1. 小组讨论与思路设计

**常见电梯调度算法**

根据学习和平时的观察积累，我们知道电梯的调度算法有许多种，可以粗略的分为传统电梯调度算法和实时电梯调度算法两类，各自分类下均有多种细化方案，多种方案各有利弊。

传统电梯调度算法可分为先来先服务算法（FCFS）、最短寻找楼层时间优先算法（SSTF）、扫描算法（SCAN）、LOOK 算法、SAFT 算法等几种。

FCFS算法的核心在于随即服务，其逻辑简单，实现容易。但是在上下楼用户较多（即电梯繁忙）的情况下，电梯的效率会受到严重影响，该种算法只能满足电梯的使用频率不太高时的情况。

SSTF算法注重优化缩短电梯寻找楼层的时间，因此优先响应附近楼层。在电梯不太繁忙的情况下，其效率较高，但是在电梯繁忙的时候，部分楼层可能会出现“久等不应”的情况，不利于部分用户的体验。

SCAN算法的核心在于扫描，电梯循环往返于最高层与最底层，运行过程中响应最近楼层的同向请求。该种算法由于电梯一直在运动，有效的减少了SSTF算法中潜在的“久等不应”问题，但是在电梯不太繁忙时，连续的运动有耗电嫌疑。

LOOK算法在SCAN算法的基础上增加了一个控制，即在电梯运行方向上无用户请求时，电梯直接反向。这一项控制一定程度上提高了效率。

SAFT算法以SSTF算法为基础，进行了拓展和完善，考虑到了乘客进出电梯的时间对整体运行的影响。

实时电梯调度算法可分为最早截止期优先调度算法（EDF）、SCAN-EDF 算法、PI 算法、FD-SCAN 算法等。

实时电梯调度算法以传统电梯调度算法为基础，在相关算法的基础上更强调实时性以及优先级问题。除此之外，随着控制手段以及软硬件的不断发展，也出现了基于专家系统、模糊逻辑、遗产算法以及模糊神经网络等的电梯群控方法。

**小组讨论最终思路**

根据两部电梯运行状态进行分类设计，以目标楼层和当前楼层比较判断是否响应按钮控制信号

（一）电梯静止状态

目前楼层X

响应所有楼层的上行和下行指令，电梯前往按下控制按钮楼层Y

X<Y，电梯上行

X>Y，电梯下行

X=Y，电梯开门

（二）响应电梯内控制面板信号

电梯上行状态，目前楼层X，目标楼层Y，即X<Y

响应处于[X,Y]之间的楼层Z控制按钮信号

X<=Z<=Y，响应Z楼信号，否则不响应

电梯下行状态，目前楼层X，目标楼层Y，即X>Y

响应处于[Y,X]之间的楼层Z控制按钮信号

Y<=Z<=X，响应Z楼信号，否则不响应

（三）响应电梯外控制面板信号

电梯上行状态，目前楼层X，目标楼层Y，即X<Y

响应处于[X,Y]之间的楼层Z向上控制按钮信号

X<=Z<=Y，响应Z楼上行信号，否则不响应

电梯下行状态，目前楼层X，目标楼层Y，即X>Y

响应处于[Y,X]之间的楼层Z向下控制按钮信号

Y<=Z<=X，响应Z楼下行信号，否则不响应

根据小组讨论的设计思路，最终我们组分成ST语言程序编写和梯形图程序编写两种设计思路，第二部分与第三部分分别是对这两种设计的介绍

1. ST语言程序编写

调用变量：

先读取目前电梯所处楼层信号

Display\_Current\_Floor\_Left

Display\_Current\_Floor\_Right

读取目前电梯运行状态信号

LeftCabin. Display\_Current\_Direction

RightCabin. Display\_Current\_Direction

再读取电梯面板内部目标楼层信号

LeftTower.Key\_Floor1 : BOOL; (\*目标楼层按钮\*)

LeftTower.Key\_Floor2 : BOOL;

LeftTower.Key\_Floor3 : BOOL;

LeftTower.Key\_Floor4 : BOOL;

LeftTower.Key\_Floor5 : BOOL;

LeftTower.Key\_Floor6 : BOOL;

LeftTower.Key\_Floor7 : BOOL;

RightTower.Key\_Floor1 : BOOL; (\*目标楼层按钮\*)

RightTower.Key\_Floor2 : BOOL;

RightTower.Key\_Floor3 : BOOL;

RightTower.Key\_Floor4 : BOOL;

RightTower.Key\_Floor5 : BOOL;

RightTower.Key\_Floor6 : BOOL;

RightTower.Key\_Floor7 : BOOL;

Floor1Panel.Key\_Up : BOOL;

Floor1Pane2.Key\_Up : BOOL;

Floor1Pane3.Key\_Up : BOOL;

Floor1Pane4.Key\_Up : BOOL;

Floor1Pane5.Key\_Up : BOOL;

Floor1Pane6.Key\_Up : BOOL;

Floor1Pane7.Key\_Up : BOOL;

电梯上下行状态分类：

(一)电梯静止状态

LeftCabin. Display\_Current\_Direction

RightCabin. Display\_Current\_Direction

目前楼层X

Display\_Current\_Floor\_Left : USINT;(\*左电梯当前楼层显示数码管\*)

Display\_Current\_Floor\_Right : USINT;(\*右电梯当前楼层显示数码管\*)

响应所有楼层的上行和下行指令，电梯前往按下控制按钮楼层Y

Key\_Floor1 : BOOL; (\*目标楼层按钮\*)

Key\_Floor2 : BOOL;

Key\_Floor3 : BOOL;

Key\_Floor4 : BOOL;

Key\_Floor5 : BOOL;

Key\_Floor6 : BOOL;

Key\_Floor7 : BOOL;

X<Y，电梯上行

X>Y，电梯下行

X=Y，电梯开门

（二）响应电梯内控制面板信号

电梯上行状态，目前楼层X，目标楼层Y，即X<Y

响应处于[X,Y]之间的楼层Z控制按钮信号

X<=Z<=Y，响应Z楼信号，否则不响应

电梯下行状态，目前楼层X，目标楼层Y，即X>Y

响应处于[Y,X]之间的楼层Z控制按钮信号

Y<=Z<=X，响应Z楼信号，否则不响应

（三）响应电梯外控制面板信号

电梯上行状态，目前楼层X，目标楼层Y，即X<Y

响应处于[X,Y]之间的楼层Z向上控制按钮信号

X<=Z<=Y，响应Z楼上行信号，否则不响应

电梯下行状态，目前楼层X，目标楼层Y，即X>Y

响应处于[Y,X]之间的楼层Z向下控制按钮信号

Y<=Z<=X，响应Z楼下行信号，否则不响应

设计代码：

Display\_Current\_Floor\_Left:=Floor1Panel.Display\_Current\_Floor\_Left; Display\_Current\_Floor\_Right:=Floor1Panel.Display\_Current\_Floor\_Right;

//读出左右电梯当前楼层  
 IF LeftTower.Signal\_Stop\_CabinMotor=1 AND RightTower.Signal\_Stop\_CabinMotor =1  
 THEN Response:=LeftResponse;   
 ELSIF LeftTower.Signal\_Stop\_CabinMotor= 1 AND RightTower.Signal\_Stop\_CabinMotor <>1  
 THEN Response:=LeftResponse;  
 ELSIF LeftTower.Signal\_Stop\_CabinMotor<>1 AND RightTower.Signal\_Stop\_CabinMotor =1  
 THEN Response:=RightResponse;  
 ELSIF LeftTower.Signal\_Start\_CabinMotor\_P=1 AND RightTower.Signal\_Start\_CabinMotor\_P=1  
 THEN Response:=Response;  
 ELSIF LeftTower.Signal\_Start\_CabinMotor\_P=1 AND RightTower.Signal\_Start\_CabinMotor\_N=1  
 THEN Response:=Response;  
 ELSIF LeftTower.Signal\_Start\_CabinMotor\_N=1 AND RightTower.Signal\_Start\_CabinMotor\_P=1  
 THEN Response:=Response;  
 ELSIF LeftTower.Signal\_Start\_CabinMotor\_N=1 AND RightTower.Signal\_Start\_CabinMotor\_N=1  
 THEN Response:=Response;  
 END\_IF

//对电梯运行状况进行分类，根据两部电梯运行或停止分为六种状态，使用停止电梯响应新来控制信号，而当两个电梯都在运行时则重复执行该段至至少有一部电梯停止再响应新来控制信号。

CASE Response OF   
 LeftResponse :  
 IF Display\_Current\_Floor\_Left = 1   
 THEN   
 IF LeftCabin.Key\_Floor2 = 1 OR Floor2Panel.Key\_Up = 1 OR Floor2Panel.Key\_Down = 1  
 THEN Target\_Floor\_Left := 2;  
 ELSIF LeftCabin.Key\_Floor3 = 1 OR Floor3Panel.Key\_Up = 1 OR Floor3Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 3;  
 ELSIF LeftCabin.Key\_Floor4 = 1 OR Floor4Panel.Key\_Up = 1 OR Floor4Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 4;  
 ELSIF LeftCabin.Key\_Floor5 = 1 OR Floor5Panel.Key\_Up = 1 OR Floor5Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 5;  
 ELSIF LeftCabin.Key\_Floor6 = 1 OR Floor6Panel.Key\_Up = 1 OR Floor6Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 6;  
 ELSIF LeftCabin.Key\_Floor7 = 1 OR Floor7Panel.Key\_Down = 1  
 THEN Target\_Floor\_Left := 7;  
 END\_IF  
 ELSIF Display\_Current\_Floor\_Left =2   
 THEN   
 IF LeftCabin.Key\_Floor1 = 1 OR Floor1Panel.Key\_Up = 1   
 THEN Target\_Floor\_Left := 1;  
 ELSIF LeftCabin.Key\_Floor3 = 1 OR Floor3Panel.Key\_Up = 1 OR Floor3Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 3;  
 ELSIF LeftCabin.Key\_Floor4 = 1 OR Floor4Panel.Key\_Up = 1 OR Floor4Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 4;  
 ELSIF LeftCabin.Key\_Floor5 = 1 OR Floor5Panel.Key\_Up = 1 OR Floor5Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 5;  
 ELSIF LeftCabin.Key\_Floor6 = 1 OR Floor6Panel.Key\_Up = 1 OR Floor6Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 6;  
 ELSIF LeftCabin.Key\_Floor7 = 1 OR Floor7Panel.Key\_Down = 1  
 THEN Target\_Floor\_Left := 7;  
 END\_IF  
 ELSIF Display\_Current\_Floor\_Left =3   
 THEN   
 IF LeftCabin.Key\_Floor1 = 1 OR Floor1Panel.Key\_Up = 1   
 THEN Target\_Floor\_Left := 1;  
 ELSIF LeftCabin.Key\_Floor2 = 1 OR Floor2Panel.Key\_Up = 1 OR Floor2Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 2;  
 ELSIF LeftCabin.Key\_Floor4 = 1 OR Floor4Panel.Key\_Up = 1 OR Floor4Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 4;  
 ELSIF LeftCabin.Key\_Floor5 = 1 OR Floor5Panel.Key\_Up = 1 OR Floor5Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 5;  
 ELSIF LeftCabin.Key\_Floor6 = 1 OR Floor6Panel.Key\_Up = 1 OR Floor6Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 6;  
 ELSIF LeftCabin.Key\_Floor7 = 1 OR Floor7Panel.Key\_Down = 1  
 THEN Target\_Floor\_Left := 7;  
 END\_IF  
 ELSIF Display\_Current\_Floor\_Left =4   
 THEN   
 IF LeftCabin.Key\_Floor1 = 1 OR Floor1Panel.Key\_Up = 1   
 THEN Target\_Floor\_Left := 1;  
 ELSIF LeftCabin.Key\_Floor2 = 1 OR Floor2Panel.Key\_Up = 1 OR Floor2Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 2;  
 ELSIF LeftCabin.Key\_Floor3 = 1 OR Floor3Panel.Key\_Up = 1 OR Floor3Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 3;  
 ELSIF LeftCabin.Key\_Floor5 = 1 OR Floor5Panel.Key\_Up = 1 OR Floor5Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 5;  
 ELSIF LeftCabin.Key\_Floor6 = 1 OR Floor6Panel.Key\_Up = 1 OR Floor6Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 6;  
 ELSIF LeftCabin.Key\_Floor7 = 1 OR Floor7Panel.Key\_Down = 1  
 THEN Target\_Floor\_Left := 7;  
 END\_IF  
 ELSIF Display\_Current\_Floor\_Left =5   
 THEN   
 IF LeftCabin.Key\_Floor1 = 1 OR Floor1Panel.Key\_Up = 1   
 THEN Target\_Floor\_Left := 1;  
 ELSIF LeftCabin.Key\_Floor2 = 1 OR Floor2Panel.Key\_Up = 1 OR Floor2Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 2;  
 ELSIF LeftCabin.Key\_Floor3 = 1 OR Floor3Panel.Key\_Up = 1 OR Floor3Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 3;  
 ELSIF LeftCabin.Key\_Floor4 = 1 OR Floor4Panel.Key\_Up = 1 OR Floor4Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 4;  
 ELSIF LeftCabin.Key\_Floor6 = 1 OR Floor6Panel.Key\_Up = 1 OR Floor6Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 6;  
 ELSIF LeftCabin.Key\_Floor7 = 1 OR Floor7Panel.Key\_Down = 1  
 THEN Target\_Floor\_Left := 7;  
 END\_IF  
 ELSIF Display\_Current\_Floor\_Left =6   
 THEN   
 IF LeftCabin.Key\_Floor1 = 1 OR Floor1Panel.Key\_Up = 1   
 THEN Target\_Floor\_Left := 1;  
 ELSIF LeftCabin.Key\_Floor2 = 1 OR Floor2Panel.Key\_Up = 1 OR Floor2Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 2;  
 ELSIF LeftCabin.Key\_Floor3 = 1 OR Floor3Panel.Key\_Up = 1 OR Floor3Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 3;  
 ELSIF LeftCabin.Key\_Floor4 = 1 OR Floor4Panel.Key\_Up = 1 OR Floor4Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 4;  
 ELSIF LeftCabin.Key\_Floor5 = 1 OR Floor5Panel.Key\_Up = 1 OR Floor5Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 5;  
 ELSIF LeftCabin.Key\_Floor7 = 1 OR Floor7Panel.Key\_Down = 1  
 THEN Target\_Floor\_Left := 7;  
 END\_IF  
 ELSIF Display\_Current\_Floor\_Left =7   
 THEN   
 IF LeftCabin.Key\_Floor1 = 1 OR Floor1Panel.Key\_Up = 1   
 THEN Target\_Floor\_Left := 1;  
 ELSIF LeftCabin.Key\_Floor2 = 1 OR Floor2Panel.Key\_Up = 1 OR Floor2Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 2;  
 ELSIF LeftCabin.Key\_Floor3 = 1 OR Floor3Panel.Key\_Up = 1 OR Floor3Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 3;  
 ELSIF LeftCabin.Key\_Floor4 = 1 OR Floor4Panel.Key\_Up = 1 OR Floor4Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 4;  
 ELSIF LeftCabin.Key\_Floor5 = 1 OR Floor5Panel.Key\_Up = 1 OR Floor5Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 5;  
 ELSIF LeftCabin.Key\_Floor6 = 1 OR Floor6Panel.Key\_Up = 1 OR Floor6Panel.Key\_Down = 1 THEN Target\_Floor\_Left := 6;  
 END\_IF  
 END\_IF;  
 RightResponse :  
 IF Display\_Current\_Floor\_Right = 1   
 THEN   
 IF RightCabin.Key\_Floor2 = 1 OR Floor2Panel.Key\_Up = 1 OR Floor2Panel.Key\_Down = 1  
 THEN Target\_Floor\_Right := 2;  
 ELSIF RightCabin.Key\_Floor3 = 1 OR Floor3Panel.Key\_Up = 1 OR Floor3Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 3;  
 ELSIF RightCabin.Key\_Floor4 = 1 OR Floor4Panel.Key\_Up = 1 OR Floor4Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 4;  
 ELSIF RightCabin.Key\_Floor5 = 1 OR Floor5Panel.Key\_Up = 1 OR Floor5Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 5;  
 ELSIF RightCabin.Key\_Floor6 = 1 OR Floor6Panel.Key\_Up = 1 OR Floor6Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 6;  
 ELSIF RightCabin.Key\_Floor7 = 1 OR Floor7Panel.Key\_Down = 1  
 THEN Target\_Floor\_Right := 7;  
 END\_IF  
 ELSIF Display\_Current\_Floor\_Right =2   
 THEN   
 IF RightCabin.Key\_Floor1 = 1 OR Floor1Panel.Key\_Up = 1   
 THEN Target\_Floor\_Right := 1;  
 ELSIF RightCabin.Key\_Floor3 = 1 OR Floor3Panel.Key\_Up = 1 OR Floor3Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 3;  
 ELSIF RightCabin.Key\_Floor4 = 1 OR Floor4Panel.Key\_Up = 1 OR Floor4Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 4;  
 ELSIF RightCabin.Key\_Floor5 = 1 OR Floor5Panel.Key\_Up = 1 OR Floor5Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 5;  
 ELSIF RightCabin.Key\_Floor6 = 1 OR Floor6Panel.Key\_Up = 1 OR Floor6Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 6;  
 ELSIF RightCabin.Key\_Floor7 = 1 OR Floor7Panel.Key\_Down = 1  
 THEN Target\_Floor\_Right := 7;  
 END\_IF  
 ELSIF Display\_Current\_Floor\_Right =3   
 THEN   
 IF RightCabin.Key\_Floor1 = 1 OR Floor1Panel.Key\_Up = 1   
 THEN Target\_Floor\_Right := 1;  
 ELSIF RightCabin.Key\_Floor2 = 1 OR Floor2Panel.Key\_Up = 1 OR Floor2Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 2;  
 ELSIF RightCabin.Key\_Floor4 = 1 OR Floor4Panel.Key\_Up = 1 OR Floor4Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 4;  
 ELSIF RightCabin.Key\_Floor5 = 1 OR Floor5Panel.Key\_Up = 1 OR Floor5Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 5;  
 ELSIF RightCabin.Key\_Floor6 = 1 OR Floor6Panel.Key\_Up = 1 OR Floor6Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 6;  
 ELSIF RightCabin.Key\_Floor7 = 1 OR Floor7Panel.Key\_Down = 1  
 THEN Target\_Floor\_Right := 7;  
 END\_IF  
 ELSIF Display\_Current\_Floor\_Right =4   
 THEN   
 IF RightCabin.Key\_Floor1 = 1 OR Floor1Panel.Key\_Up = 1   
 THEN Target\_Floor\_Right := 1;  
 ELSIF RightCabin.Key\_Floor2 = 1 OR Floor2Panel.Key\_Up = 1 OR Floor2Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 2;  
 ELSIF RightCabin.Key\_Floor3 = 1 OR Floor3Panel.Key\_Up = 1 OR Floor3Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 3;  
 ELSIF RightCabin.Key\_Floor5 = 1 OR Floor5Panel.Key\_Up = 1 OR Floor5Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 5;  
 ELSIF RightCabin.Key\_Floor6 = 1 OR Floor6Panel.Key\_Up = 1 OR Floor6Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 6;  
 ELSIF RightCabin.Key\_Floor7 = 1 OR Floor7Panel.Key\_Down = 1  
 THEN Target\_Floor\_Right := 7;  
 END\_IF  
 ELSIF Display\_Current\_Floor\_Right =5   
 THEN   
 IF RightCabin.Key\_Floor1 = 1 OR Floor1Panel.Key\_Up = 1   
 THEN Target\_Floor\_Right := 1;  
 ELSIF RightCabin.Key\_Floor2 = 1 OR Floor2Panel.Key\_Up = 1 OR Floor2Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 2;  
 ELSIF RightCabin.Key\_Floor3 = 1 OR Floor3Panel.Key\_Up = 1 OR Floor3Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 3;  
 ELSIF RightCabin.Key\_Floor4 = 1 OR Floor4Panel.Key\_Up = 1 OR Floor4Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 4;  
 ELSIF RightCabin.Key\_Floor6 = 1 OR Floor6Panel.Key\_Up = 1 OR Floor6Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 6;  
 ELSIF RightCabin.Key\_Floor7 = 1 OR Floor7Panel.Key\_Down= 1  
 THEN Target\_Floor\_Right := 7;  
 END\_IF  
 ELSIF Display\_Current\_Floor\_Right =6   
 THEN   
 IF RightCabin.Key\_Floor1 = 1 OR Floor1Panel.Key\_Up = 1   
 THEN Target\_Floor\_Right := 1;  
 ELSIF RightCabin.Key\_Floor2 = 1 OR Floor2Panel.Key\_Up = 1 OR Floor2Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 2;  
 ELSIF RightCabin.Key\_Floor3 = 1 OR Floor3Panel.Key\_Up = 1 OR Floor3Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 3;  
 ELSIF RightCabin.Key\_Floor4 = 1 OR Floor4Panel.Key\_Up = 1 OR Floor4Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 4;  
 ELSIF RightCabin.Key\_Floor5 = 1 OR Floor5Panel.Key\_Up = 1 OR Floor5Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 5;  
 ELSIF RightCabin.Key\_Floor7 = 1 OR Floor7Panel.Key\_Down = 1  
 THEN Target\_Floor\_Right := 7;  
 END\_IF  
 ELSIF Display\_Current\_Floor\_Right =7   
 THEN   
 IF RightCabin.Key\_Floor1 = 1 OR Floor1Panel.Key\_Up = 1   
 THEN Target\_Floor\_Right := 1;  
 ELSIF RightCabin.Key\_Floor2 = 1 OR Floor2Panel.Key\_Up = 1 OR Floor2Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 2;  
 ELSIF RightCabin.Key\_Floor3 = 1 OR Floor3Panel.Key\_Up = 1 OR Floor3Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 3;  
 ELSIF RightCabin.Key\_Floor4 = 1 OR Floor4Panel.Key\_Up = 1 OR Floor4Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 4;  
 ELSIF RightCabin.Key\_Floor5 = 1 OR Floor5Panel.Key\_Up = 1 OR Floor5Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 5;  
 ELSIF RightCabin.Key\_Floor6 = 1 OR Floor6Panel.Key\_Up = 1 OR Floor6Panel.Key\_Down = 1 THEN Target\_Floor\_Right := 6;  
 END\_IF  
 END\_IF  
 END\_CASE

//根据各楼层控制面板上下行按钮与电梯内目标楼层按钮对各楼层进行逐一判别而选取目标楼层，进行扫描识别响应，而两部电梯的响应顺序遵从前一步中停止电梯优先响应控制信号的原则。

CASE STOPCONTORL

IF Target\_Floor\_Left=1 AND LeftTower.Sensor\_Position\_Cabin = 0

THEN LeftTower.Signal\_Start\_DoorMotor\_P=1 AND LeftTower.Signal\_Stop\_CabinMotor=1;

ELSIF Target\_Floor\_Left=2 AND LeftTower.Sensor\_Position\_Cabin = 100

THEN LeftTower.Signal\_Start\_DoorMotor\_P=1 AND LeftTower.Signal\_Stop\_CabinMotor=1;

ELSIF Target\_Floor\_Left=3 AND LeftTower.Sensor\_Position\_Cabin = 200

THEN LeftTower.Signal\_Start\_DoorMotor\_P=1 AND LeftTower.Signal\_Stop\_CabinMotor=1;

ELSIF Target\_Floor\_Left=4 AND LeftTower.Sensor\_Position\_Cabin = 300

THEN LeftTower.Signal\_Start\_DoorMotor\_P=1 AND LeftTower.Signal\_Stop\_CabinMotor=1;

ELSIF Target\_Floor\_Left=5 AND LeftTower.Sensor\_Position\_Cabin = 400

THEN LeftTower.Signal\_Start\_DoorMotor\_P=1 AND LeftTower.Signal\_Stop\_CabinMotor=1;

ELSIF Target\_Floor\_Left=6 AND LeftTower.Sensor\_Position\_Cabin = 500

THEN LeftTower.Signal\_Start\_DoorMotor\_P=1 AND LeftTower.Signal\_Stop\_CabinMotor=1;

ELSIF Target\_Floor\_Left=7 AND LeftTower.Sensor\_Position\_Cabin = 600

THEN LeftTower.Signal\_Start\_DoorMotor\_P=1 AND LeftTower.Signal\_Stop\_CabinMotor=1;

END\_IF

IF Target\_Floor\_Right=1 AND RightTower.Sensor\_Position\_Cabin = 0

THEN RightTower.Signal\_Start\_DoorMotor\_P=1 AND RightTower.Signal\_Stop\_CabinMotor=1;

ELSIF Target\_Floor\_Right=2 AND RightTower.Sensor\_Position\_Cabin = 100

THEN RightTower.Signal\_Start\_DoorMotor\_P=1 AND RightTower.Signal\_Stop\_CabinMotor=1;

ELSIF Target\_Floor\_Right=3 AND RightTower.Sensor\_Position\_Cabin = 200

THEN RightTower.Signal\_Start\_DoorMotor\_P=1 AND RightTower.Signal\_Stop\_CabinMotor=1;

ELSIF Target\_Floor\_Right=4 AND RightTower.Sensor\_Position\_Cabin = 300

THEN RightTower.Signal\_Start\_DoorMotor\_P=1 AND RightTower.Signal\_Stop\_CabinMotor=1;

ELSIF Target\_Floor\_Right=5 AND RightTower.Sensor\_Position\_Cabin = 400

THEN RightTower.Signal\_Start\_DoorMotor\_P=1 AND RightTower.Signal\_Stop\_CabinMotor=1;

ELSIF Target\_Floor\_Right=6 AND RightTower.Sensor\_Position\_Cabin = 500

THEN RightTower.Signal\_Start\_DoorMotor\_P=1 AND RightTower.Signal\_Stop\_CabinMotor=1;

ELSIF Target\_Floor\_Right=7 AND RightTower.Sensor\_Position\_Cabin = 600

THEN RightTower.Signal\_Start\_DoorMotor\_P=1 AND RightTower.Signal\_Stop\_CabinMotor=1;

END\_IF

END\_CASE

//根据目标楼层信号和电梯当前所处楼层传感器信号来判断是否到达楼层及是否停止开门，左右两部电梯独立运行。

//定义变量DownLast与UpLast分别指示DownList与UpList数组的最后一个数字，变量DownList与UpList用于记录所有暂未响应的下行与上行指令

VAR

DownLast : USINT := 0;

UpLast : USINT := 0;

i : USINT := 0;

j : USINT := 0;

k : USINT := 0;

UpList : ARRAY[0..6] OF USINT := [7(0)];

DownList : ARRAY[0..6] OF USINT := [7(0)];

END\_VAR

//Model的循环算法编程

PROGRAM \_CYCLIC

(\*左电梯轿厢门状态机\*)

CASE LeftDoorState OF

DoorIsClosed:

IF LeftTower.Signal\_Start\_DoorMotor\_P THEN

LeftTower.Sensor\_DoorClosed := FALSE;

LeftDoorState := DoorIsOpening;

END\_IF

DoorIsOpened:

IF LeftTower.Signal\_Start\_DoorMotor\_N THEN

LeftTower.Sensor\_DoorOpened := FALSE;

LeftDoorState := DoorIsClosing;

END\_IF

DoorIsClosing:

IF LeftTower.Sensor\_Position\_Door > 0 THEN

LeftTower.Sensor\_Position\_Door := LeftTower.Sensor\_Position\_Door-1;

ELSE

LeftTower.Sensor\_DoorClosed := TRUE;

LeftDoorState := DoorIsClosed;

END\_IF

DoorIsOpening:

IF LeftTower.Sensor\_Position\_Door < 50 THEN

LeftTower.Sensor\_Position\_Door := LeftTower.Sensor\_Position\_Door+1;

ELSE

LeftTower.Sensor\_DoorOpened := TRUE;

LeftDoorState := DoorIsOpened;

END\_IF

END\_CASE

(\*左电梯轿厢运动状态机\*)

CASE LeftCabinState OF

CabinIsStoped:

IF LeftTower.Signal\_Start\_CabinMotor\_P THEN

LeftCabinState := CabinIsGoingUp;

ELSIF LeftTower.Signal\_Start\_CabinMotor\_N THEN

LeftCabinState := CabinIsGoingDown;

END\_IF

LeftCabin.Display\_Current\_Direction := '--';

FOR i := 0 TO 6 DO

FloorPanel[i].Display\_Current\_Direction\_Left := '--';

END\_FOR

CabinIsGoingUp:

IF LeftTower.Sensor\_Position\_Cabin < 600 THEN

LeftTower.Sensor\_Position\_Cabin := LeftTower.Sensor\_Position\_Cabin+1;

END\_IF

IF LeftTower.Signal\_Stop\_CabinMotor THEN

LeftCabinState := CabinIsStoped;

END\_IF

LeftCabin.Display\_Current\_Direction := '/\';

FOR i := 0 TO 6 DO

FloorPanel[i].Display\_Current\_Direction\_Left := '/\';

END\_FOR

CabinIsGoingDown:

IF LeftTower.Sensor\_Position\_Cabin > 0 THEN

LeftTower.Sensor\_Position\_Cabin := LeftTower.Sensor\_Position\_Cabin-1;

END\_IF

IF LeftTower.Signal\_Stop\_CabinMotor THEN

LeftCabinState := CabinIsStoped;

END\_IF

LeftCabin.Display\_Current\_Direction := '\/';

FOR i := 0 TO 6 DO

FloorPanel[i].Display\_Current\_Direction\_Left := '\/';

END\_FOR

END\_CASE

IF LeftTower.Sensor\_Position\_Cabin = 0 THEN

LeftTower.Sensor\_Floor1Approached := TRUE;

ELSE

LeftTower.Sensor\_Floor1Approached := FALSE;

END\_IF

IF LeftTower.Sensor\_Position\_Cabin = 100 THEN

LeftTower.Sensor\_Floor2Approached := TRUE;

ELSE

LeftTower.Sensor\_Floor2Approached := FALSE;

END\_IF

IF LeftTower.Sensor\_Position\_Cabin = 200 THEN

LeftTower.Sensor\_Floor3Approached := TRUE;

ELSE

LeftTower.Sensor\_Floor3Approached := FALSE;

END\_IF

IF LeftTower.Sensor\_Position\_Cabin = 300 THEN

LeftTower.Sensor\_Floor4Approached := TRUE;

ELSE

LeftTower.Sensor\_Floor4Approached := FALSE;

END\_IF

IF LeftTower.Sensor\_Position\_Cabin = 400 THEN

LeftTower.Sensor\_Floor5Approached := TRUE;

ELSE

LeftTower.Sensor\_Floor5Approached := FALSE;

END\_IF

IF LeftTower.Sensor\_Position\_Cabin = 500 THEN

LeftTower.Sensor\_Floor6Approached := TRUE;

ELSE

LeftTower.Sensor\_Floor6Approached := FALSE;

END\_IF

IF LeftTower.Sensor\_Position\_Cabin = 600 THEN

LeftTower.Sensor\_Floor7Approached := TRUE;

ELSE

LeftTower.Sensor\_Floor7Approached := FALSE;

END\_IF

IF LeftTower.Sensor\_Position\_Cabin < 100 THEN

LeftCabin.Display\_Current\_Floor := 1;

ELSIF LeftTower.Sensor\_Position\_Cabin < 200 THEN

LeftCabin.Display\_Current\_Floor := 2;

ELSIF LeftTower.Sensor\_Position\_Cabin < 300 THEN

LeftCabin.Display\_Current\_Floor := 3;

ELSIF LeftTower.Sensor\_Position\_Cabin < 400 THEN

LeftCabin.Display\_Current\_Floor := 4;

ELSIF LeftTower.Sensor\_Position\_Cabin < 500 THEN

LeftCabin.Display\_Current\_Floor := 5;

ELSIF LeftTower.Sensor\_Position\_Cabin < 600 THEN

LeftCabin.Display\_Current\_Floor := 6;

ELSE

LeftCabin.Display\_Current\_Floor := 7;

END\_IF

FOR i := 0 TO 6 DO

FloorPanel[i].Display\_Current\_Floor\_Left := LeftCabin.Display\_Current\_Floor;

END\_FOR

(\*右电梯轿厢门状态机\*)

CASE RightDoorState OF

DoorIsClosed:

IF RightTower.Signal\_Start\_DoorMotor\_P THEN

RightTower.Sensor\_DoorClosed := FALSE;

RightDoorState := DoorIsOpening;

END\_IF

DoorIsOpened:

IF RightTower.Signal\_Start\_DoorMotor\_N THEN

RightTower.Sensor\_DoorOpened := FALSE;

RightDoorState := DoorIsClosing;

END\_IF

DoorIsClosing:

IF RightTower.Sensor\_Position\_Door > 0 THEN

RightTower.Sensor\_Position\_Door := RightTower.Sensor\_Position\_Door-1;

ELSE

RightTower.Sensor\_DoorClosed := TRUE;

RightDoorState := DoorIsClosed;

END\_IF

DoorIsOpening:

IF RightTower.Sensor\_Position\_Door < 50 THEN

RightTower.Sensor\_Position\_Door := RightTower.Sensor\_Position\_Door+1;

ELSE

RightTower.Sensor\_DoorOpened := TRUE;

RightDoorState := DoorIsOpened;

END\_IF

END\_CASE

(\*右电梯轿厢运动状态机\*)

CASE RightCabinState OF

CabinIsStoped:

IF RightTower.Signal\_Start\_CabinMotor\_P THEN

RightCabinState := CabinIsGoingUp;

ELSIF RightTower.Signal\_Start\_CabinMotor\_N THEN

RightCabinState := CabinIsGoingDown;

END\_IF

RightCabin.Display\_Current\_Direction := '--';

FOR i := 0 TO 6 DO

FloorPanel[i].Display\_Current\_Direction\_Right := '--';

END\_FOR

CabinIsGoingUp:

IF RightTower.Sensor\_Position\_Cabin < 600 THEN

RightTower.Sensor\_Position\_Cabin := RightTower.Sensor\_Position\_Cabin+1;

END\_IF

IF RightTower.Signal\_Stop\_CabinMotor THEN

RightCabinState := CabinIsStoped;

END\_IF

RightCabin.Display\_Current\_Direction := '/\';

FOR i := 0 TO 6 DO

FloorPanel[i].Display\_Current\_Direction\_Right := '/\';

END\_FOR

CabinIsGoingDown:

IF RightTower.Sensor\_Position\_Cabin > 0 THEN

RightTower.Sensor\_Position\_Cabin := RightTower.Sensor\_Position\_Cabin-1;

END\_IF

IF RightTower.Signal\_Stop\_CabinMotor THEN

RightCabinState := CabinIsStoped;

END\_IF

RightCabin.Display\_Current\_Direction := '\/';

FOR i := 0 TO 6 DO

FloorPanel[i].Display\_Current\_Direction\_Right := '\/';

END\_FOR

END\_CASE

IF RightTower.Sensor\_Position\_Cabin = 0 THEN

RightTower.Sensor\_Floor1Approached := TRUE;

ELSE

RightTower.Sensor\_Floor1Approached := FALSE;

END\_IF

IF RightTower.Sensor\_Position\_Cabin = 100 THEN

RightTower.Sensor\_Floor2Approached := TRUE;

ELSE

RightTower.Sensor\_Floor2Approached := FALSE;

END\_IF

IF RightTower.Sensor\_Position\_Cabin = 200 THEN

RightTower.Sensor\_Floor3Approached := TRUE;

ELSE

RightTower.Sensor\_Floor3Approached := FALSE;

END\_IF

IF RightTower.Sensor\_Position\_Cabin = 300 THEN

RightTower.Sensor\_Floor4Approached := TRUE;

ELSE

RightTower.Sensor\_Floor4Approached := FALSE;

END\_IF

IF RightTower.Sensor\_Position\_Cabin = 400 THEN

RightTower.Sensor\_Floor5Approached := TRUE;

ELSE

RightTower.Sensor\_Floor5Approached := FALSE;

END\_IF

IF RightTower.Sensor\_Position\_Cabin = 500 THEN

RightTower.Sensor\_Floor6Approached := TRUE;

ELSE

RightTower.Sensor\_Floor6Approached := FALSE;

END\_IF

IF RightTower.Sensor\_Position\_Cabin = 600 THEN

RightTower.Sensor\_Floor7Approached := TRUE;

ELSE

RightTower.Sensor\_Floor7Approached := FALSE;

END\_IF

IF RightTower.Sensor\_Position\_Cabin < 100 THEN

RightCabin.Display\_Current\_Floor := 1;

ELSIF RightTower.Sensor\_Position\_Cabin < 200 THEN

RightCabin.Display\_Current\_Floor := 2;

ELSIF RightTower.Sensor\_Position\_Cabin < 300 THEN

RightCabin.Display\_Current\_Floor := 3;

ELSIF RightTower.Sensor\_Position\_Cabin < 400 THEN

RightCabin.Display\_Current\_Floor := 4;

ELSIF RightTower.Sensor\_Position\_Cabin < 500 THEN

RightCabin.Display\_Current\_Floor := 5;

ELSIF RightTower.Sensor\_Position\_Cabin < 600 THEN

RightCabin.Display\_Current\_Floor := 6;

ELSE

RightCabin.Display\_Current\_Floor := 7;

END\_IF

FOR i := 0 TO 6 DO

FloorPanel[i].Display\_Current\_Floor\_Right := RightCabin.Display\_Current\_Floor;

END\_FOR

END\_PROGRAM

//Controller循环算法编程

PROGRAM \_CYCLIC

LeftTower.Signal\_Start\_DoorMotor\_P := LeftCabin.Key\_Open ;

LeftTower.Signal\_Start\_DoorMotor\_N := LeftCabin.Key\_Close ;

RightTower.Signal\_Start\_DoorMotor\_P := RightCabin.Key\_Open ;

RightTower.Signal\_Start\_DoorMotor\_N := RightCabin.Key\_Close ;

//refresh the UpList

FOR i := 0 TO 6 DO

IF UpList[i] = 0 THEN

UpLast := i;

EXIT;

END\_IF

END\_FOR

FOR i := 0 TO 6 DO

IF FloorPanel[i].Key\_Up THEN

FOR j := 0 TO 6 DO

IF UpList[j] = i+1 THEN

EXIT;

END\_IF

IF j = 6 THEN

UpList[UpLast] := i+1;

UpLast := UpLast+1;

END\_IF

END\_FOR

END\_IF

END\_FOR

//refresh the DownList

FOR i := 0 TO 6 DO

IF DownList[i] = 0 THEN

DownLast := i;

EXIT;

END\_IF

END\_FOR

FOR i := 0 TO 6 DO

IF FloorPanel[i].Key\_Down THEN

FOR j := 0 TO 6 DO

IF DownList[j] = i+1 THEN

EXIT;

END\_IF

IF j = 6 THEN

DownList[DownLast] := i+1;

DownLast := DownLast+1;

END\_IF

END\_FOR

END\_IF

END\_FOR

IF UpList[0] <> 0 THEN

IF LeftCabin.Display\_Current\_Floor < UpList[0] THEN

LeftTower.Signal\_Start\_CabinMotor\_P := TRUE;

LeftTower.Signal\_Start\_CabinMotor\_N := FALSE;

END\_IF

IF LeftCabin.Display\_Current\_Floor > UpList[0] THEN

LeftTower.Signal\_Start\_CabinMotor\_N := TRUE;

LeftTower.Signal\_Start\_CabinMotor\_N := FALSE;

END\_IF

IF LeftCabin.Display\_Current\_Floor = UpList[0] THEN

LeftTower.Signal\_Stop\_CabinMotor := TRUE;

LeftTower.Signal\_Start\_CabinMotor\_P := FALSE;

LeftTower.Signal\_Start\_CabinMotor\_N := FALSE;

LeftTower.Signal\_Start\_DoorMotor\_P := TRUE;

FOR i := 0 TO 6 DO

UpList[i] := UpList[i+1] ;

EXIT;

END\_FOR

END\_IF

END\_IF

// LeftTower.Signal\_Start\_CabinMotor\_P := FloorPanel[0].Key\_Up ;

// RightTower.Signal\_Start\_CabinMotor\_P := FloorPanel[0].Key\_Up ;

//

// LeftTower.Signal\_Start\_CabinMotor\_N := FloorPanel[6].Key\_Down ;

// RightTower.Signal\_Start\_CabinMotor\_N := FloorPanel[6].Key\_Down ;

//

// LeftTower.Signal\_Stop\_CabinMotor := FloorPanel[3].Key\_Up ;

// RightTower.Signal\_Stop\_CabinMotor := FloorPanel[3].Key\_Up ;

IF LeftCabin.Display\_Floor1Selected = UnChecked AND LeftCabin.Key\_Floor1 =1 THEN

LeftCabin.Display\_Floor1Selected := Checked;

END\_IF

IF LeftCabin.Display\_Floor2Selected = UnChecked AND LeftCabin.Key\_Floor2 =1 THEN

LeftCabin.Display\_Floor2Selected := Checked;

END\_IF

IF LeftCabin.Display\_Floor3Selected = UnChecked AND LeftCabin.Key\_Floor3 =1 THEN

LeftCabin.Display\_Floor3Selected := Checked;

END\_IF

IF LeftCabin.Display\_Floor4Selected = UnChecked AND LeftCabin.Key\_Floor4 =1 THEN

LeftCabin.Display\_Floor4Selected := Checked;

END\_IF

IF LeftCabin.Display\_Floor5Selected = UnChecked AND LeftCabin.Key\_Floor5 =1 THEN

LeftCabin.Display\_Floor5Selected := Checked;

END\_IF

IF LeftCabin.Display\_Floor6Selected = UnChecked AND LeftCabin.Key\_Floor6 =1 THEN

LeftCabin.Display\_Floor6Selected := Checked;

END\_IF

IF LeftCabin.Display\_Floor7Selected = UnChecked AND LeftCabin.Key\_Floor7 =1 THEN

LeftCabin.Display\_Floor7Selected := Checked;

END\_IF

IF RightCabin.Display\_Floor1Selected = UnChecked AND RightCabin.Key\_Floor1 =1 THEN

RightCabin.Display\_Floor1Selected := Checked;

END\_IF

IF RightCabin.Display\_Floor2Selected = UnChecked AND RightCabin.Key\_Floor2 =1 THEN

RightCabin.Display\_Floor2Selected := Checked;

END\_IF

IF RightCabin.Display\_Floor3Selected = UnChecked AND RightCabin.Key\_Floor3 =1 THEN

RightCabin.Display\_Floor3Selected := Checked;

END\_IF

IF RightCabin.Display\_Floor4Selected = UnChecked AND RightCabin.Key\_Floor4 =1 THEN

RightCabin.Display\_Floor4Selected := Checked;

END\_IF

IF RightCabin.Display\_Floor5Selected = UnChecked AND RightCabin.Key\_Floor5 =1 THEN

RightCabin.Display\_Floor5Selected := Checked;

END\_IF

IF RightCabin.Display\_Floor6Selected = UnChecked AND RightCabin.Key\_Floor6 =1 THEN

RightCabin.Display\_Floor6Selected := Checked;

END\_IF

IF RightCabin.Display\_Floor7Selected = UnChecked AND RightCabin.Key\_Floor7 =1 THEN

RightCabin.Display\_Floor7Selected := Checked;

END\_IF

END\_PROGRAM

三. 梯形图程序编写

在获得资料和自身经验的基础上，我们讨论出了如下电梯调度方案：

电梯静止，位于x楼

响应所有上、下行请求，记请求楼层数为y

x>y 下行

x<y 上行

x=y 静止在原地，电梯开门

电梯运行，目前位于x楼，目标楼层为y（x≠y），此时z楼发出请求

①指令来自电梯内

x<y 电梯上行

x≤z≤y 响应z楼

z<x 或y<z 暂不响应

x>y 电梯下行

y≤z≤x 响应z楼

z>x 或y>z 暂不响应

②指令来自电梯外

x<y 电梯上行

x≤z≤y且请求上行 响应z楼

x≤z≤y且请求下行 暂不响应

z<x 或y<z 暂不响应

x>y 电梯下行

y≤z≤x且请求下行 响应z楼

y≤z≤x且请求上行 暂不响应

z>x 或y>z 暂不响应

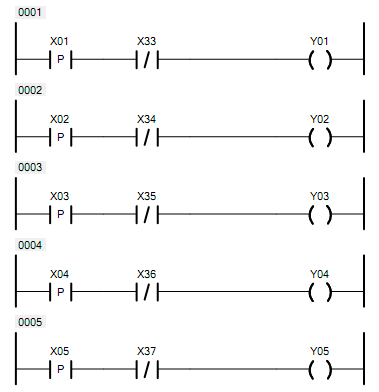
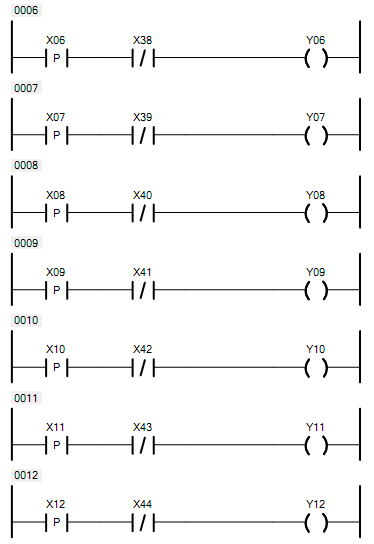
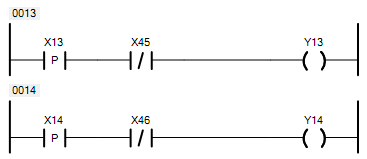
其余响应均以时间顺序为准

简单梯形图设计

首先，根据已有面板定义硬件接口，如下表：

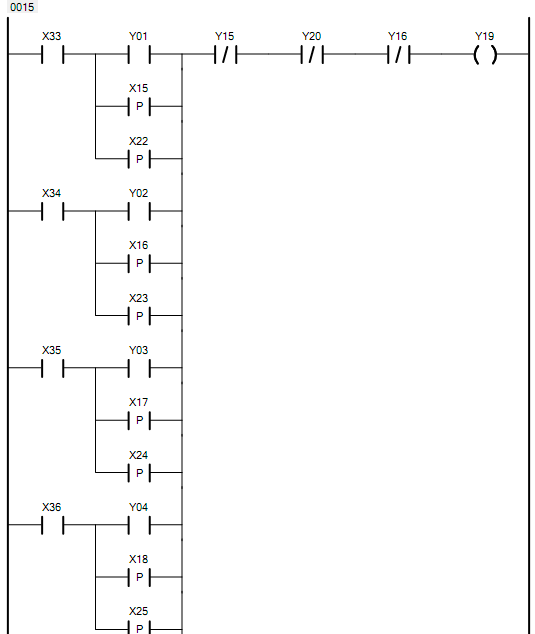
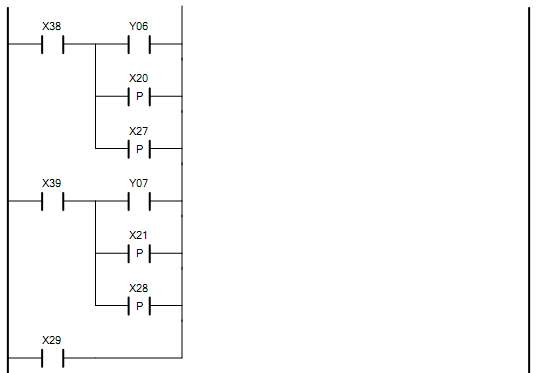
|  |  |  |  |
| --- | --- | --- | --- |
| 电梯与PLC接口对应关系 | | | |
| 电梯输入接口 | PLC输入 | 电梯输出接口 | PLC输出 |
| 左边电梯1楼按键 | X0001 | 左边电梯1楼按键LED | Y0001 |
| 左边电梯2楼按键 | X0002 | 左边电梯2楼按键LED | Y0002 |
| 左边电梯3楼按键 | X0003 | 左边电梯3楼按键LED | Y0003 |
| 左边电梯4楼按键 | X0004 | 左边电梯4楼按键LED | Y0004 |
| 左边电梯5楼按键 | X0005 | 左边电梯5楼按键LED | Y0005 |
| 左边电梯6楼按键 | X0006 | 左边电梯6楼按键LED | Y0006 |
| 左边电梯7楼按键 | X0007 | 左边电梯7楼按键LED | Y0007 |
| 右边电梯1楼按键 | X0008 | 右边电梯1楼按键LED | Y0008 |
| 右边电梯2楼按键 | X0009 | 右边电梯2楼按键LED | Y0009 |
| 右边电梯3楼按键 | X0010 | 右边电梯3楼按键LED | Y0010 |
| 右边电梯4楼按键 | X0011 | 右边电梯4楼按键LED | Y0011 |
| 右边电梯5楼按键 | X0012 | 右边电梯5楼按键LED | Y0012 |
| 右边电梯6楼按键 | X0013 | 右边电梯6楼按键LED | Y0013 |
| 右边电梯7楼按键 | X0014 | 右边电梯7楼按键LED | Y0014 |
| 1楼上按键 | X0015 | 左边电梯电机正转 | Y0015 |
| 2楼上按键 | X0016 | 左边电梯电机反转 | Y0016 |
| 3楼上按键 | X0017 | 右边电梯电机正转 | Y0017 |
| 4楼上按键 | X0018 | 右边电梯电机反转 | Y0018 |
| 5楼上按键 | X0019 | 左边电梯门电机正转 | Y0019 |
| 6楼上按键 | X0020 | 左边电梯门电机反转 | Y0020 |
| 7楼上按键 | X0021 | 右边电梯门电机正转 | Y0021 |
| 1楼下按键 | X0022 | 右边电梯门电机反转 | Y0022 |
| 2楼下按键 | X0023 |  |  |
| 3楼下按键 | X0024 |  |  |
| 4楼下按键 | X0025 |  |  |
| 5楼下按键 | X0026 |  |  |
| 6楼下按键 | X0027 |  |  |
| 7楼下按键 | X0028 |  |  |
| 左边电梯开门 | X0029 |  |  |
| 左边电梯关门 | X0030 |  |  |
| 右边电梯开门 | X0031 |  |  |
| 右边电梯关门 | X0032 |  |  |
| 左边电梯1楼行程开关 | X0033 |  |  |
| 左边电梯2楼行程开关 | X0034 |  |  |
| 左边电梯3楼行程开关 | X0035 |  |  |
| 左边电梯4楼行程开关 | X0036 |  |  |
| 左边电梯5楼行程开关 | X0037 |  |  |
| 左边电梯6楼行程开关 | X0038 |  |  |
| 左边电梯7楼行程开关 | X0039 |  |  |
| 右边电梯1楼行程开关 | X0040 |  |  |
| 右边电梯2楼行程开关 | X0041 |  |  |
| 右边电梯3楼行程开关 | X0042 |  |  |
| 右边电梯4楼行程开关 | X0043 |  |  |
| 右边电梯5楼行程开关 | X0044 |  |  |
| 右边电梯6楼行程开关 | X0045 |  |  |
| 右边电梯7楼行程开关 | X0046 |  |  |
| 左边电梯门开行程开关 | X0047 |  |  |
| 左边电梯门关行程开关 | X0048 |  |  |
| 右边电梯门开行程开关 | X0049 |  |  |
| 右边电梯门关行程开关 | X0050 |  |  |

首先在梯形图中定义按钮上LED显示状态：

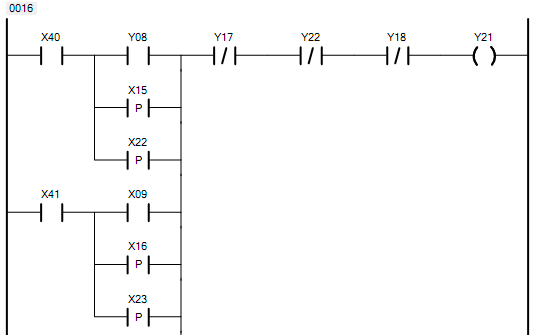
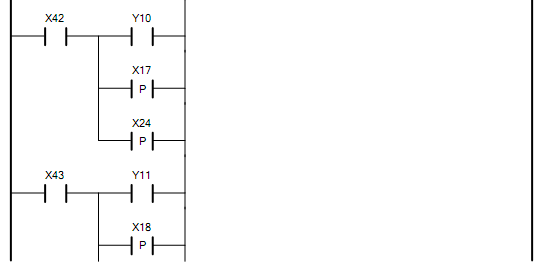
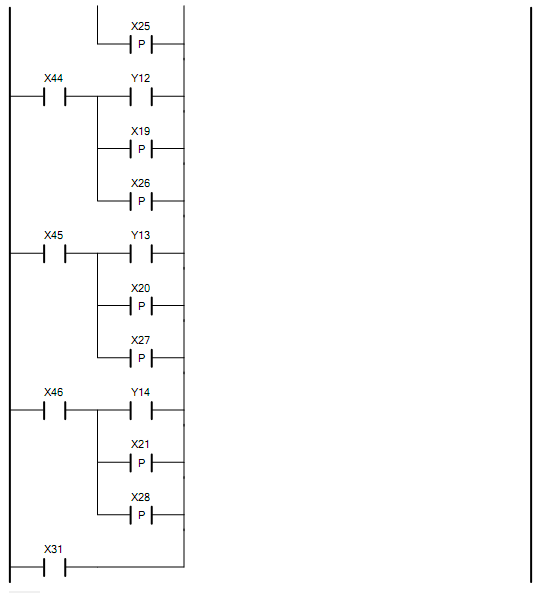
其基本逻辑是：按下电梯内按键，如果此时电梯不处于按键对应楼层，则LED灯亮，一旦电梯运行到对应楼层，LED灯灭。

接下来进行电梯门的控制，以左边电梯门开启指令的触发为例：

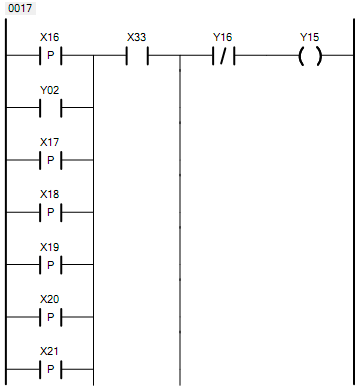
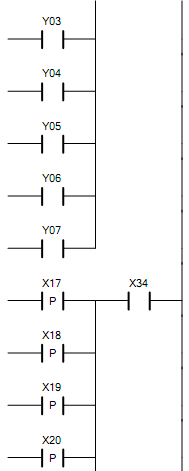
  

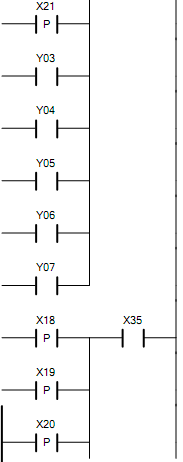
其基础逻辑为：当电梯运行到某一楼层，电梯电机停止，且电梯不处于正在关门状态下，若该楼层为电梯内按键对应目标楼层，或者该楼层有用户发出上行或下行请求时，电梯门控制开启，响应请求。若此时电梯电机停止且电梯门不处于正在关闭的状态，按下电梯内开门按键，也可以开门。

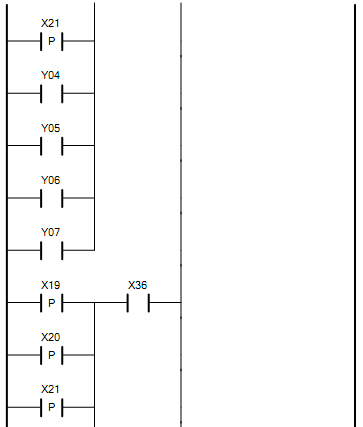
右边电梯开门的情况同理：

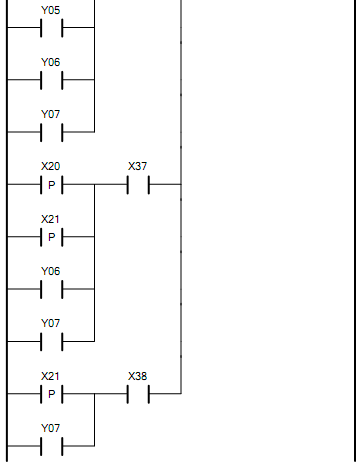
  

对于电梯电机运行情况，梯形图内容较多，仅以电梯停在某层楼，响应其他楼层请求或者电梯内用户按键请求为例：

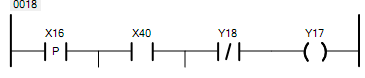
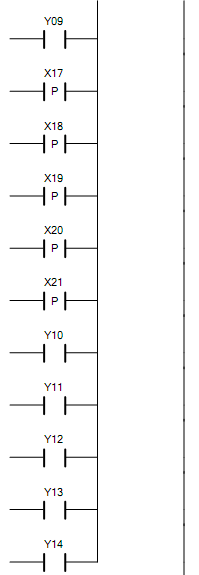


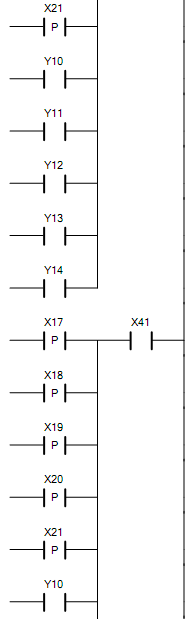


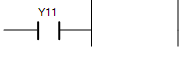


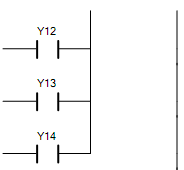
其基本思路是当电梯位于某一楼层，更高楼层有用户发出请求或者电梯内用户发出目标为高楼层的请求时，左边电梯电机正转。

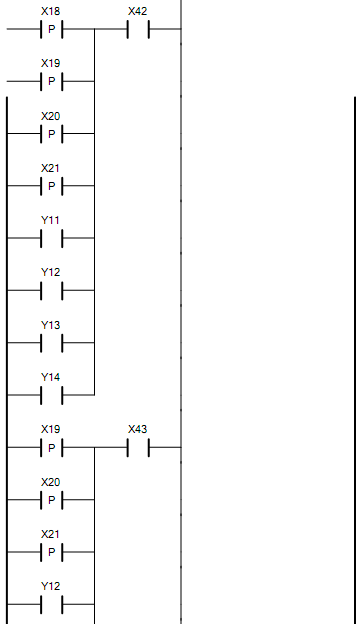
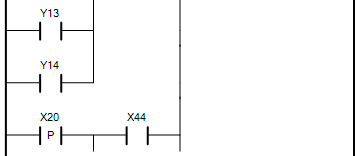
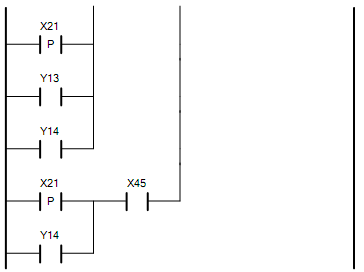
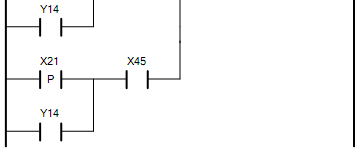
右边同理：

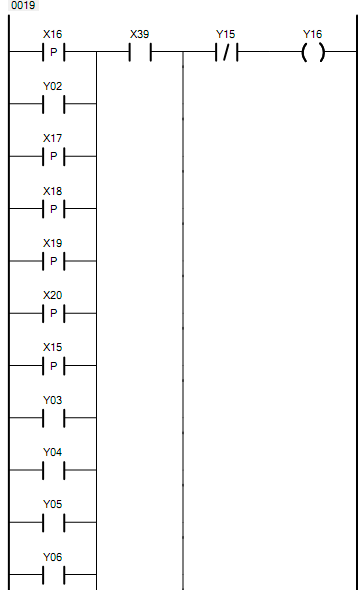
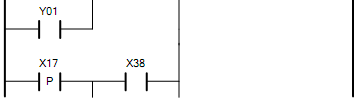
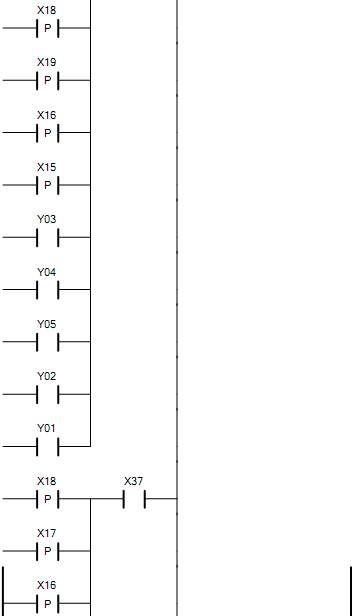
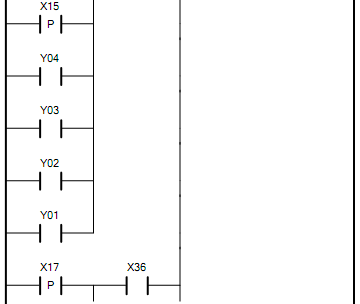
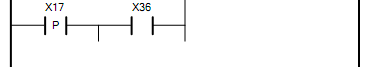
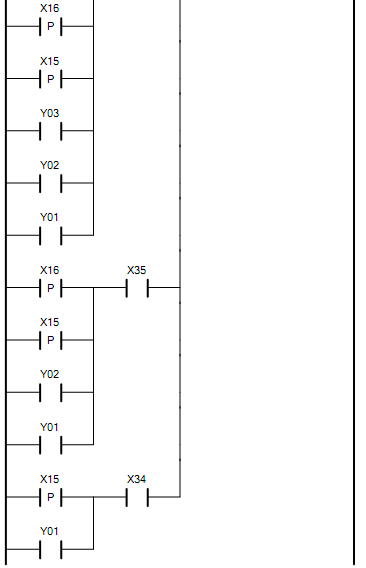




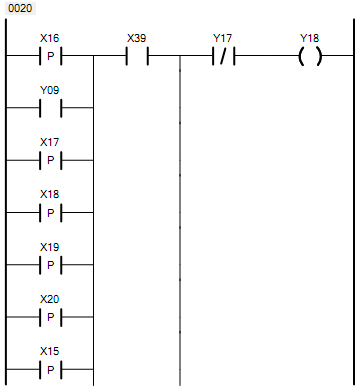
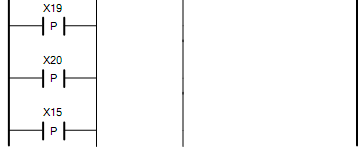
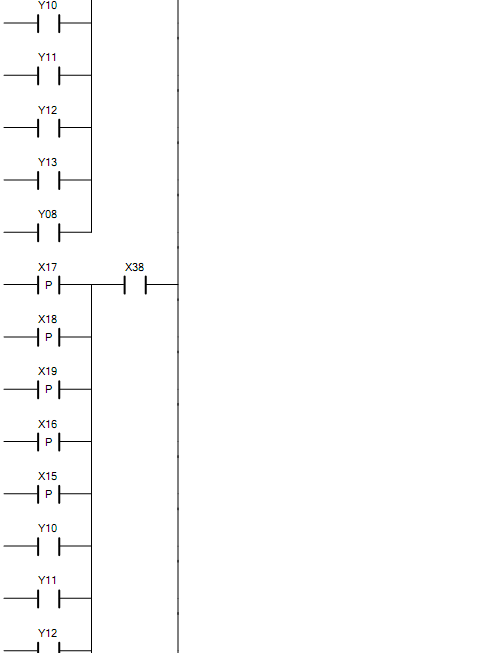
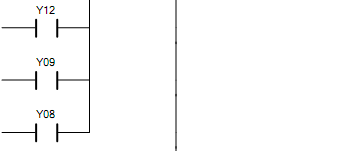
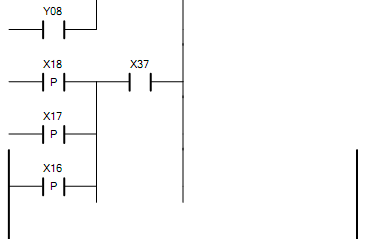
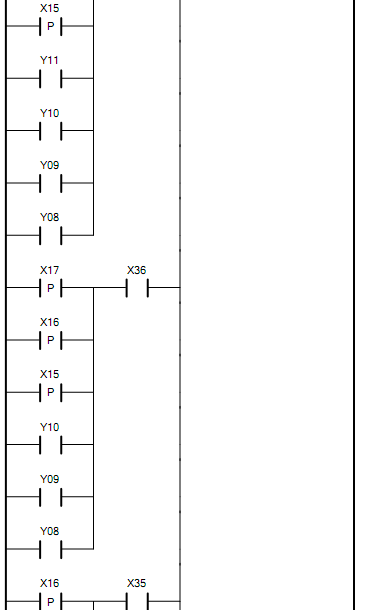
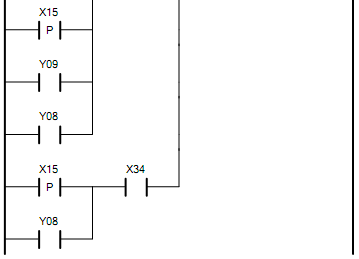


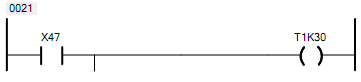
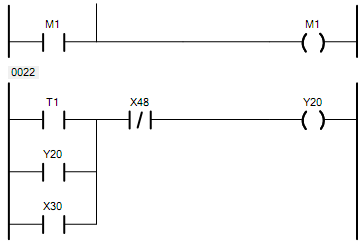
对于电机反转，即电梯下行，考虑的情况与上行同理，即电梯位于某一层楼，响应低楼层请求。

右边电梯同理：

最后考虑电梯门的自动关闭，当电梯门完全开启，行程开关被触发后，借助计时器进行一段延时，然后电梯门电机反转，电梯门自动关闭，电梯门关闭的行程开关被触发后，电梯门电机停止。

右边电梯同理：

