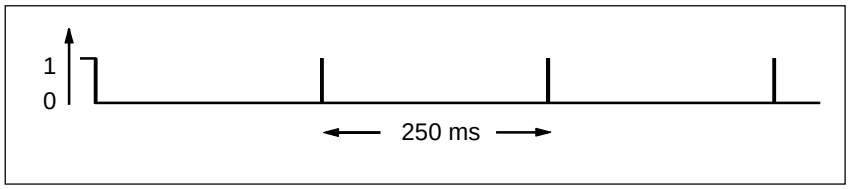


程序段5：**MOVE**指令允许在输出Q12.0到Q13.7输出不同的时钟频率。



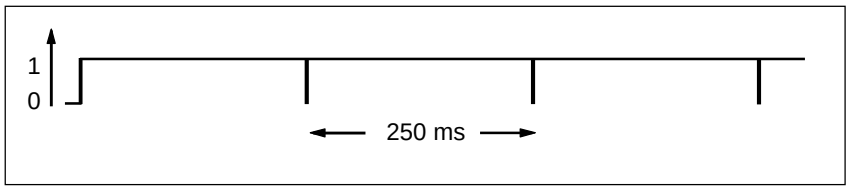
信号检查

定时器T1的信号检查为opener M0.2生成以下逻辑运算(RLO)结果。



一旦定时时间到，就会重新启动定时器。因此，由 |/| / |/| M0.2进行的信号检查只简单产生信号状态1。

RLO取反(反向)：



每隔250毫秒RLO位为0。忽略跳转且存储器字MW100的内容增加1。

实现特定频率

通过存储器字节MB101和MB100的单个位，可以实现下列频率：

MB101/MB100的位	**频率(赫兹)**	**持续时间**	
M 101.0	2.0	0.5s	(250毫秒开/ 250毫秒关)
M 101.1	1.0	1 s	(0.5秒开/ 0.5秒关)
M 101.2	0.5	2s	(1秒开/ 1秒关)
M 101.3	0.25	4s	(2秒开/ 2秒关)
M 101.4	0.125	8s	(4秒开/ 4秒关)
M 101.5	0.0625	16s	(8秒开/ 8秒关)

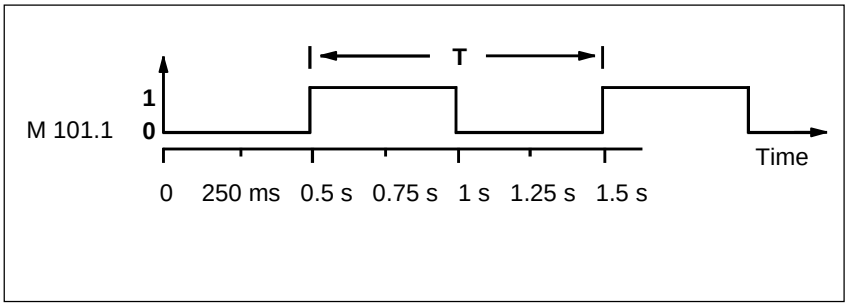
M 101.6	0.03125	32s	(16秒开/ 16秒关)
M 101.7	0.015625	64s	(32秒开/ 32秒关)
M 100.0	0.0078125	128 s	(64秒开/64秒关)
M 100.1	0.0039062	256 s	(128秒开/128秒关)
M 100.2	0.0019531	512 s	(256秒开/256秒关)
M 100.3	0.0009765	1024 s	(512秒开/512秒关)
M 100.4	0.0004882	2048 s	(1024秒开/1024秒关)
M 100.5	0.0002441	4096 s	(2048秒开/2048秒关)
M 100.6	0.000122	8192 s	(4096秒开/4096秒关)
M 100.7	0.000061	16384 s	(8192秒开/8192秒关)

存储器MB 101的位信号状态

扫描周期	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位	时间值(单位:毫秒)
0	0	0	0	0	0	0	**0**	0	250
1	0	0	0	0	0	0	**0**	1	250
2	0	0	0	0	0	0	1	0	250
3	0	0	0	0	0	0	1	1	250
4	0	0	0	0	0	1	**0**	0	250
5	0	0	0	0	0	1	**0**	1	250
6	0	0	0	0	0	1	1	0	250
7	0	0	0	0	0	1	1	1	250
8	0	0	0	0	1	0	**0**	0	250
9	0	0	0	0	1	0	**0**	1	250
10	0	0	0	0	1	0	1	0	250
11	0	0	0	0	1	0	1	1	250
12	0	0	0	0	1	1	**0**	0	250

MB 101 (M 101.1)第1位的信号状态

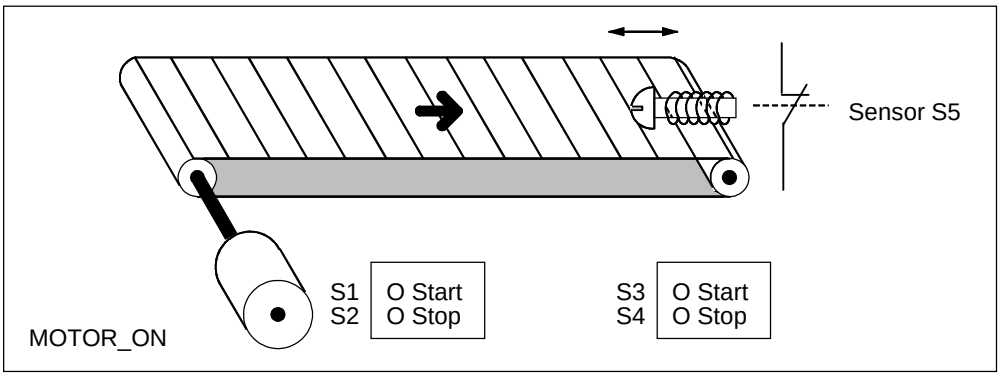
频率 = 1/T = 1/1 s = 1赫兹



实例：位逻辑指令

实例1：控制传送带

下图显示可用电动方式激活的传送带。在传送带的开始位置有两个按钮开关：用于启动的S1和用于停止的S2。在传送带末端也有两个按钮开关：用于启动的S3和用于停止的S4。可从任何一端启动或停止传送带。此外，当传送带上的部件到达终点时，传感器S5将停止传送带。



绝对地址和符号编程

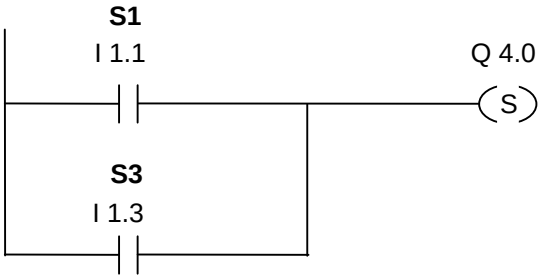
您可编写程序使用**绝对地址**或代表传送带系统各种组件的**符号**来控制传送带。

需要制定一个符号表，以建立所选择的符号与绝对地址的联系(参见STEP 7在线帮助)。

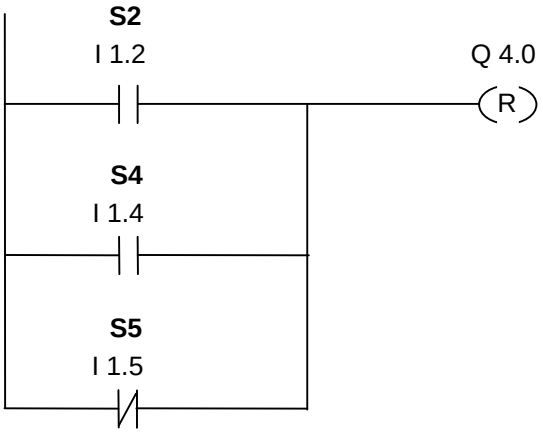
系统组件	绝对地址	符号	符号表
按钮启动开关	I 1.1	S1	I 1.1 S1
按钮停止开关	I 1.2	S2	I 1.2 S2
按钮启动开关	I 1.3	S3	I 1.3 S3
按钮停止开关	I 1.4	S4	I 1.4 S4
传感器	I 1.5	S5	I 1.5 S5
电机	Q 4.0	MOTOR_ON	Q 4.0 MOTOR_ON

控制传送带的梯形图程序

程序段1：按下任一启动开关打开电机。

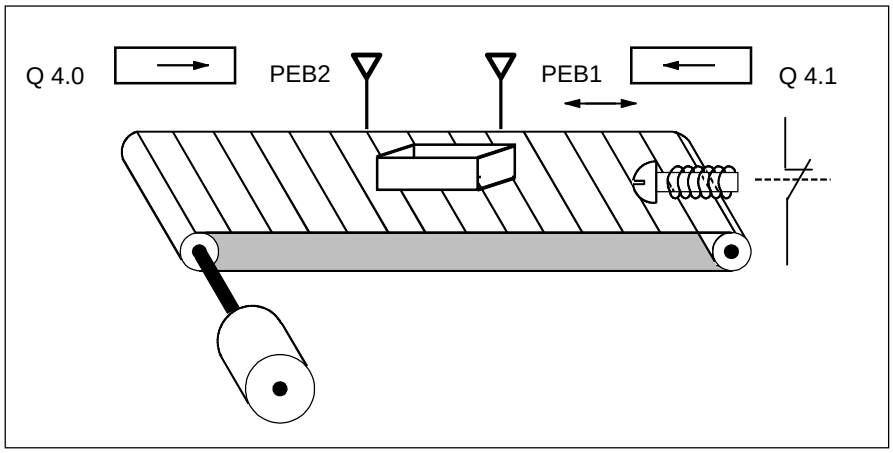


程序段2：按下任一停止开关或打开传送带尾部的常闭触点以关闭电机。



实例2：检测传送带方向

下图显示配备两个光电屏障(PEB1和PEB2)的传送带，这两个光电屏障专用于检测包裹在传送带上移动的方向。每个光电屏障的功能类似常开触点。



绝对地址和符号编程

您可编写程序以使用**绝对地址**或代表传送带系统各种组件的**符号**来激活传送带系统的方向显示。

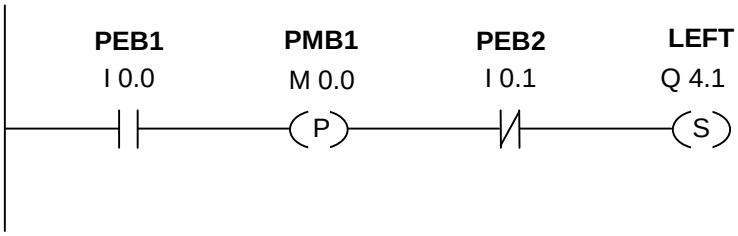
需要制定一个符号表，以建立所选择的符号与绝对地址的联系(参见STEP 7在线帮助)。

系统组件	**绝对地址**	**符号**	**符号表**
光电屏障1	I 0.0	PEB1	I 0.0 PEB1
光电屏障2	I 0.1	PEB2	I 0.1 PEB2
显示向右移动	Q 4.0	RIGHT	Q 4.0 RIGHT

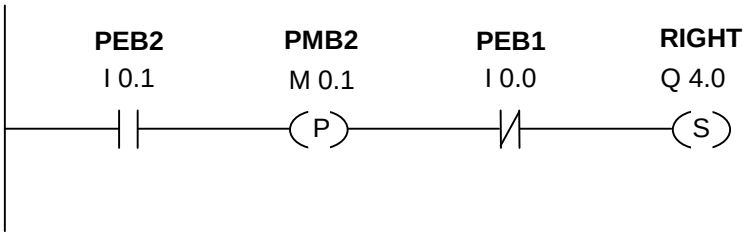
显示向左移动	Q 4.1	LEFT	Q 4.1	LEFT
脉冲存储器位1	M 0.0	PMB1	M 0.0	PMB1
脉冲存储器位2	M 0.1	PMB2	M 0.1	PMB2

用于检测传送带方向的梯形图程序

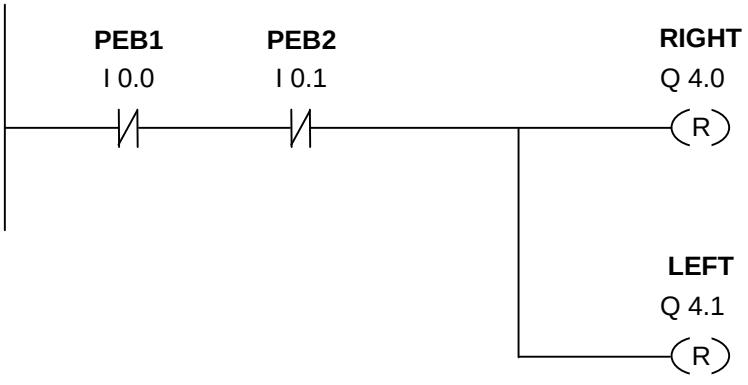
程序段1：如果输入I 0.0处信号状态从0过渡到1(上升沿)，与此同时，输入I 0.1处信号状态为0，则传送带上的包裹向左移动。



程序段2：如果输入I 0.1处信号状态从0过渡到1(上升沿)，与此同时，输入I 0.0处信号状态为0，则传送带上的包裹向右移动。如果光电屏障之一被中断，则表明屏障之间有包裹。



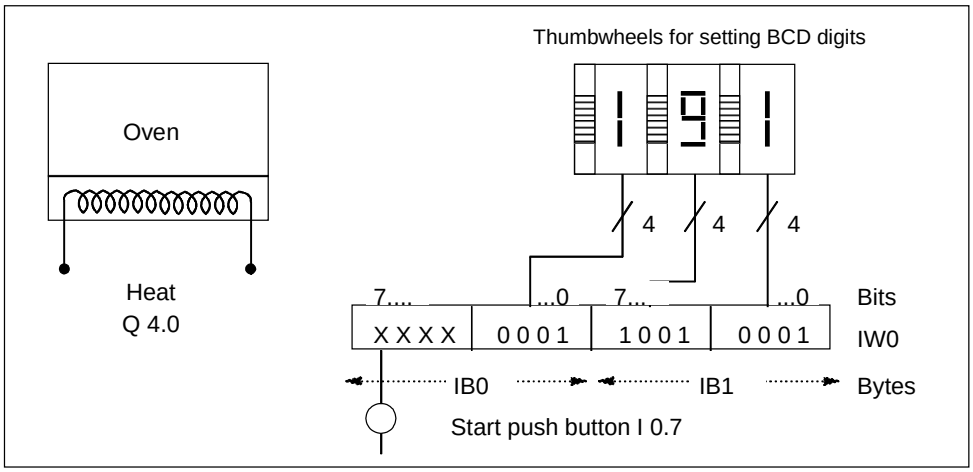
程序段3：如果两个光电屏障都未中断，则表明屏障之间没有包裹。方向指针关闭。



实例：字逻辑指令

加热烘炉

烘炉操作员通过按启动按钮来启动烘炉加热。操作员可用图中所示的码盘开关来设置加热的时间。操作员设置的值以二进制编码的十进制(BCD)格式显示，单位为秒。



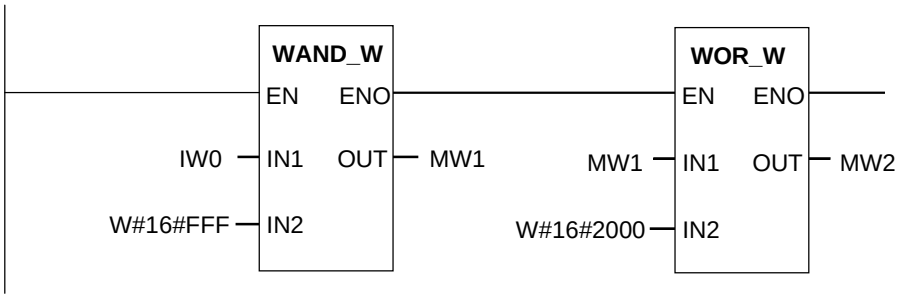
系统组件	**绝对地址**
启动按钮	I 0.7
个位码盘	I 1.0 到 I 1.3
十位指轮开关	I 1.4 到 I 1.7
百位码盘	I 0.0 到 I 0.3
加热启动	Q 4.0

梯形图程序

程序段1：如果定时器正在运行，则打开加热器。

程序段2：如果定时器正在运行，**返回**指令结束此处的处理。

程序段3：屏蔽输入位I 0.4到I 0.7 (即，将它们复位为0)。指轮开关输入的这些位未被使用。16位指轮开关输入根据**(字)与运算**指令与W#16#0FFF组合。结果载入存储器字MW1中。为了设置时间基准的秒数，预设值根据**(字)或运算**指令与W#16#2000组合，将位13设置为1，并将位12复位为0。

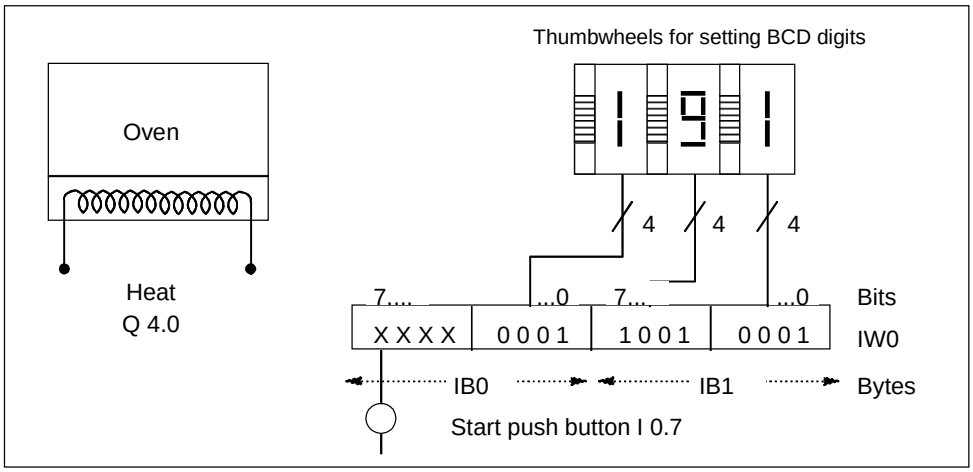


程序段4：如果按下启动按钮，则将定时器T1作为扩展脉冲定时器启动，并作为预设值存储器字MW2装载(来自于上述逻辑)。

实例：字逻辑指令

加热烘炉

烘炉操作员通过按启动按钮来启动烘炉加热。操作员可用图中所示的码盘开关来设置加热的时间。操作员设置的值以二进制编码的十进制(BCD)格式显示，单位为秒。



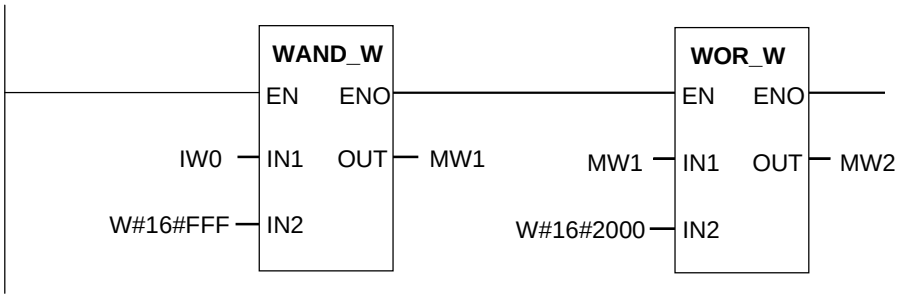
系统组件	**绝对地址**
启动按钮	I 0.7
个位码盘	I 1.0 到 I 1.3
十位指轮开关	I 1.4 到 I 1.7
百位码盘	I 0.0 到 I 0.3
加热启动	Q 4.0

梯形图程序

程序段1：如果定时器正在运行，则打开加热器。

程序段2：如果定时器正在运行，**返回**指令结束此处的处理。

程序段3：屏蔽输入位I 0.4到I 0.7 (即，将它们复位为0)。指轮开关输入的这些位未被使用。16位指轮开关输入根据**(字)与运算**指令与W#16#0FFF组合。结果载入存储器字MW1中。为了设置时间基准的秒数，预设值根据**(字)或运算**指令与W#16#2000组合，将位13设置为1，并将位12复位为0。



程序段4：如果按下启动按钮，则将定时器T1作为扩展脉冲定时器启动，并作为预设值存储器字MW2装载(来自于上述逻辑)。

实例：整型数学运算指令

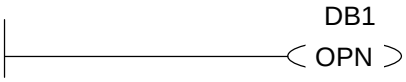
解决数学问题

实例程序显示了如何使用三个整数数学运算指令来产生与下列方程式相同的结果：

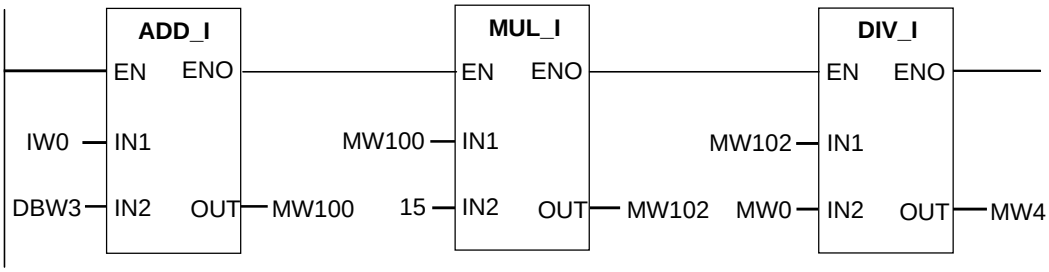
MW4 = ((IW0 + DBW3) x 15) / MW0

梯形图程序

程序段1：打开数据块DB1。



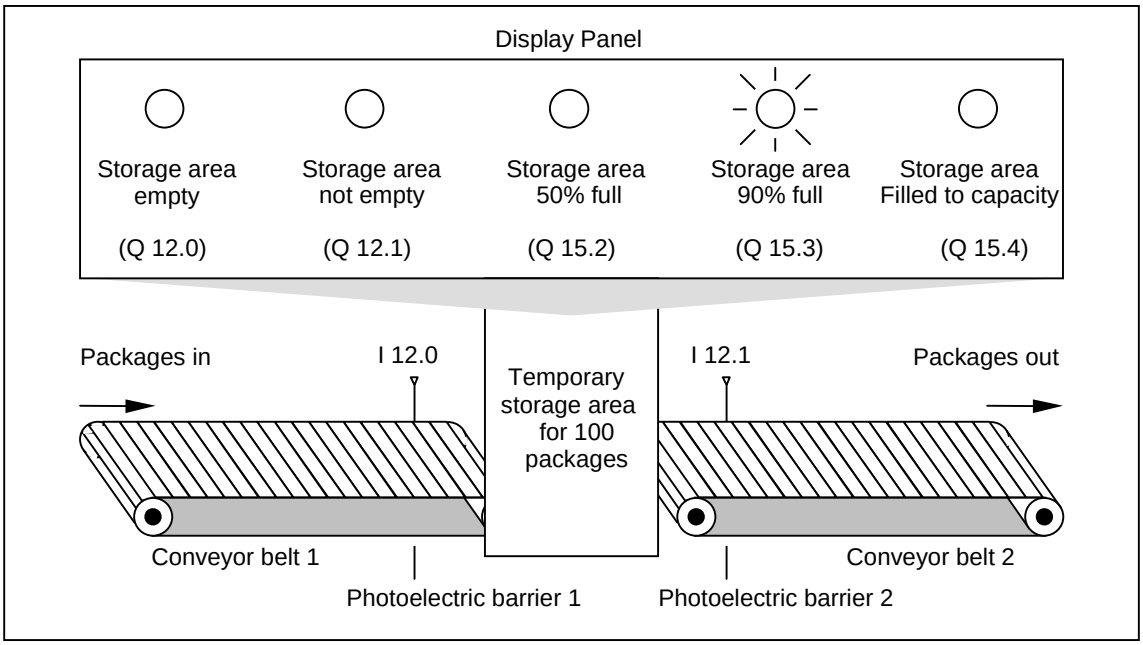
程序段2：输入字IW0加到共享数据字DBW3(必须定义和打开数据块)，总和被载入存储器字MW100。然后，MW100乘以15，结果存储到存储器字MW102中。MW102除以MW0，结果存储到MW4中。



实例：计数器和比较指令

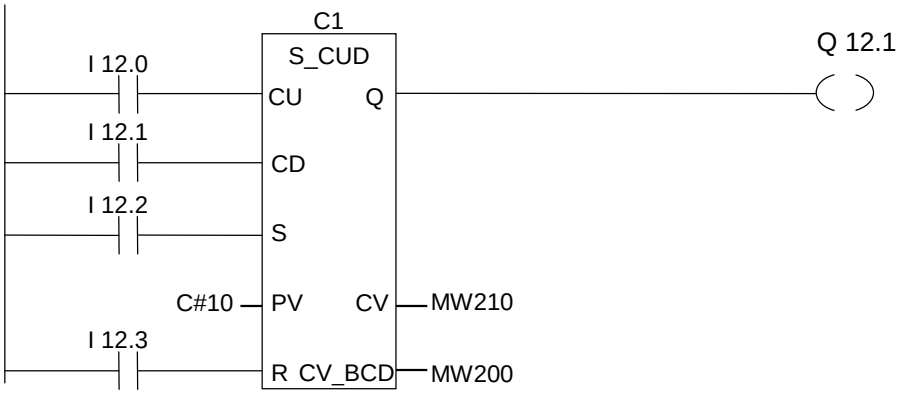
带计数器和比较器的存储区域

下图显示了具有两个传送带且在传送带之间有临时存储区域的系统。传送带1将包裹传送到存储区域。存储区域附近的传送带1末端的光电屏障确定向存储区域传送的包裹数量。传送带2会将包裹从临时存储区域传输到装载码头，而卡车在此将包裹发送给客户。存储区域附近的传送带2末端的光电屏障确定离开存储区域而转向装载码头的包裹数量。带五个指示灯的显示面板将指示临时存储区域的填充量。



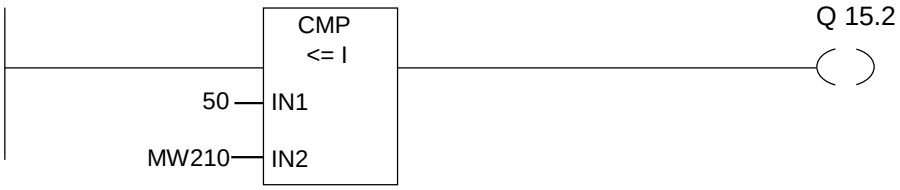
激活显示面板上的指示灯的梯形图程序

程序段1：计数器C1对输入CU处每次从"0"到"1"的信号改变都进行正计数，而对输入CD处每次从"0"到"1"的信号改变都进行倒计数。对于输入S处从"0"到"1"的信号改变，计数器值被设置为值PV。输入R处从"0"到"1"的信号改变将计数器值复位为"0"。MW200包含C1的当前计数器值。Q12.1指示"存储区域非空"。

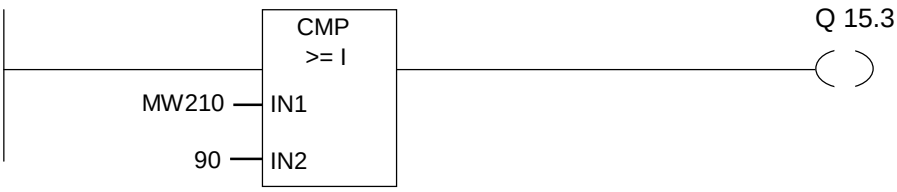


程序段2：Q12.0表明"存储区域为空"。

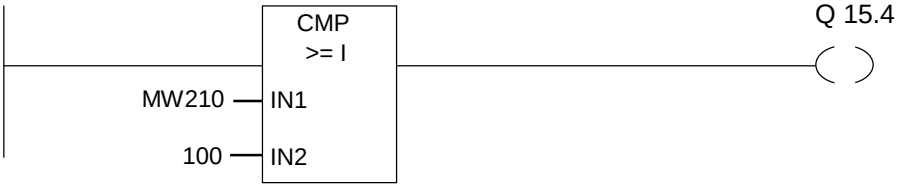
程序段3：如果50小于等于计数器值(换句话说，如果当前计数器值大于等于50)，则表示"存储区域50%满"的指示灯变亮。



程序段4：程序段4：如果计数器值大于或等于90，则表示"存储区域90%满"的指示灯变亮。



程序段5：如果计数器值大于或等于100，则表示"存储区域满"的指示灯变亮。



实例：定时器指令

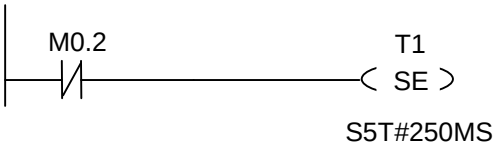
时钟脉冲发生器

当需要生成定期重复的信号时，可使用时钟脉冲发生器或闪烁继电器。时钟脉冲发生器在控制指示灯闪烁的信号系统中很常见。

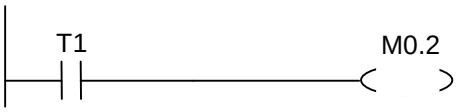
当使用S7-300时，您可用特殊组织块中的时间处理功能来执行时钟脉冲发生器功能。但下列梯形图程序中显示的实例说明的是使用定时器功能产生时钟脉冲。实例程序显示如何通过使用定时器实现任意的时钟脉冲发生器。

产生时钟脉冲(脉冲占空比1:1)的梯形图程序

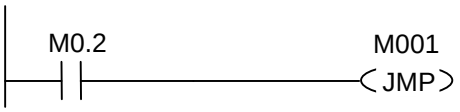
程序段1：如果定时器T1的信号状态为0，将时间值250毫秒载入T1，并将T1作为扩展脉冲定时器启动。



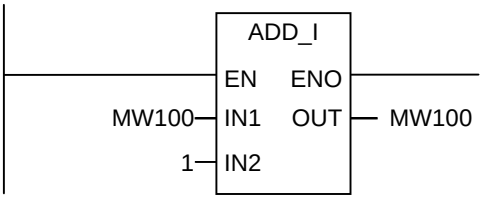
程序段2：该定时器的状态临时保存在一个辅助存储器符号中。



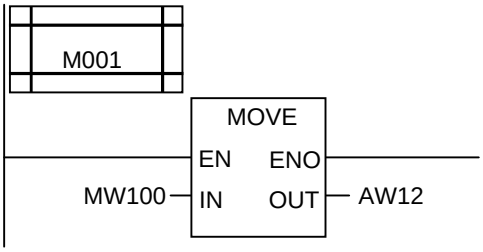
程序段3：如果定时器T1的信号状态为1，则跳转至跳转标签M001。



程序段4：定时器T1超时后，存储器字100增加1。

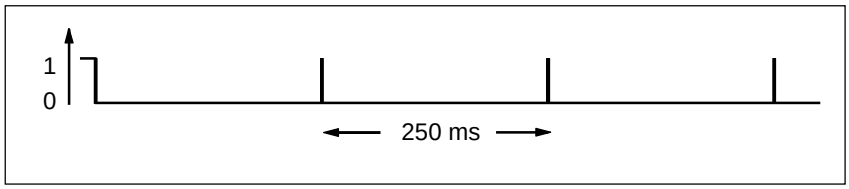


程序段5：**MOVE**指令允许在输出Q12.0到Q13.7输出不同的时钟频率。



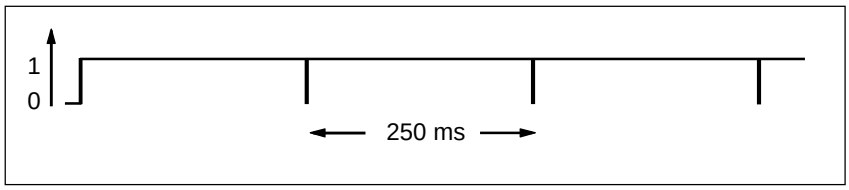
信号检查

定时器T1的信号检查为opener M0.2生成以下逻辑运算(RLO)结果。



一旦定时时间到，就会重新启动定时器。因此，由 |/| / |/| M0.2进行的信号检查只简单产生信号状态1。

RLO取反(反向)：



每隔250毫秒RLO位为0。忽略跳转且存储器字MW100的内容增加1。

实现特定频率

通过存储器字节MB101和MB100的单个位，可以实现下列频率：

MB101/MB100的位	**频率(赫兹)**	**持续时间**	
M 101.0	2.0	0.5s	(250毫秒开/ 250毫秒关)
M 101.1	1.0	1 s	(0.5秒开/ 0.5秒关)
M 101.2	0.5	2s	(1秒开/ 1秒关)
M 101.3	0.25	4s	(2秒开/ 2秒关)
M 101.4	0.125	8s	(4秒开/ 4秒关)
M 101.5	0.0625	16s	(8秒开/ 8秒关)

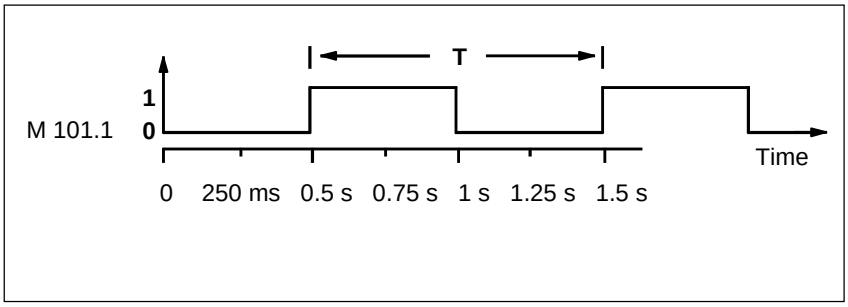
M 101.6	0.03125	32s	(16秒开/ 16秒关)
M 101.7	0.015625	64s	(32秒开/ 32秒关)
M 100.0	0.0078125	128 s	(64秒开/64秒关)
M 100.1	0.0039062	256 s	(128秒开/128秒关)
M 100.2	0.0019531	512 s	(256秒开/256秒关)
M 100.3	0.0009765	1024 s	(512秒开/512秒关)
M 100.4	0.0004882	2048 s	(1024秒开/1024秒关)
M 100.5	0.0002441	4096 s	(2048秒开/2048秒关)
M 100.6	0.000122	8192 s	(4096秒开/4096秒关)
M 100.7	0.000061	16384 s	(8192秒开/8192秒关)

存储器MB 101的位信号状态

扫描周期	第7位	第6位	第5位	第4位	第3位	第2位	第1位	第0位	时间值(单位:毫秒)
0	0	0	0	0	0	0	**0**	0	250
1	0	0	0	0	0	0	**0**	1	250
2	0	0	0	0	0	0	1	0	250
3	0	0	0	0	0	0	1	1	250
4	0	0	0	0	0	1	**0**	0	250
5	0	0	0	0	0	1	**0**	1	250
6	0	0	0	0	0	1	1	0	250
7	0	0	0	0	0	1	1	1	250
8	0	0	0	0	1	0	**0**	0	250
9	0	0	0	0	1	0	**0**	1	250
10	0	0	0	0	1	0	1	0	250
11	0	0	0	0	1	0	1	1	250
12	0	0	0	0	1	1	**0**	0	250

MB 101 (M 101.1)第1位的信号状态

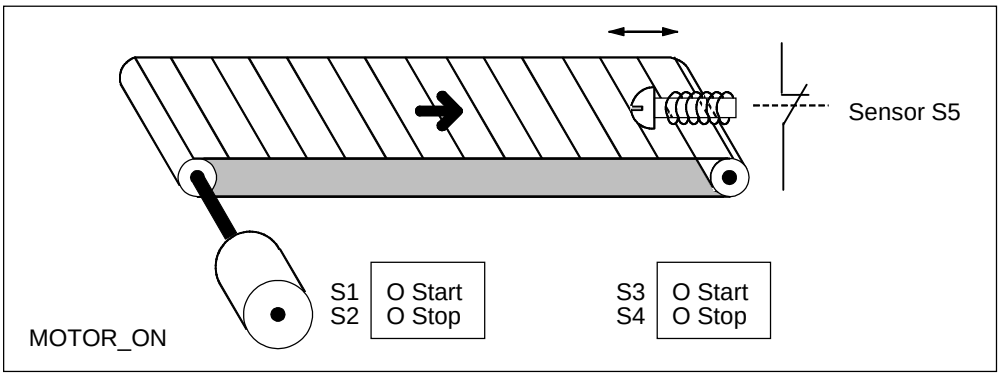
频率 = 1/T = 1/1 s = 1赫兹



实例：位逻辑指令

实例1：控制传送带

下图显示可用电动方式激活的传送带。在传送带的开始位置有两个按钮开关：用于启动的S1和用于停止的S2。在传送带末端也有两个按钮开关：用于启动的S3和用于停止的S4。可从任何一端启动或停止传送带。此外，当传送带上的部件到达终点时，传感器S5将停止传送带。



绝对地址和符号编程

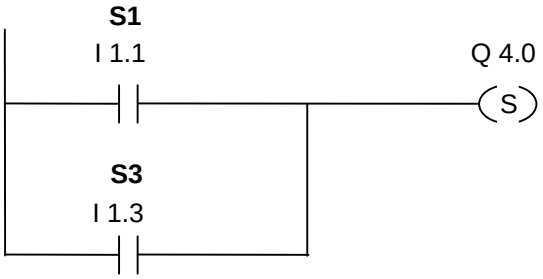
您可编写程序使用**绝对地址**或代表传送带系统各种组件的**符号**来控制传送带。

需要制定一个符号表，以建立所选择的符号与绝对地址的联系(参见STEP 7在线帮助)。

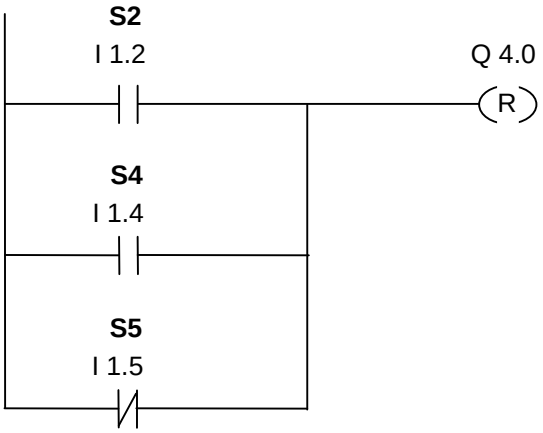
系统组件	绝对地址	符号	符号表
按钮启动开关	I 1.1	S1	I 1.1 S1
按钮停止开关	I 1.2	S2	I 1.2 S2
按钮启动开关	I 1.3	S3	I 1.3 S3
按钮停止开关	I 1.4	S4	I 1.4 S4
传感器	I 1.5	S5	I 1.5 S5
电机	Q 4.0	MOTOR_ON	Q 4.0 MOTOR_ON

控制传送带的梯形图程序

程序段1：按下任一启动开关打开电机。

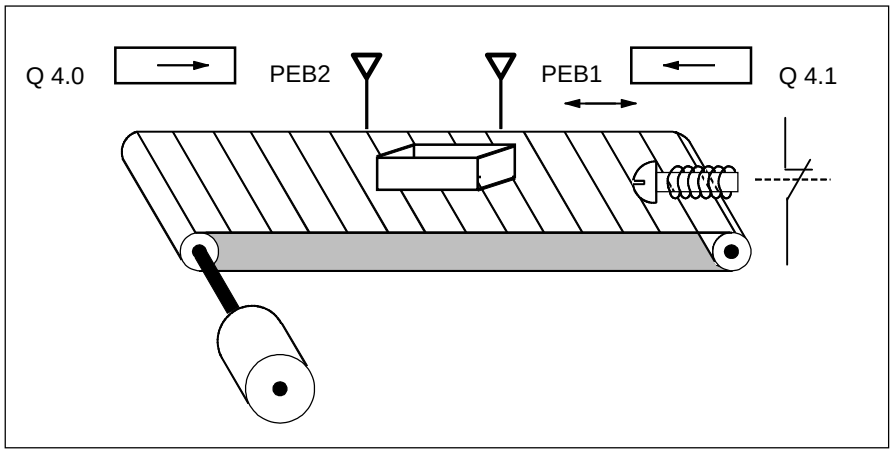


程序段2：按下任一停止开关或打开传送带尾部的常闭触点以关闭电机。



实例2：检测传送带方向

下图显示配备两个光电屏障(PEB1和PEB2)的传送带，这两个光电屏障专用于检测包裹在传送带上移动的方向。每个光电屏障的功能类似常开触点。



绝对地址和符号编程

您可编写程序以使用**绝对地址**或代表传送带系统各种组件的**符号**来激活传送带系统的方向显示。

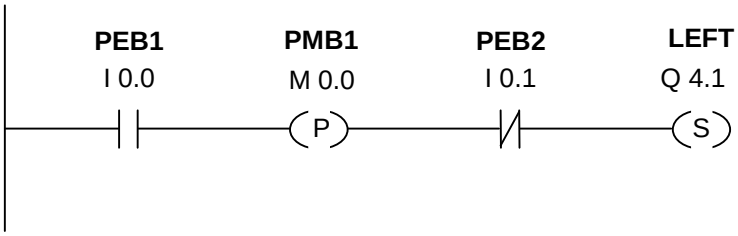
需要制定一个符号表，以建立所选择的符号与绝对地址的联系(参见STEP 7在线帮助)。

系统组件	**绝对地址**	**符号**	**符号表**
光电屏障1	I 0.0	PEB1	I 0.0 PEB1
光电屏障2	I 0.1	PEB2	I 0.1 PEB2
显示向右移动	Q 4.0	RIGHT	Q 4.0 RIGHT

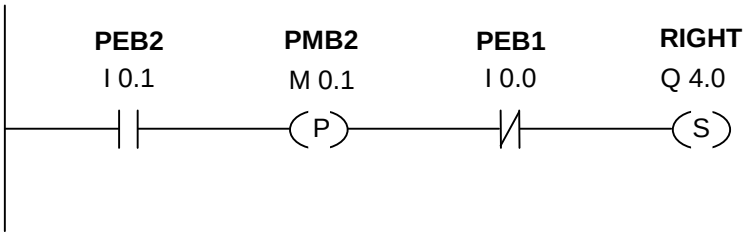
显示向左移动	Q 4.1	LEFT	Q 4.1	LEFT
脉冲存储器位1	M 0.0	PMB1	M 0.0	PMB1
脉冲存储器位2	M 0.1	PMB2	M 0.1	PMB2

用于检测传送带方向的梯形图程序

程序段1：如果输入I 0.0处信号状态从0过渡到1(上升沿)，与此同时，输入I 0.1处信号状态为0，则传送带上的包裹向左移动。



程序段2：如果输入I 0.1处信号状态从0过渡到1(上升沿)，与此同时，输入I 0.0处信号状态为0，则传送带上的包裹向右移动。如果光电屏障之一被中断，则表明屏障之间有包裹。



程序段3：如果两个光电屏障都未中断，则表明屏障之间没有包裹。方向指针关闭。