三层电梯状态机课程设计报告

1. 状态机图及其分解描述

DoorClosing

DoorOpen

MovingDown

Idle

MovingUp

5

D 1 3

7 8

4

4

9

2

6

D

1. 电梯静止，有人按呼叫灯或电梯内楼层按钮（向上的）
2. 电梯静止，有人按呼叫灯或电梯内楼层按钮（向下的）
3. 电梯静止，有人按开门键或当前楼层的呼叫灯
4. 无动作
5. 到达目标楼层，自动开门

6. 到达目标楼层，自动开门

7．正在开门，按了关门键，转而关门或开门之后自动关门

8 正在关门，按了开门灯转而关门

9. 关门结束，进入Idle状态

D 安全设置无动作

二.用流程图描述课程设计报告

1.Idle状态

梯外呼叫键

梯内楼层键

是否有人按开门键

Idle

是 是

是

MovingUp

MovingDown

是否在当前楼层上方

DoorOpen

是

否

2．MovingUp状态

DoorOpen

是否到达目标楼层

MovingUp

是

3．MovingDown状态

DoorOpen

是否到达目标楼层

MovingDown

4.DoorOpen状态

DoorOpen

DoorClosing

开门是否完成

是

5.DoorClosing状态

DoorOpen

是否有人按开门按钮

DoorClosing

否

是

是否完成开门

是

Idle

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。

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

/\*Idle状态函数\*/

void StateIdle(int \*state)

{

int floor; bool up;

floor = IdleWhatFloorToGoTo(&up);

if (floor > 0)

{

if (up)//Event 检查门内楼层按钮和门外呼叫按钮

{

SetMotorPower(1);//Transition

\*state = MovingUp;

printf("transition:from Idle to MovingUp.\n");

SetCallLight(floor, true, false);//关闭门外呼叫up灯

SetCallLight(floor, false, false);//关闭门外呼叫down灯

}

else

{

SetMotorPower(-1);//Transition

\*state = MovingDown;

printf("transition:from Idle to MovingDown.\n");

SetCallLight(floor, true, false);//关闭门外呼叫up灯

SetCallLight(floor, false, false);//关闭门外呼叫down灯

}

if (GetOpenDoorLight())//Event 检查门内开门按钮

{

SetDoor(floor, true);//Transition

\*state = DoorOpen;

printf("transition:from Idle to DoorOpen.\n");

SetOpenDoorLight(false);//消费开门按钮

}

floor = GetNearestFloor();

if (up&&GetCallLight(floor, true)) //Event 检查门外up按钮

{

SetDoor(floor, true);//Transition

\*state = DoorOpen;

printf("transition:from Idle to DoorOpen.\n");

SetCallLight(floor, true, false);//消费门外up按钮

}

if (!up&&GetCallLight(floor, false))//Event 检查门外down按钮

{

SetDoor(floor, true);//Transition

\*state = DoorOpen;

printf("transition:from Idle to DoorOpen.\n");

SetCallLight(floor, false, false);//消费门外down按钮

}

if (GetCloseDoorLight())//Event

{

SetCloseDoorLight(false);//Transition消费关门按钮

return;

}

}

}

void StateMovingUp(int \*state)

{

int floor;

floor = GoingUpToFloor();//目标楼层

if (fabs(GetFloor() - floor) < Lib\_FloorTolerance)//Event 到达目标楼层

{

SetMotorPower(0);

SetDoor(floor,true);//Transition

\*state = DoorOpen;

printf("transition:from Moving Up to DoorOpen.\n");

SetCallLight(GetNearestFloor(), true, false);//消费门外up按钮

if (GetNearestFloor() == Lib\_FloorNum)

SetCallLight(GetNearestFloor(), false, false);//消费门外down按钮

SetPanelFloorLight(floor, false);//消费梯内楼层按钮

}

/\*安全设置，无动作，消费开/关门按钮\*/

if(GetOpenDoorLight())

SetOpenDoorLight(false);

if(GetCloseDoorLight())

SetCloseDoorLight(false);

}

void StateMovingDown(int \*state)

{

int floor;

floor = GoingUpToFloor();

if (fabs(GetFloor() - floor) < Lib\_FloorTolerance)//Event 到达目标楼层

{

SetMotorPower(0);//停止

SetDoor(floor,true);//Transition

\*state = DoorOpen;

printf("transition:from Moving Down to DoorOpen.\n");

SetCallLight(GetNearestFloor(), false, false);//消费门外down按钮

if (GetNearestFloor()==1)

SetCallLight(GetNearestFloor(), true, false);//消费门外up按钮

SetPanelFloorLight(floor,false);//消费当前楼层按钮

}

/\*安全设置无动作，消费开/关门按钮\*/

if (GetOpenDoorLight())

SetOpenDoorLight(false);

if (GetCloseDoorLight())

SetCloseDoorLight(false);

}

void StateDoorOpen(int \*state)

{

int floor;

floor = GetNearestFloor();

if (GetCloseDoorLight())//Event

{

SetDoor(floor,false);//Transition

\*state = DoorClosing;

printf("transition:from DoorOpen to DoorClosing.\n");

SetCloseDoorLight(false);//消费关门按钮

}

if (IsDoorOpen(floor))//Event

{

SetDoor(floor, false);//Transition开门结束后自动关门

\*state = DoorClosing;

printf("transition:from DoorOpen to DoorClosing.\n");

}

if (GetOpenDoorLight())

{

SetOpenDoorLight(false);//正在开门，按了开门灯，无动作消费开门按钮

}

}

void StateDoorClosing(int \*state)

{

int floor;

floor = GetNearestFloor();

if (GetOpenDoorLight())//Event

{

SetDoor(floor,true);//Transition正在关门，按了开门灯，转而开门

\*state = DoorOpen;

printf("transition:from Doorclosing to Dooropen.\n");

SetOpenDoorLight(false);//消费开门按钮

}

if (GetCloseDoorLight())

{

SetCloseDoorLight(false);//正在关门，按了关门灯，无动作，消费关门按钮

}

if (IsBeamBroken())//Event探测到障碍

{

SetDoor(floor,true);//Transition开门

\*state = DoorOpen;

printf("transition:from Doorclosing to Dooropen.\n");

}

if (IsDoorClosed(floor))//Event关门结束

{

\*state = Idle;

printf("transition:from Doorclosing to Idle.\n");

}

}

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

\* 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【延时功能在库函数中实现，不用在状态函数中实现】，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。