## 三层电梯课程设计报告

## 吴斯祺16040310040

**1.状态机图及其分解描述**：

将三层升降电梯的运行过程分为空闲，向上，向下，开门，关门五个状态。如下图所示。

(a1)

Idle

1

(a3)

(a4)

(a9)

(a2)

(a7)

MovingDown

Movingup

DoorOpen

DoorClosing

(a8)

(B)

(a5)

(B)

(a6)

Idle状态下转换成其他状态或保持不变

a1：由闲置转向向上升的过程，对应的事件是非当前楼层门外calllight被呼叫，或者门内向上的楼层数被点亮。

a2：由闲置转向向下降的过程，对应的事件是非当前楼层门外calllight被呼叫，或者门内向下的楼层数被点亮。

a3：由闲置转向开门，对应的事件是当前楼层门外呼叫或者门内开门按钮点亮。

a4：闲置状态下关门，无动作，灭掉门内该按键。

MovingUp/Down状态下转换成其他状态或者保持不变

a5/a6：由向上升/向下降转向开门的过程，对应的事件有两个：其一是目的地到了，自然开门；其二是若上升/下降的过程中途（在来得及刹车的情况下）楼层同方向被呼叫，则在中途楼层停车，开门上客。

B：在运动过程中出于安全考虑不允许开关门，若有人按了开关门按键，无动作，并且及时灭掉按键。

DoorOpen状态下转换成其他状态或者保持不变

a7：由开门到关门的过程。对应三个事件，其一是门内乘客手动关门；其二是开门动作结束后，门自动关闭；其三是在开门状态下按门内开门按钮电梯无反应，只是灭掉开门按钮的灯。

DoorClosing状态下转换成其他状态或者保持不变

a8：由关门到开门的过程，对应门内开门按键被点亮，电梯开门。

a9：由关门转向闲置状态的过程。对应关门结束后，电梯没有下一步动作，处于空闲的状态。

**2.各状态函数设计过程**

把主要事件记为C1—C5。具体规定如下：

C1: 门内开门按钮(OpenDoorLight)

C2: 门内关门按钮(CloseDoorLight)

C3: 门内楼层按钮(PanelFloorLight)

C4: 门外up呼叫按钮(Call Light)

C5: 门外down呼叫按钮(Call Light)

**静态：**

(a1) 检查C3、C4、C5事件。（根据布尔变量up的值，以及 IdleWhatFloorToGoTo（）返回值来确定电梯下一步应该是怎样的状态。）

属于静态检测。bool up; floor=IdleWhatFloorToGoTo(&up);

关闭本层门外up按钮，SetCallLight()—— 即消费门外up按钮，防止下一周期重复处理此按钮行为。

(a2) 同(C1)，消费门外down按钮。

1层以上，一定时间无动作，自动下降到1楼。AutoTo1Floor();[其它状态，取消此功能，CancelTo1Floor()]

(a3) 检查C1事件, 开门, 消费开门按钮;

上升 (up && E4事件), 开门，消费门外up按钮

下降 (!up && E5事件), 开门，消费门外down按钮

(a4) 检查C2事件,此时门应该是关闭的,因此仅读取关门灯，并关闭关门灯，即消费按键行为，防止下一周期重复处理该按钮的行为。

if(GetCloseDoorLight()) { SetCloseDoorLight(false); return; }

**运动中状态：**MovingUp/MovingDown 🡪 DoorOpen

(a5) 检查C3、C4、C5事件。

动态检测，目标楼层floor=GoingUpToFloor();

if(fabs(GetFloor() - floor) < Lib\_FloorTolerance) 到达目标楼层，停止，开门，消费门外按钮。

(B) 检查C1、C2事件，无动作，消费开/关门按钮。

(a6) 检查C3、C4、C5事件。动态检测，目标楼层=GoingDownToFloor();其它与(C5)类似。

**开门状态:** 电梯门打开 🡪 DoorClosing

(a7) 检查C2事件，转而关门，GetCloseDoorLight(),SetDoor(); 消费关门按钮。

开门结束后，自动进入关门状态。IsDoorOpen();SetDoor();

检查C1事件, 无动作，消费开门按钮。

**关门状态:**

(a8) 检查C1事件，转而开门。GetOpenDoorLight();SetDoor(); 消费关门按钮。进入DoorOpen状态

检查C2事件，无动作，消费关门按钮。

(a9) 关门结束后，进入Idle状态。IsDoorClosed();

**下面是本程序的主要内容：**

#include "stdafx.h"

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include "ElevatorLib.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

三层电梯仿真程序状态函数

学号：16040310040

姓名：吴斯祺

日期：2017年5月

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 主控循环： 状态机，每隔一定时间(如，100ms)被调用一次，采集系统的运行状态

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include "stdafx.h"

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include "ElevatorLib.h"

void StateIdle(int \*state)

{

if (GetCloseDoorLight())

{

SetCloseDoorLight(false); return;

}

if (IsDoorClosed(true))

SetCloseDoorLight(false); return;

}

void StateMovingUp(int \*state)

{

int floor;

bool up;

floor = IdleWhatFloorToGoTo(&up);

if (floor > 0 && up){

SetMotorPower(1);

GetOpenDoorLight();

GetCloseDoorLight();

SetOpenDoorLight(1);

SetCloseDoorLight(0);

}

}

void StateMovingDown(int \*state)

{

bool up;

int floor = IdleWhatFloorToGoTo(&up);

if (floor > 0 && !up){

SetMotorPower(-1);

GetOpenDoorLight();

GetCloseDoorLight();

SetOpenDoorLight(1);

SetCloseDoorLight(0);

}

}

void StateDoorOpen(int \*state)

{

int floor = GoingUpToFloor();

if (GetOpenDoorLight()){

SetDoor(floor, 1);

SetOpenDoorLight(1);

}

if (GetCloseDoorLight()){

SetCloseDoorLight(1);

}

if (IsBeamBroken()){

SetDoor(floor, 1);

}

if (fabs(GetFloor() - floor)< Lib\_FloorNum)

SetMotorPower(0);

SetDoor(floor, 1);

SetCallLight(floor, 0, 0);

SetPanelFloorLight(floor, 0);

GetNearestFloor();

GetNearestFloor();

if (GetOpenDoorLight()){

SetDoor(floor, 1);

SetOpenDoorLight(0);

return;

if (GetCallLight(floor, 0)){

SetDoor(floor, 1);

SetCallLight(floor, 0, 0);

return;

}

}

void StateDoorClosing(int \*state);

{

GetNearestFloor();

if (GetOpenDoorLight()){

SetOpenDoorLight(0);

}

if (IsDoorOpen(floor)){

SetDoor(floor, 0);

}

if (GetOpenDoorLight()){

SetDoor(floor, 1);

SetCloseDoorLight(1);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 状态机，每隔一定时间(如，100ms)被调用一次，采集系统的运行状态

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void main\_control(int \*state)

{

if(IsElevatorRunning()) // 仿真正在运行

{

switch(\*state)

{

case Idle:

// Idle状态，一定时间无动作，自动到一楼

if(GetNearestFloor() !=1 ) {

AutoTo1Floor();

}

StateIdle(state);

break;

case MovingUp:

CancelTo1Floor(); // 其它状态，取消自动到一楼

StateMovingUp(state);

break;

case MovingDown:

CancelTo1Floor();

StateMovingDown(state);

break;

case DoorOpen:

CancelTo1Floor();

StateDoorOpen(state);

break;

case DoorClosing:

CancelTo1Floor();

StateDoorClosing(state);

break;

default:

printf("没有这种状态!!!\n");

}

}

}

吴斯祺 16040310040

2017/5/18