



“十三五”职业教育国家规划教材



西门子S7-1200 PLC应用技术

项目教程

(第2版)

主 编
副主编

吴繁红
雷 宁
陈 岭
陆 斌



扫码看微课

- ◆ 结合西门子公司的培训认证课程编写
- ◆ 图文并茂、实用、易懂、可操作性强
- ◆ 配套资源包括微课视频、电子课件等



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.php.com.cn>

西门子S7-1200 PLC应用技术项目教程 (第2版)

电子工业出版社

项目 5 运料小车往返控制

能力目标

1. 运料小车往返控制电路的硬件组态与接线方式。
2. 正确使用定时器、计数器指令编写控制程序。
3. PLC 应用技术项目设计的过程和方法。

知识目标

1. S7-1200 PLC 计数器指令的格式、功能及典型应用。
2. 顺序控制程序的基本概念和设计方法。
3. 手/自动运行切换程序的设计。

5.1 项目导入

PLC 是在传统的继电控制系统的基础上发展而来的，继电控制系统是典型的数字量控制系统。数字量控制系统梯形图的设计方法分为两种，即经验设计法和顺序控制设计法。前面的项目中梯形图的设计方法就属于经验设计法，这种设计方法是根据被控对象对控制系统的具体要求，不断地修改和完善梯形图，没有普遍的规律可循，具有很大的试探性和随意性，最后结果不是唯一的，设计所用的时间、设计的质量与设计者的经验有关，因此称为经验设计法。同时，用经验设计法设计出的梯形图因人而异，没有可普遍遵循的规律，往往很难阅读，给系统维修和改进带来很大困难。

在生产实践中经常可见顺序控制的运动规律，如多工步组合机床的运动。采用顺序控制的车床实物图如图 5-1 所示，其用于黑色金属、有色金属、非金属工件的内外圆柱面、圆锥面、旋转表面等的粗、精车削加工，能车削各种常用公制、寸制、模数、径节的螺纹及多头

螺纹，还能进行钻孔、铰孔和油槽的拉削加工等，进行的是典型的顺序控制运动。

顺序控制设计法是指使各个执行机构按照生产工艺预先规定的顺序，在各个输入信号作用下，根据内部状态和时间顺序，在生产过程中自动、有序地进行操作的设计方法。使用顺序控制设计法，首先根据系统的工艺过程和运动规律画出顺序功能图；然后根据顺序功能图编写程

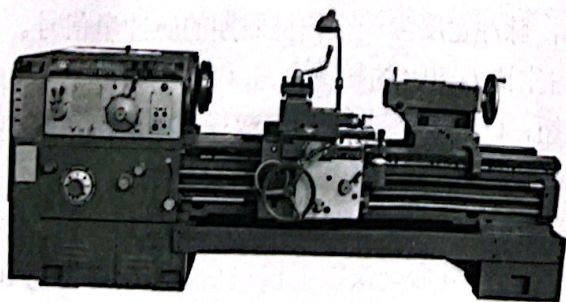


图 5-1 采用顺序控制的车床实物图

序。这种方法有一定的设计步骤和规律,初学者很容易学会,有经验的工程师采用这种方法也可以提高设计效率。采用顺序控制设计法设计程序可使程序的阅读、调试、修改十分方便。

本项目要设计一个运料小车自动往返控制器,具体设计要求如下。

(1) 设计一个运料小车自动往返控制器,小车自动运行状态如下:当 SA 闭合时,系统启动后小车先在原位进行装料,15s 后装料停止,小车右行;右行至右侧行程开关 SQ2 处停止,进行卸料,10s 后卸料停止,小车左行;左行至左侧行程开关 SQ1 处停止,进行装料,如此循环 3 次停止。在运行过程中,无论小车在什么位置,按下停止按钮,小车运行到装料处方可停止。

(2) 当 SA 断开时,小车只能手动控制:① 当按下点动前进按钮后,小车接通前进电动机,点动前进至右侧行程开关 SQ2 处停止;② 当按下点动后退按钮后,小车接通后退电动机,点动后退至左侧行程开关 SQ1 处停止。

5.2 项目分析

运料小车的往返控制设计是一个典型的顺序控制设计,顺序控制过程包括装料、右行、卸料、左行 4 个状态,各个状态按照一定的规律循环转换。因此,本项目宜采用顺序控制设计法。

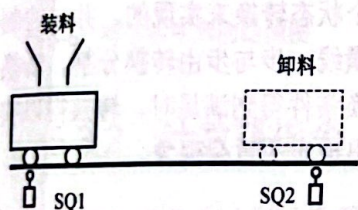


图 5-2 小车自动运料示意图

装料和卸料的状态受时间控制,因此设计中要用到定时器指令,小车右行、左行状态的结束由行程开关的位置决定。本项目中需要统计循环次数,因此还要用到计数器指令。小车自动运料示意图如图 5-2 所示。

本项目设计中需要用到的指令有基本逻辑指令中的触点指令、置位指令、复位指令及定时器指令,此外还需要用计数器指令来统计循环次数。

5.3 相关知识

5.3.1 顺序控制设计法

1. 顺序功能图法

在工业控制领域,许多场合中要应用顺序控制的方式进行控制。顺序控制是指使生产机械按根据生产工艺的要求预先安排的顺序自动动作。

顺序功能图是描述控制系统的控制过程、功能和特性的一种图形,也是设计 PLC 顺序控制程序的有力工具。

顺序功能图法就是依据顺序功能图设计 PLC 顺序控制程序的方法,基本思想是将系统的一个工作周期分解成若干个顺序相连的阶段,即步。顺序功能图主要由步、与步相关的动作(或命令)、有向连线、转换和转换条件组成。



1) 步

在顺序功能图中, 把系统循环工作过程分解成若干个顺序相连的阶段, 即步。步用矩形框表示, 框内的数字表示步的编号。在控制过程中的某给定时刻, 一个步可以是活动的也可以是非活动的。当步处于活动状态时, 称为活动步; 反之, 称为非活动步。控制过程开始阶段的活动步与初始状态对应, 称为起始步, 用双线矩形框表示, 每个顺序功能图中至少应有一个起始步。

2) 与步相关的动作 (或命令)

控制系统的每个步都要完成某些动作 (或命令), 当某步处于活动状态时, 与该步相关的动作 (或命令) 被执行; 反之, 动作 (或命令) 不被执行。与该步相关的动作 (或命令) 用矩形框表示, 框内的文字或符号表示动作 (或命令) 的内容, 该矩形框应与相应步的矩形框相连。在顺序功能图中, 动作 (或命令) 可分为非存储型和存储型两类。当相应步活动时, 动作 (或命令) 被执行, 当相应步不活动时, 动作 (或命令) 如果返回到该步活动前的状态, 则该动作 (或命令) 是非存储型的; 如果继续保存它的状态, 则该动作 (或命令) 是存储型的。

3) 有向连线

在顺序功能图中, 会发生步的活动状态的进展, 用有向连线表示, 它将步连接到转换并将转换连接到步。步的活动状态的进展按有向连线规定的线路进行, 有向连线是垂直或水平的, 按习惯进展的方向总是从上到下或从左到右, 如果不遵守上述习惯就必须加箭头, 必要时为了易于理解也可加箭头, 箭头表示步的活动状态的进展方向。

4) 转换和转换条件

在顺序功能图中, 步的活动状态的进展是由一个或多个状态转换来实现的, 并与控制过程的发展相对应。转换符号是一根与有向连线垂直的短横线, 步与步由转换分割。转换条件在转换符号旁边用文字或符号说明。当两步之间的转换条件得到满足时, 转换得以实现, 即上一步的活动结束而下一步的活动开始, 因此不会出现步的重叠现象。

2. 顺序功能图的基本结构

依据步的活动状态的进展形式, 顺序功能图有以下几种基本结构。

1) 单序列结构

单序列由一系列相继激活的步组成, 每个步的后面仅有一个转换, 每个转换后面仅有一个步, 如图 5-3 所示。

2) 选择序列结构

(1) 选择序列的开始称为分支。某个步的后面有几个步, 当满足不同的转换条件时, 转向不同的步, 如图 5-4 (a) 所示。

(2) 选择序列的结束称为合并。几个选择序列合并到同一个序列上, 各个序列上的步在各自转换条件满足时转换到同一个步, 如图 5-4 (b) 所示。

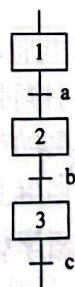


图 5-3 单序列结构图

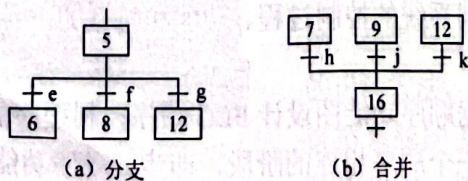


图 5-4 选择序列结构图

3) 并行序列结构

(1) 并行序列的开始称为分支。当转换的实现导致几个序列同时激活时, 这些序列称为并行序列, 它们被同时激活后, 每个序列中步的活动状态的进展将是独立的, 如图 5-5 (a) 所示。

(2) 并行序列的结束称为合并。在并行序列中, 处于水平双线以上的各个步都为活动步, 当转换条件满足时, 同时转换到同一个步, 如图 5-5 (b) 所示。

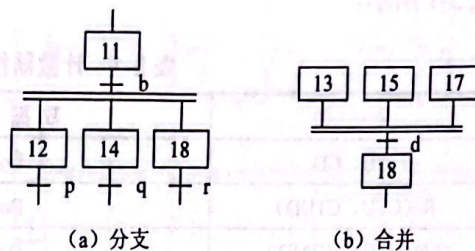


图 5-5 并行序列结构图

4) 有子步序列的结构

根据需要, 在顺序功能图中, 某个步又可分为几个子步, 如图 5-6 所示。图 5-6 (a)

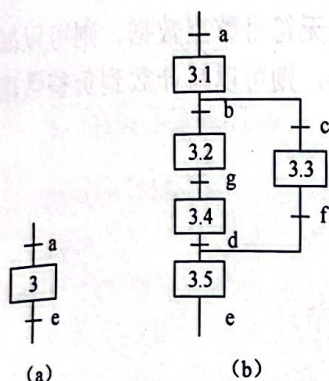


图 5-6 有子步序列的结构图

是以简略形式表示的步 3, 图 5-6 (b) 将步 3 细分为 5 个子步, 详细表示了步 3 的具体细节。这种步的详细表示方法 (子步) 可以使系统的设计者在进行总体设计时以更加简洁的方式表达系统的总体功能和概貌, 从功能入手对整个系统简要地进行全面描述。在总体设计被确认后, 再进行深入的细节设计。这样就可以使系统设计者在设计初期抓住系统的主要矛盾而免于陷入某些细节的纠缠, 减少总体设计的错误。同时, 也便于设计人员和其他相关人员进行设计思想的沟通, 便于程序的分工设计和检查及调试, 从而可以缩短程序设计时间和调试时间。

5.3.2 计数器指令

S7-1200 PLC 有 3 种计数器: 加计数器 (CTU)、减计数器 (CTD) 和加减计数器 (CTUD)。它们都属于软件计数器, 其最大计数速率受其所在的组织块执行速率的限制, 如果需要速率更高的计数器, 则可以使用 CPU 内置的高速计数器。在调用计数器指令时, 需要生成用于保存计数器数据的背景数据块。3 种计数器指令的符号如图 5-7 所示。

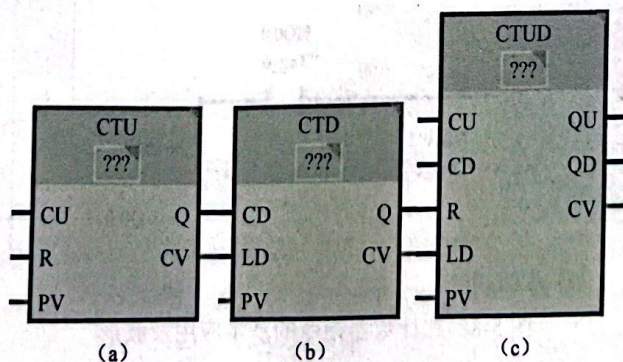


图 5-7 3 种计数器指令的符号

CU 和 CD 分别是加计数和减计数, 当 CU 或 CD 从 0 变为 1 时, 当前计数值 CV 加 1 或减 1。当复位参数 R 为 1 时, 计数器被复位, CV 被清 0, 计数器的 Q 输出变为 0。LD 为 1,

将预置计数值 PV 装载到计数器中作为当前计数值。计数器指令的参数、数据类型及说明如表 5-1 所示。

表 5-1 计数器指令的参数、数据类型及说明

参 数	数 据 类 型	说 明
CU、CD	Bool	加计数、减计数, 按加 1、减 1 计数
R (CTU、CTUD)	Bool	将计数值重置为 0
LD (CTD、CTUD)	Bool	预置计数值的装载控制
PV	SInt、Int、DInt、USInt、UInt、UDInt	预置计数值
Q、QU	Bool	当计数器当前计数值大于预置计数值时为 1
QD	Bool	当 $CV \leq 0$ 时为真
CV	SInt、Int、DInt、USInt、UInt、UDInt	当前计数值

计数值的数值范围取决于所选的数据类型: 如果计数值是无符号整型数据, 则可以减计数到零或加计数到范围限值; 如果计数值是有符号整型数据, 则可以减计数到负整数限值或加计数到正整数限值。

用户程序中使用以下数据类型。

- (1) 对于 SInt 或 USInt 数据类型, 计数器指令占用 3 字节。
- (2) 对于 Int 或 UInt 数据类型, 计数器指令占用 6 字节。
- (3) 对于 DInt 或 UDInt 数据类型, 计数器指令占用 12 字节。

1. 加计数器指令

加计数器指令: 当 CU 从 0 变为 1 时, 加计数器会使 CV 加 1, 直到 CV 达到指定的数据类型取值范围的上限值, 此后, CU 变化, CV 不再增加。如果 CV 大于或等于 PV, 则加计数器的 Q 输出为 1。如果 R 从 0 变为 1, 则 CV 复位为 0。在第一次执行程序时, CV 被清零。加计数器指令的基本应用及波形如图 5-8 所示。

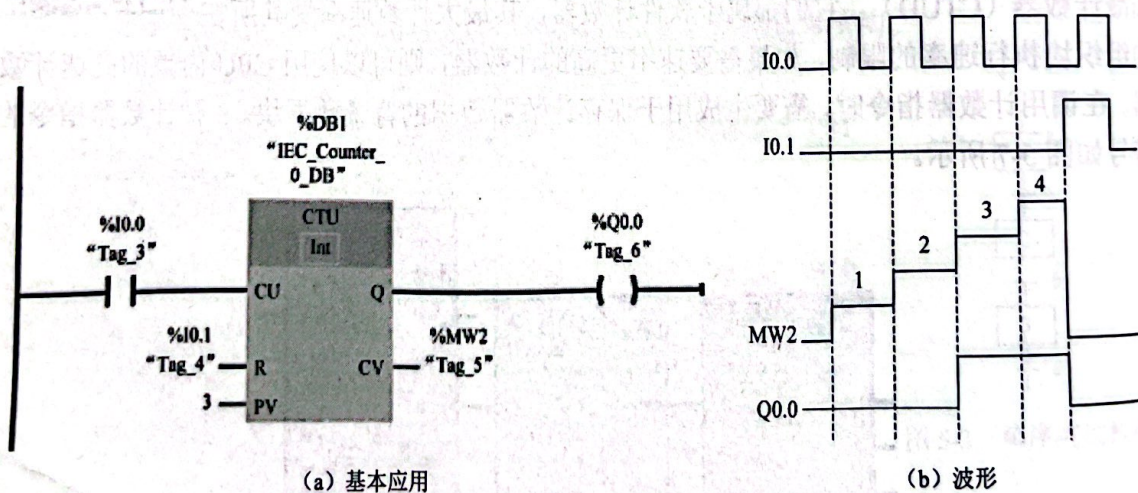


图 5-8 加计数器指令的基本应用及波形

2. 减计数器指令

减计数器指令: 当 LD 从 0 变为 1 时, PV 将作为新的 CV 被装载到减计数器中, Q 输出为 0。当 CD 由 0 变为 1 时, 减计数器会使 CV 减 1。如果 CV 小于或等于 0, 则减计数器

的Q输出为1。第一次执行程序时，CV被清零。减计数器指令的基本应用及波形如图5-9所示。

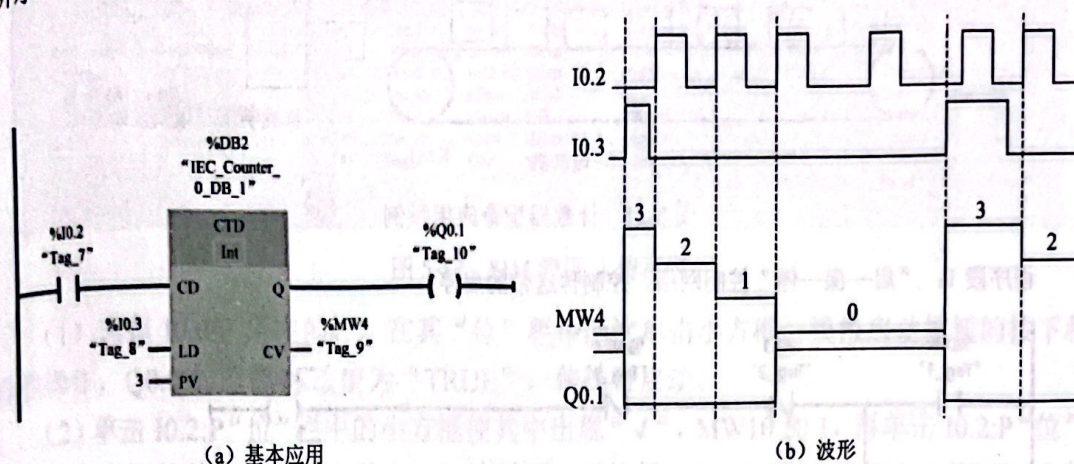


图 5-9 减计数器指令的基本应用及波形

3. 加减计数器指令

加减计数器指令：当CU或CD从0变为1时，加减计数器会使CV加1或减1。

如果CV大于或等于PV，则加减计数器的QU输出为1；如果CV小于或等于0，则加减计数器的QD输出为1。

如果LD从0变为1，则PV将作为新的CV被装载到加减计数器中。

如果R从0变为1，则CV复位为0。加减计数器指令的基本应用及波形如图5-10所示。

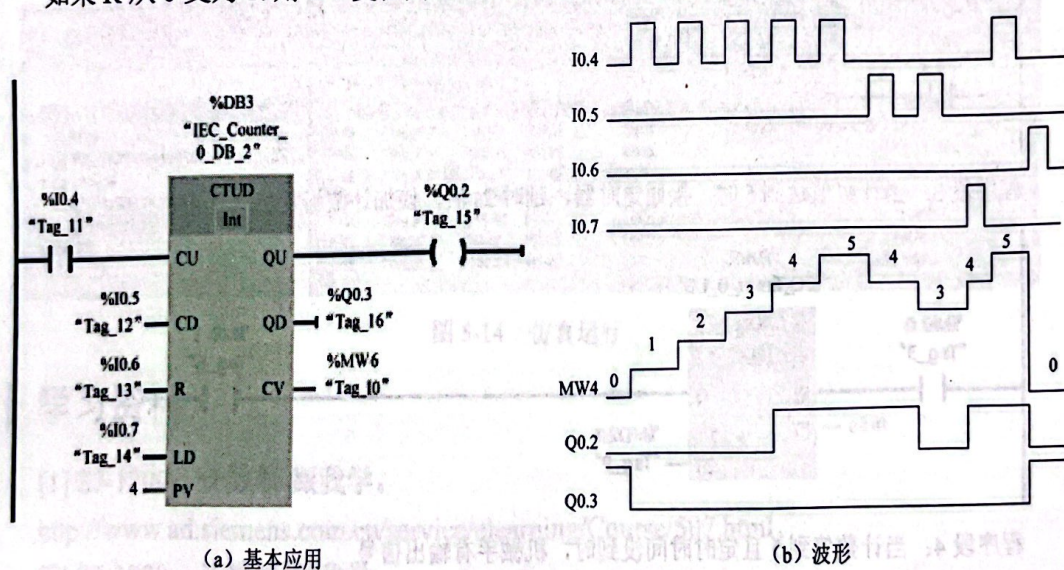


图 5-10 加减计数器指令的基本应用及波形

【应用举例】由电动机带动传送带 KM1 启停，I0.0 接传送带的启动按钮，I0.1 接传送带的停止按钮，I0.2 接产品检测器 PH，电动机接 Q0.0，Q0.1 控制机械手动动作。传送带开始运行后，产品通过产品检测器 PH，检测到信号，每检测到 5 个产品机械手动动作 1 次，机械手动动作后，延时 5s，机械手电磁铁切断，重新开始下一次计数，如图 5-11 所示，其梯形图如图 5-12 所示。

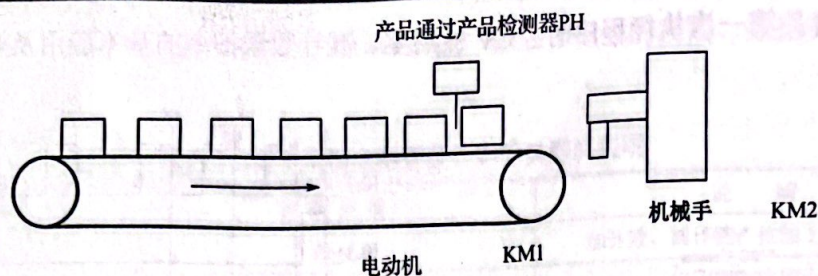
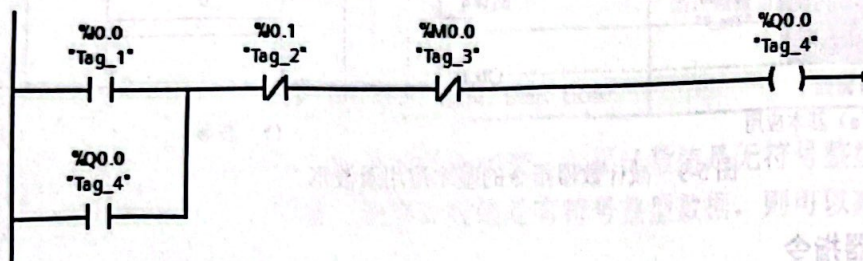
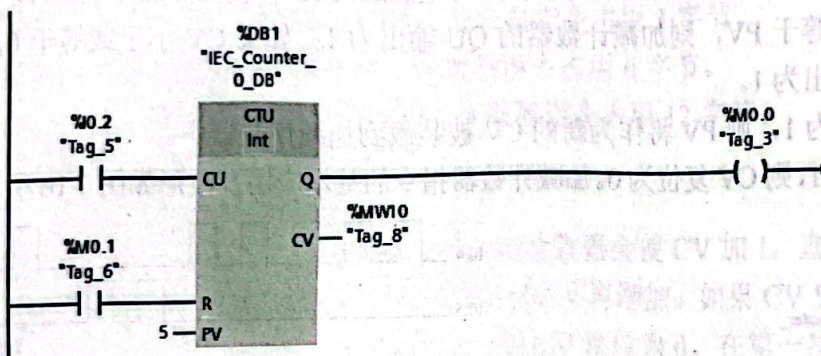


图 5-11 计数器指令应用举例

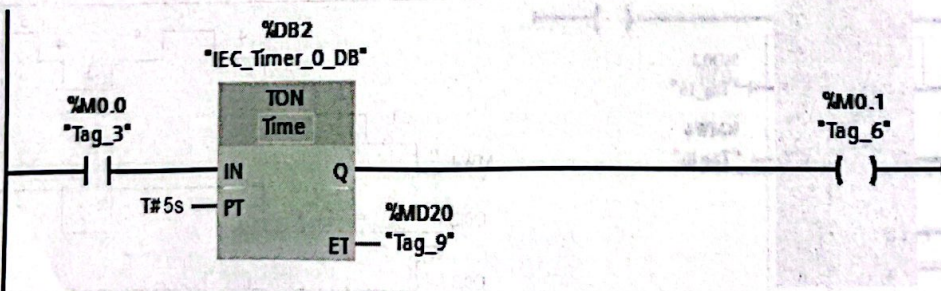
程序段 1：“启—保—停”控制网络，控制传送带的启停



程序段 2：用加计数器指令累计检测到的产品个数



程序段 3：当计数值达到 5 时，接通定时器，延时 2s 后，使加计数器复位



程序段 4：当计数值到 5 且定时时间没到时，机械手有输出信号

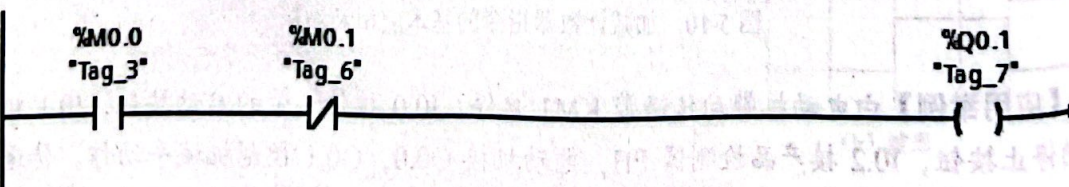
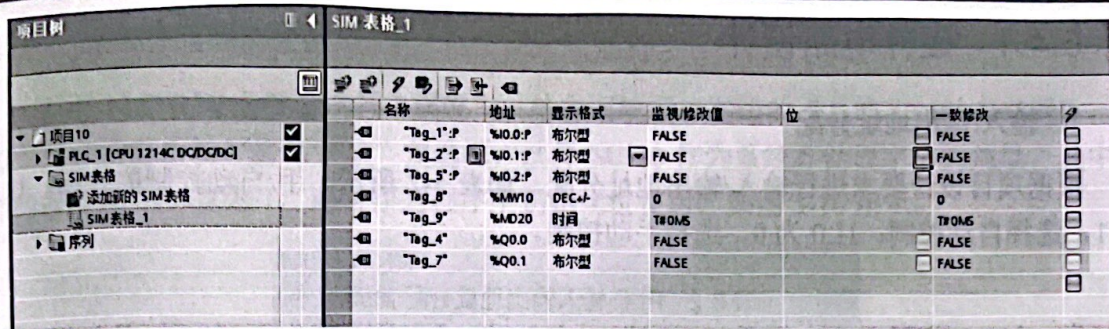


图 5-12 计数器指令应用举例梯形图

可以通过仿真调试验证程序的正确性。将 PLC 站点下载到仿真器中，打开仿真器的目视图，并在 SIM 表格_1 的“地址”栏中输入如图 5-13 所示的绝对地址，其中 MW10 元中为产品个数，MD20 单元中为定时器 DB2 已经消耗的时间。



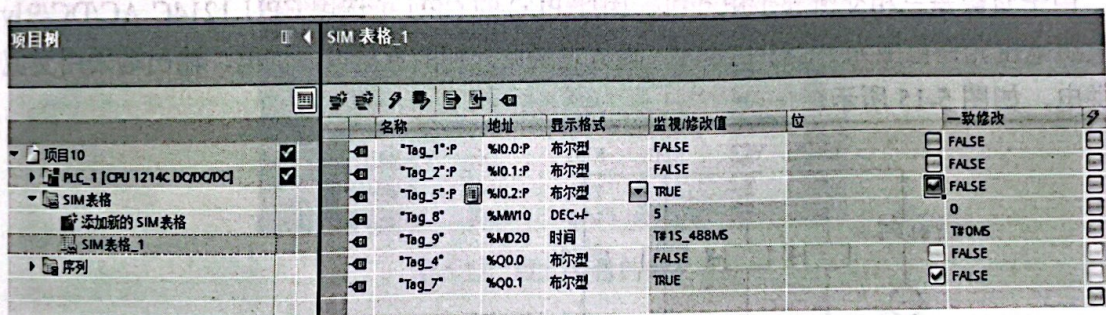
名称	地址	显示格式	监视/修改值	位	一致修改
"Tag_1":P	%I0.0:P	布尔型	FALSE		<input type="checkbox"/> FALSE
"Tag_2":P	%I0.1:P	布尔型	FALSE		<input type="checkbox"/> FALSE
"Tag_5":P	%I0.2:P	布尔型	FALSE		<input type="checkbox"/> FALSE
"Tag_8"	%MW10	DEC+	0		<input type="checkbox"/> 0
"Tag_9"	%MD20	时间	T#0MS		<input type="checkbox"/> T#0MS
"Tag_4"	%Q0.0	布尔型	FALSE		<input type="checkbox"/> FALSE
"Tag_7"	%Q0.1	布尔型	FALSE		<input type="checkbox"/> FALSE

图 5-13 SIM 表格_1 仿真表

(1) 选择 I0.0:P 所在的行, 在其“位”栏中两次单击小方框, 模拟启动按钮的按下和释放操作, Q0.0 的监视/修改值为“TRUE”, 传送带启动。

(2) 单击 I0.2:P“位”栏中的小方框使其中出现“√”, MW10 加 1, 再单击 I0.2:P“位”栏中的小方框使其中的“√”消失, 如此循环, 每执行一次, MW10 自加 1, 当 MW10 的监视/修改值为 5 时, Q0.0 的监视/修改值为“FALSE”, 传送带停止, Q0.1 的监视/修改值为“TRUE”, 机械手动作, 定时器消耗的时间 MD20 增加, 当 MD20 等于 5s 时, MW10 的监视/修改值为 0, MD20 的监视/修改值为 0, Q0.1 的监视/修改值为“FALSE”, 等待再次按下启动按钮, 如图 5-14 所示。

(3) 在计数过程中, 当计数值小于 5 时, 按下停止按钮, Q0.0 的监视/修改值为“FALSE”, 传送带停止, 但计数值不会清零, 再次按下启动按钮, 传送带启动, 计数器在当前计数值的基础上继续计数直到计数值为 5 为止。



名称	地址	显示格式	监视/修改值	位	一致修改
"Tag_1":P	%I0.0:P	布尔型	FALSE		<input type="checkbox"/> FALSE
"Tag_2":P	%I0.1:P	布尔型	FALSE		<input type="checkbox"/> FALSE
"Tag_5":P	%I0.2:P	布尔型	TRUE		<input checked="" type="checkbox"/> FALSE
"Tag_8"	%MW10	DEC+	5		<input type="checkbox"/> 0
"Tag_9"	%MD20	时间	T#1S_488MS		<input type="checkbox"/> T#0MS
"Tag_4"	%Q0.0	布尔型	FALSE		<input type="checkbox"/> FALSE
"Tag_7"	%Q0.1	布尔型	TRUE		<input checked="" type="checkbox"/> FALSE

图 5-14 仿真运行

学习资料

[1] S7-1200: 计数器-跟我学。

<http://www.ad.siemens.com.cn/service/elearning/Course/507.html>。

[2] S7-1200: 计数器-跟我做。

<http://www.ad.siemens.com.cn/service/elearning/Course/508.html>。

5.4 项目实施



运料小车的往返控制实际上就是电动机的正反转控制, 但需要注意的是, 常常需要加热继电器和熔断器以实现过载保护和短路保护。本项目的实施将从输入/输出地址分配、控制电路接线图及软件设计等方面展开。

5.4.1 硬件电路设计

1. 输入/输出地址分配

根据项目设计要求进行输入/输出地址分配，如表 5-2 所示。手/自动控制的选择：I1.0 为 1，选择自动控制；I1.0 为 0，选择手动控制。

表 5-2 PLC 输入输出地址分配表

输入/输出	元 器 件	地 址
输入	启动按钮 SB1	I0.0
	停止按钮 SB2	I0.1
	左侧行程开关 SQ1	I0.2
	右侧行程开关 SQ2	I0.3
	手/自动选择开关 SA	I1.0
	手动前进按钮	I0.6
	手动后退按钮	I0.7
输出	装料电磁阀 YV1	Q0.0
	右行线圈 KM1	Q0.1
	卸料电磁阀 YV2	Q0.2
	左行线圈 KM2	Q0.3

2. 控制电路接线图

由于负载是三相交流异步电动机，因此 PLC 的 CPU 可选用 CPU 1214C AC/DC/Rly。该电源电压允许范围为 AC85~264V，输入端采用外加的直流电源供电，输出端采用交流电源供电，如图 5-15 所示。

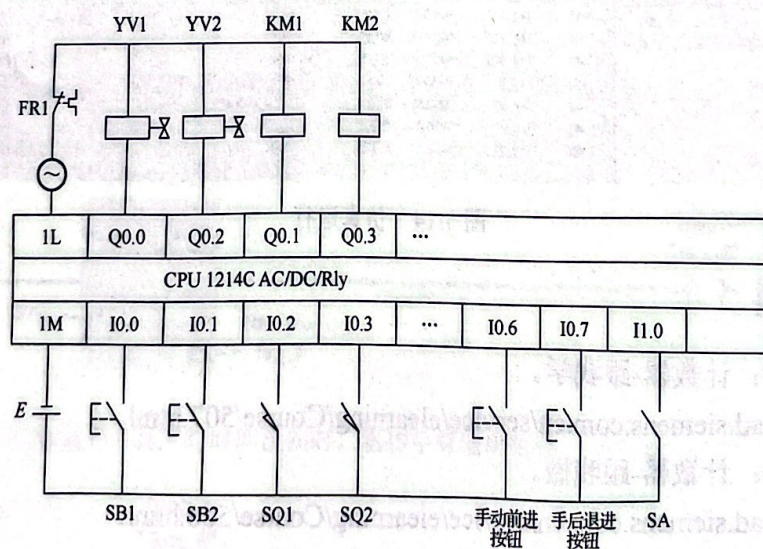


图 5-15 控制电路的接线图

5.4.2 软件设计

软件设计包括 PLC 变量的定义、状态图和顺序功能图的设计及梯形图的设计。在顺序控制程序的设计中，状态图显得尤为重要。一般根据系统的运动规律，只要能够画出正确的状态图，程序设计就十分简单。

1. 系统存储器的设置

打开 PLC 的设备视图选择“属性”，单击巡视窗口左边的“系统和时钟存储器”，勾选“允许使用系统存储器字节”复选框，选择 MB10 作为系统存储器字节的地址，其中的首次循环位 M10.0 为 1，通常作为程序中初始化位使用，如图 5-16 所示。

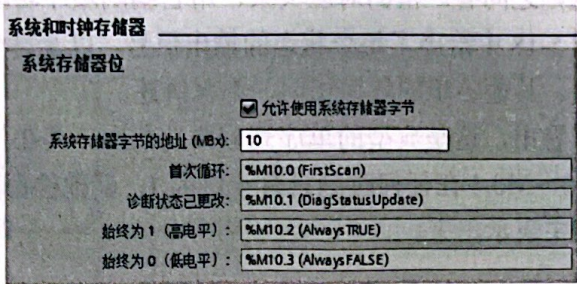


图 5-16 系统存储器的设置

2. PLC 变量的定义

根据设计要求和输入/输出地址分配，为了增加程序的可读性，PLC 变量的定义如图 5-17 所示。

	名称	数据类型	地址	保持	在 H...	可从 ...
1	start	Bool	%I0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	stop	Bool	%I0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	SQ1	Bool	%I0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	SQ2	Bool	%I0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	装料	Bool	%Q0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	右行	Bool	%Q0.1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	卸料	Bool	%Q0.2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	左行	Bool	%Q0.3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	SA选择开关	Bool	%I1.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	手动前进	Bool	%I0.6	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
11	手动后退	Bool	%I0.7	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

图 5-17 PLC 变量的定义

3. 状态图和顺序功能图的设计

采用顺序控制设计法设计程序，首先要画出顺序功能图，顺序功能图中的各步在实现转换时，使前级步的活动结束而使后级步的活动开始，步之间没有重叠。这使系统中大量复杂的联锁关系问题在步的转换中得以解决。对于每个步的程序段，只需要处理极其简单的逻辑关系，编程方法简单易学、规律性强，程序结构清晰、可读性好，调试方便，工作效率高。

系统的工作过程可以分为若干个状态（本项目中共分为 5 个状态，首先是起始状态，接着分别为装料、右行、卸料、左行 4 个状态），当满足某个条件时（时间条件或小车碰到行程开关），系统从当前状态转入下一状态，同时上一状态的动作结束。每个状态对应一个步，可将状态图转换为顺序功能图。顺序功能图可以非常直观、清晰地描述小车自动往返运料的控制过程。

本项目中，5 个状态对应于 5 个步，每个步用一个位存储器来表示，分别为 M0.0~M0.4，如图 5-18 所示。M0.0 为起始步，M0.1 为装料步，M0.2 为右行步，M0.3 为卸料步，M0.4 为左行步。以转换为中心的编程方法的顺序功能图描述如下。

转换的前级步是活动步，实现图 5-18（b）中 I0.0 • I0.2 对应的转换需要同时满足两个

条件, 即该步为活动步 ($M0.0=1$) 和转换条件 ($I0.0 \cdot I0.2=1$), 当这两个条件同时满足时, 就从当前步 $M0.0$ 转换为 $M0.1$ 步, 此时 $M0.0$ 为非活动步, 而 $M0.1$ 为活动步。在顺序功能图中, 可以用 $M0.0$ 和 $I0.0$ 、 $I0.2$ 的常开触点组成的串联关系来表示上述条件。当条件同时满足时, 应将该转换的后续步变成活动步并将该转换的前级步变成非活动步。这种编程方法与转换实现的基本规则之间有严密的对应关系, 用它编制顺序功能图复杂的程序, 更能显示出它的优越性。图 5-18 中给出了每个状态的输出信号, 以及每个状态的转换条件, 给编程提供了极大的方便。其他各步的转换相同, 不再讲述。

由顺序功能图可以看出, 这是典型的单序列顺序功能图, 其在任何时刻只有一个步为活动步, 也就是说 $M0.0 \sim M0.4$ 在任何时刻只有一位为 1, 其他位都为 0。每个步对应的输出也必须在顺序功能图中表示出来。

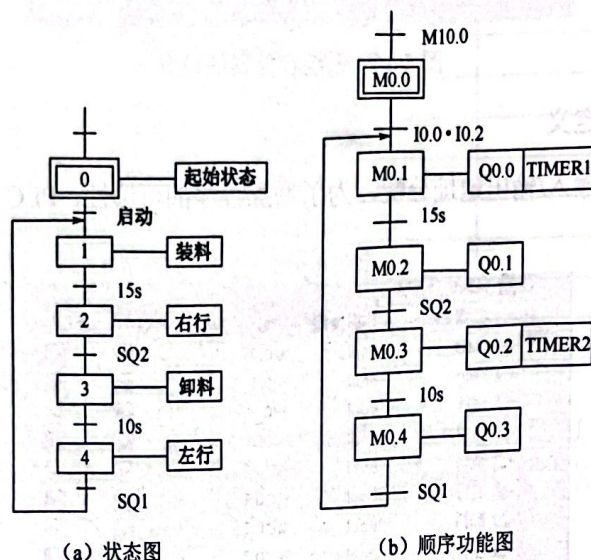


图 5-18 运料小车往返控制的状态图和顺序功能图

4. 梯形图的设计

运料小车往返控制的梯形图如图 5-19 所示。在硬件组态时, 已设置了系统存储器字节的地址为 $MB10$, 首次循环位 $M10.0$ 为 1, 一般用于初始化子程序。整个梯形图采用以转换为中心的程序设计方法, 结构清晰, 程序易读。

程序段 1: 初始化起始步, 并对其他步的标志位和内部标志位清零。在 3 种情况下会初始化起始步: 首次扫描; 选择手动控制; 3 次循环结束。

程序段 2: 在自动控制状态下 ($I1.0=1$), 当前活动步为 $M0.0$, 当满足小车在起始位置 ($I0.2=1$) 条件, 并按下启动按钮 ($I0.0=1$) 时, 由起始步 $M0.0$ 转换为 $M0.1$ 步, 进入装料步, 此时 $M0.0$ 为非活动步, $M0.1$ 为活动步。

程序段 3: 当前活动步为 $M0.1$, 当装料时间到时, 由 $M0.1$ 步转换为 $M0.2$ 步, 此时 $M0.2$ 为活动步, 进入右行状态。

程序段 4: 当前活动步为 $M0.2$, 小车右行到右侧行程开关处, $I0.3$ 为 1, 由 $M0.2$ 步转换为 $M0.3$ 步, 此时 $M0.3$ 为活动步, 进入卸料状态。

程序段 5、6: 卸料 10s, 由 $M0.3$ 步转换为 $M0.4$ 步, 进入左行状态, 小车主行到左侧行程开关处, $I0.2$ 为 1, 回到 $M0.1$ 步, 完成一次循环。

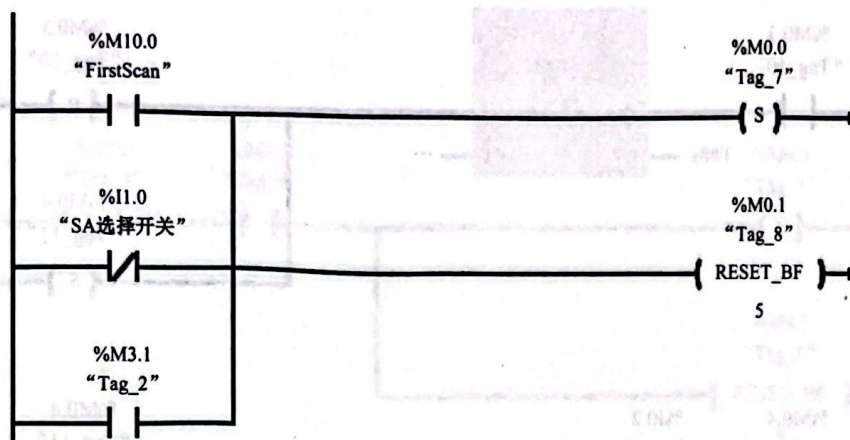
程序段 7: 用计数器指令累计循环次数, 设计要求循环 3 次, 所以 $M0.1$ 的计数值必须

达到 4 次；当循环次数到或选择手动控制或按下停止按钮时，必须对计数器复位。

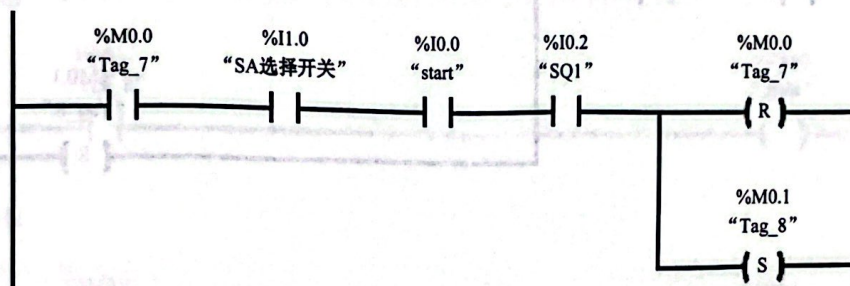
程序段 8、9：按下停止按钮的处理，建立停止运行标志位 M0.5，并回到起始步。

程序段 10~13：输出处理，包括手动输出处理。

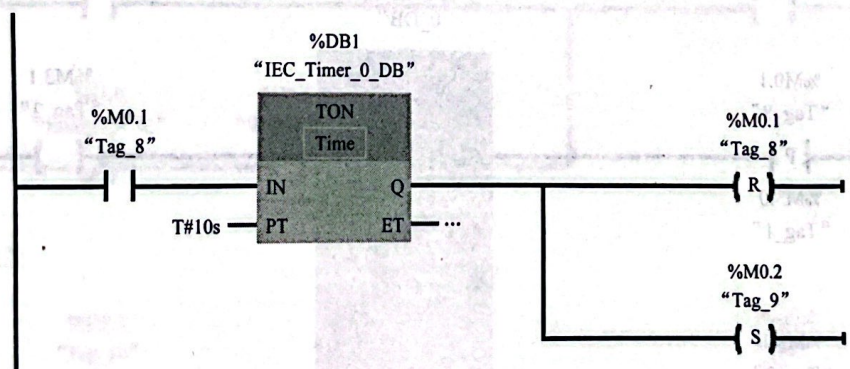
程序段 1：



程序段 2：



程序段 3：



程序段 4：

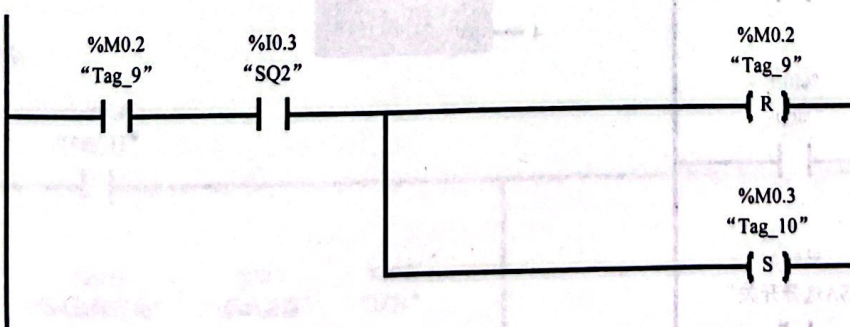
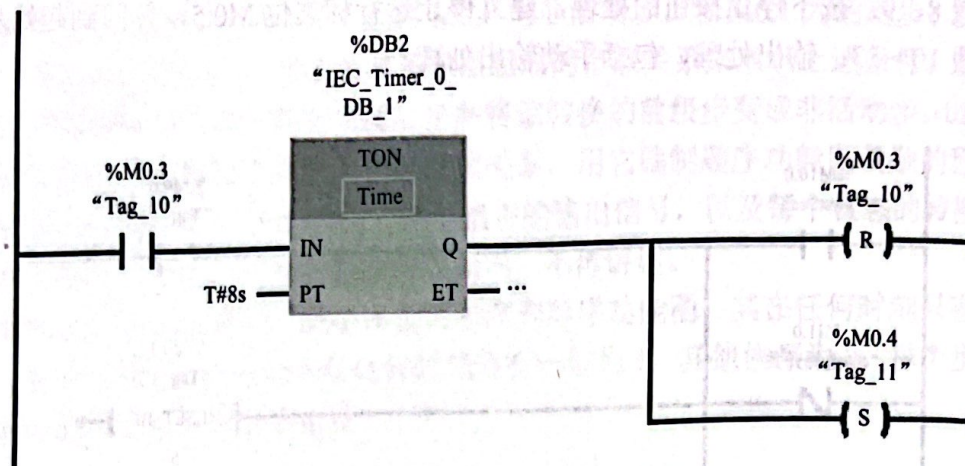
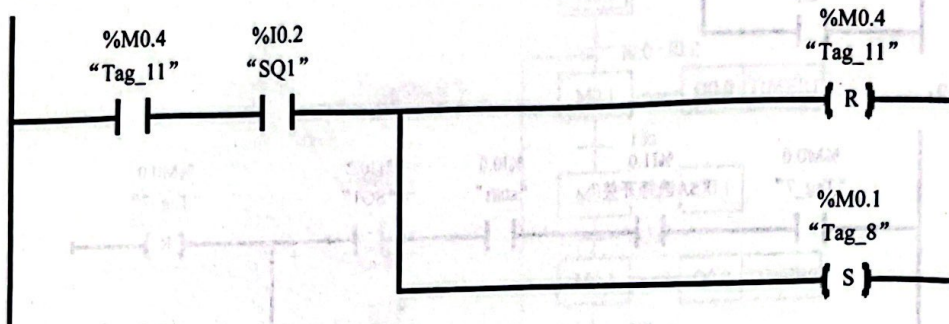


图 5-19 送料小车往返控制的梯形图

程序段 5:



程序段 6:



程序段 7:

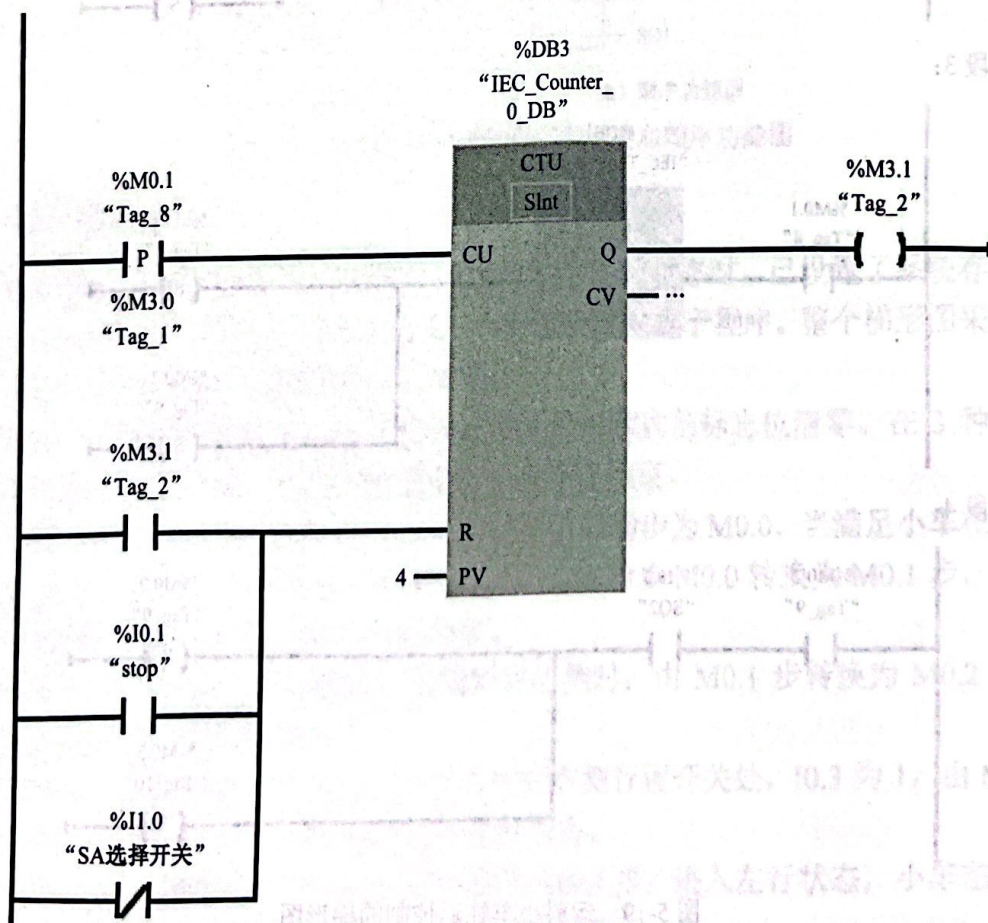
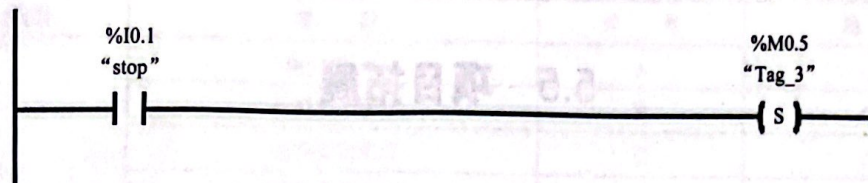
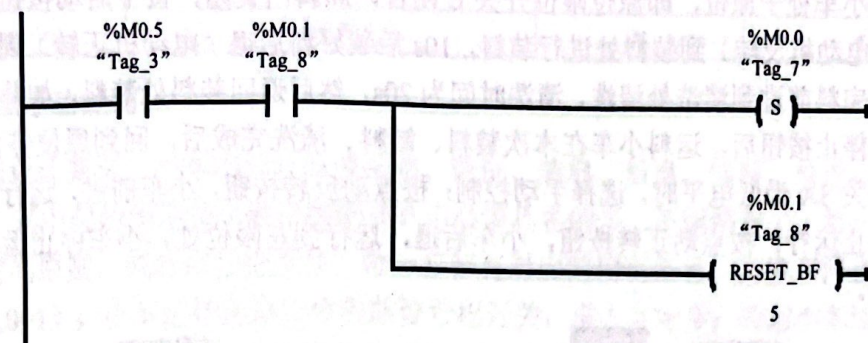


图 5-19 运料小车往返控制的梯形图 (续)

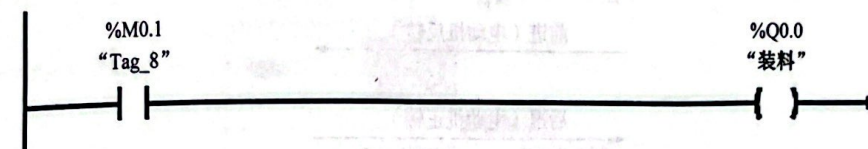
程序段 8:



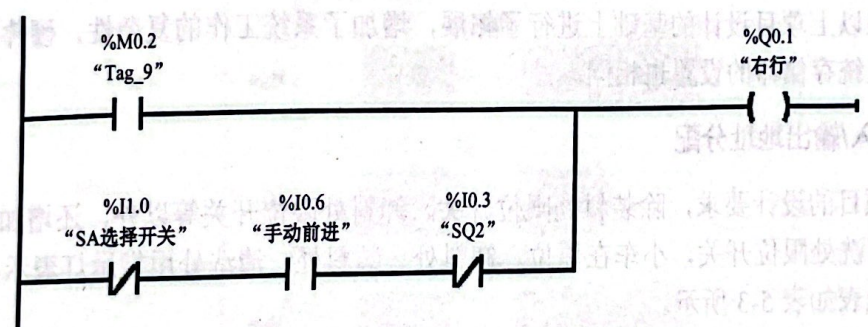
程序段 9:



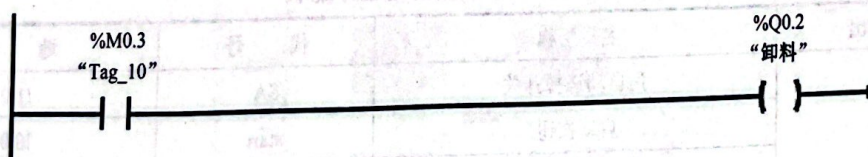
程序段 10:



程序段 11:



程序段 12:



程序段 13:

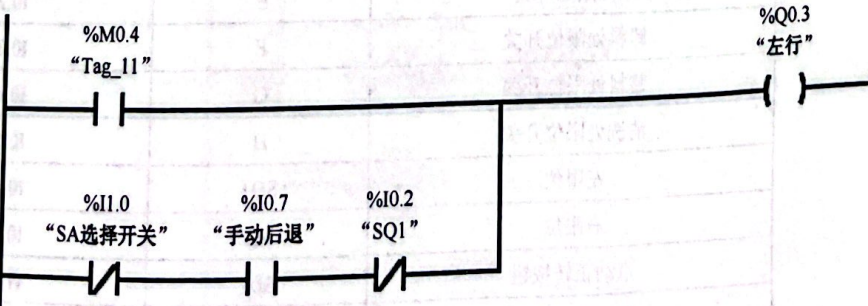


图 5-19 送料小车往返控制的梯形图 (续)