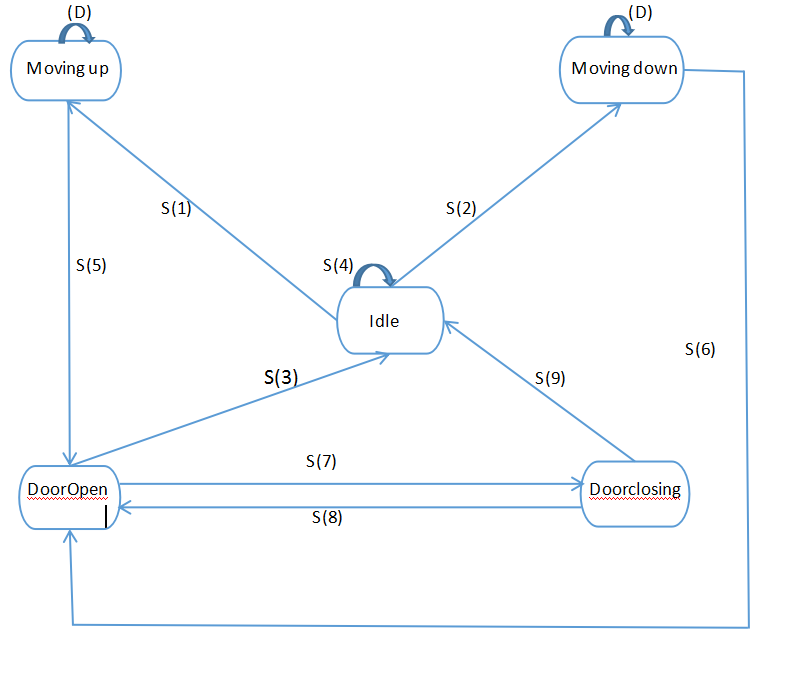
**三层电梯状态机课程设计报告**

**学号：15069130006**

**姓名：张新松**

**一、状态机图及其分解描述**



**二、分解描述：**

如图所示，电梯状态可以分为5大部分，分别为Idle、MovingUp、MovingDown、DoorOpen和DoorClosing。相关事件分别为E1：门内开门按钮；E2：门内关门按钮；E3：门内楼层按钮；E4：门外up呼叫按钮；E5：门外down呼叫按钮。

1. Idle

Idle状态表示电梯停止在某楼层，门是关闭的，处于静止状态，等待相关事件发生，从而转换到下一个状态。从状态机图中可以看出从Idle状态可以通过S1、S2、S3、S4转换成MovingUp、MovingDown、DoorOpen以及保持原状态不变。S1：首先检查E3、E4、E5事件。此为静态检测，bool up；目标楼层floor=IdleWhatFloorToGoTo(&up)；然后关闭本层外的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。S2:大致与S1相同，消费门外down按钮。S3：检查E1事件, 开门, 消费开门按钮;如果上升，则 (up && E4事件), 开门，消费门外up按钮；同理，如果下降，则有 (!up && E5事件), 开门，消费门外down按钮。S4：检查E2事件，此时门应该是关闭的，因此仅读取关门灯，并消费： if(GetCloseDoorLight()) { SetCloseDoorLight(false); return; }。

1. Moving：MovingUp/MovingDown

如状态图所示，无论是Moving的哪种状态，电梯都会从Moving状态转换到开门状态。S5：检查E3、E4、E5事件。此为动态检测，目标楼层floor=GoingUpToFloor();如果到达目标楼层，则if(fabs(GetFloor() - floor) < Lib\_FloorTolerance)，停止，开门并消费门外up按钮；如果到了最高层，则if(floor=Lib\_FloorNum),然后消费门外的down按钮，消费门内楼层按钮。S6：与S5相似，但此时floor=GoingDownToFloor();注意要检查E1、E2事件，如果没有动作，则消费开/关门按钮。

1. DoorOpen

此状态为电梯门打开，它只能转换为DoorClosing状态。S7:检查E2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();

1. DoorClosing

此状态为门正在关闭，由图可知它能转换成两种状态，一是DoorOpen，即S8：检查E1事件，如果有则开门，GetOpenDoorLight();SetDoor(); 然后消费关门按钮；

检测到有障碍物，电梯门要由正在关门转换为开门状态。**IsBeamBroken()；SetDoor()；**

检查E2事件，无动作，消费关门按钮。二是Idle，即S9：关门结束后进入Idle状态。If(IsDoorClosed()){\*state=Idle;return;

以上为三层电梯的5种状态，然后用switch语句写出主控函数，完成此程序。

**三、状态机代码**

#include "stdafx.h"

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include "ElevatorLib.h"

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

\* Idle状态，电梯停止在某楼层，门是关闭的，处于静止状态，等待相关事件的发生，从而转换到下一个状态。

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

void StateIdle(int \*state)

{

int floor;

bool up;

floor = IdleWhatFloorToGoTo(&up);

if (floor > 0 && up) /\* 转换状态，movingup\*/

{

SetMotorPower(1);

\*state = MovingUp;

}

if (floor > 0 && !up) /\*转换状态，movingdown\*/

{

SetMotorPower(-1);

\*state = MovingDown;

}

if (GetOpenDoorLight()) /\*由门内开门按钮转换成dooropen状态\*/

{

SetDoor(GetNearestFloor(), true);

\*state = DoorOpen;

SetOpenDoorLight(false);

}

if (GetCallLight(GetNearestFloor(), true)) /\*门外呼叫按钮转换成dooropen状态\*/

{

SetDoor(GetNearestFloor(), true);

SetCallLight(GetNearestFloor(), true, false);

\*state = DoorOpen;

}

if (GetCallLight(GetNearestFloor(), false))

{

SetDoor(GetNearestFloor(), true);

SetCallLight(GetNearestFloor(), false, false);

\*state = DoorOpen;

}

if (GetCloseDoorLight()) /\*门内关闭按钮转换成doorclose状态\*/

{

SetCloseDoorLight(false);

}

}

void StateMovingUp(int \*state)

{

int floor;

floor = GoingUpToFloor();

if (fabs(GetFloor() - floor) < Lib\_FloorTolerance) /\*检测是否到达目标楼层\*/

{

if (GetNearestFloor() == Lib\_FloorNum) /\*检测是否到达顶层\*/

{

SetMotorPower(0);

SetDoor(floor, true);

SetCallLight(floor, false, false);

SetPanelFloorLight(floor, false);

\*state = DoorOpen;

}

else

{

SetMotorPower(0);

SetDoor(GetNearestFloor(), true);

SetPanelFloorLight(GetNearestFloor(), false);

SetCallLight(GetNearestFloor(), true, false);

SetCallLight(GetNearestFloor(), false, false);

\*state = DoorOpen;

GetNearestFloor(); /\*获得所在层数\*/

}

}

if (GetOpenDoorLight()) /\*消费开关门按钮\*/

{

SetOpenDoorLight(false);

}

if (GetCloseDoorLight())

{

SetCloseDoorLight(false);

}

}

void StateMovingDown(int \*state)

{

int floor;

floor = GoingDownToFloor();

if (fabs(GetFloor() - floor)< Lib\_FloorTolerance) /\*检测是否快到目的楼层\*/

{

if (floor == 1) /\*检测是否到达1层 \*/

{

SetMotorPower(0);

SetDoor(floor, true);

SetCallLight(floor, true, false);

SetPanelFloorLight(floor, false);

\*state = DoorOpen;

}

else

{

SetMotorPower(0);

SetDoor(GetNearestFloor(), true);

SetPanelFloorLight(GetNearestFloor(), false);

SetCallLight(GetNearestFloor(), false, false);

SetCallLight(GetNearestFloor(), true, false);

\*state = DoorOpen;

GetNearestFloor();

}

}

if (GetOpenDoorLight()) /\*消费开关门按钮\*/

{

SetOpenDoorLight(false);

}

if (GetCloseDoorLight())

{

SetCloseDoorLight(false);

}

}

void StateDoorOpen(int \*state)

{

int floor;

floor = GetNearestFloor();

if (GetCloseDoorLight()) /\*正在开门，按下关门灯，转而关门\*/

{

SetDoor(floor, false);

SetCloseDoorLight(false);

\*state = DoorClosing;

}

if (GetOpenDoorLight()) /\*消费开门按钮\*/

{

SetOpenDoorLight(false);

}

if (IsDoorOpen(floor)) /\*开门结束后自动进行关门\*/

{

SetDoor(floor, false);

\*state = DoorClosing;

}

}

void StateDoorClosing(int \*state)

{

int floor;

floor = GetNearestFloor();

if (GetOpenDoorLight()) /\*正在关门，按下开门按钮，转为开门状态\*/

{

SetDoor(GetNearestFloor(), true);

SetOpenDoorLight(false);

\*state = DoorOpen;

}

if (GetCloseDoorLight()) /\*消费关门按钮\*/

{

SetCloseDoorLight(false);

}

if (IsBeamBroken()) /\*检查是否有障碍物，进而转换为开门状态\*/

{

SetDoor(GetNearestFloor(), true);

\*state = DoorOpen;

}

if (IsDoorClosed(floor)) /\*关门结束后切换为闲置状态\*/

{

\*state = Idle;

}

}

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

\* 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");

}

}

}

**四、运行测试**

1.电梯停于1F或2F时，按3F向下呼叫按钮；电梯上升到3F停止，开门/关门。

2.电梯停于2F或3F时，按1F向上呼叫按钮；电梯下降到1F停止，开门/关门。

3.电梯停于1F时，按2F向上呼叫按钮；电梯上升到2F停止，开门/关门。

4.电梯停于3F时，按2F向下呼叫按钮；电梯下降到2F停止，开门/关门。

5.电梯停于1F，2F和3F均有按钮呼叫；电梯先上升到2F，开门/关门，然后上升到3F停止，开门/关门。

6.电梯停于3F，2F和1F均有按钮呼叫；电梯先下降到2F，开门/关门，然后下降到1F停止，开门/关门。

7.电梯上升途中或下降途中，任何反方向按钮呼叫均无效。

8.电梯停于1F，按3F向下呼叫按钮，然后立即按2F向下呼叫按钮；电梯上升到3F停止，开门/关门，然后下降到2F停止，开门/关门。

9.电梯停于2F，按门内楼层按钮3，然后门内楼层按钮1；电梯上手到3F停止，开门/关门，然后下降到1F，开门/关门。

10.电梯停于1F，按门内楼层按钮3，当电梯上升在1F到2F中间以下，按2F向上呼叫按钮；电梯先上升到2F，开门/关门；然后再到3F，开门/关门。

11.电梯停于1F，按门内楼层按钮3，当电梯上升在1F到2F中间以上，按2F向上呼叫按钮；电梯先上升到3F，开门/关门；然后再到2F，开门/关门。

12.电梯停于1F，按2F向下呼叫按钮和向上呼叫按钮以及3F的向下呼叫按钮；电梯上升到2F停止，开门/关门，2F的向上呼叫按钮灯关闭，开门/关门， 2F的向下呼叫按钮灯关闭，然后上升到3F停止，开门/关门，3F的向下呼叫按钮关闭。

13.电梯在2F以上，10s无动作，自动降到1楼。

14.所有停止，开门/关门后，对应楼层的同方向门外呼叫按钮灯（最高楼向下呼叫按钮，最底层向上呼叫按钮）和门内楼层按钮灯关闭。

15.运动状态，开关门按钮失效。正在开门，开门按钮失效；正在关门，关门按钮失效。

16.空闲状态，门是关闭的，因此按关门按钮失效。

17.开关门结束，延时2秒用于乘客上下电梯，然后进入关门状态。

18.开门结束前，按关门按钮，转而进入关门状态；关门结束前，按开门按钮，转而进入开门状态。