**三层电梯状态机课程设计报告**

1. **状态机图及其分解描述**

Movingup

Idle

S1

S4 D

DoorOpen

S3

S9 S5

DoorClosing

S8

S7

S2

S6

MovingDown

D

**促进状态改变的事件（Events）**

**E1：门内开门按钮(OpenDoorLight)**

**E2：门内关门按钮(CloseDoorLight)**

**E3：门内楼层按钮(PanelFloorLight)**

**E4：门外up按钮(Call light)**

**E5；门外down按钮(Call light)**

**状态，状态改变及其检查事件见下页**

|  |  |
| --- | --- |
| **状态** | **状态改变** |
| **Idle** | **S1，S2，S3，S4** |
| **Movingup** | **S5，D** |
| **DoorClosing** | **S8，S9** |
| **DoorOpen** | **S7** |
| **MovingDown** | **S6，D** |
| **状态改变** | **检查事件** |
| **S1** | **E3，E4，E5** |
| **S2** | **E3，E4，E5** |
| **S3** | **E1，E4/E5** |
| **S4** | **E2** |
| **S5** | **E3，E4，E5** |
| **S6** | **E3，E4，E5** |
| **S7** | **E1，E2** |
| **S8** | **E1，E2** |
| **S9** | **无** |
| **D** | **E1，E2** |

(二)流程

**1： Idle 🡪 MovingUp(S1)**

**MovingDown(S2)**

**DoorOpen(S3)**

**Idle(S4)**

**2：MovingUp 🡪 DoorOpen(S5)**

**Movingup(D)**

**3：MovingDown 🡪 DoorOpen(S6)**

**MovingDown(D)**

**4：DoorOpen🡪 DoorClosing(S7)**

**5：DoorClosing 🡪 DoorOpen(S8)**

**6：DoorClosing 🡪 Idle(S9)**

**（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)

{

SetMotorPower(1); //电梯上升

\*state = MovingUp;

}

else if (floor > 0 && !up)

{

SetMotorPower(-1); //电梯下降

\*state = MovingDown;

}

else if (GetCallLight(floor, true))

{

SetDoor(floor, true); //开门

SetCallLight(floor, true, false); //关闭开门灯

\*state = DoorOpen;

}

else if (GetCallLight(floor, false))

{

SetDoor(floor, true); //开门

SetCallLight(floor, false, false); //关灯

\*state = DoorOpen;

}

else if (GetOpenDoorLight())

{

SetDoor(GetNearestFloor(),true);

SetOpenDoorLight(false); //消费开门按钮

\*state = DoorOpen;

}

else (GetCloseDoorLight)

{

SetCloseDoorLight(false);

return; //消费按键

}

}

void StateMovingUp(int \*state)

{

int floor;

floor = GoingUpToFloor();

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

{

SetMotorPower(0);

SetDoor(GetNearestFloor(), true); //开门

\*state = DoorOpen;

}

else if (GetCallLight(GetNearestFloor(),true)) //消费up键

SetCallLight(GetNearestFloor(),true,false);

else if (Lib\_FloorNum)

{

SetCallLight(GetNearestFloor(), false,false); //消费3层down键

}

SetPanelFloorLight(GetNearestFloor(), false); //消费门内楼层按钮

else if (GetOpenDoorLight())

SetOpenDoorLight(false);

else (GetCloseDoorLight())

SetCloseDoorLight(false); //消费门内开门灯以及关门灯

}

void StateMovingDown(int \*state)

{

int floor;

floor = GoingDownToFloor(); //定义floor为当前层

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

{

SetMotorPower(0); //电梯停止

SetDoor(GetNearestFloor(), true); //开门

\*state = DoorOpen; //转到dooropen状态

}

else if (GetCallLight(GetNearestFloor(), true))

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

else (Lib\_FloorNum)

{

SetCallLight(GetNearestFloor(), false, false);

}

SetPanelFloorLight(GetNearestFloor(), false); //消费门内楼层按钮

}

void StateDoorOpen(int \*state)

{

if (GetCloseDoorLight())

{

SetDoor(GetNearestFloor(), false); //关门

SetCloseDoorLight(false); //消费门内关门灯

\*state = DoorClosing;

}

else if (IsDoorOpen(GetNearestFloor()))

{

SetDoor(GetNearestFloor(),false); //自行关门

\*state = DoorClosing;

}

else (GetOpenDoorLight())

SetOpenDoorLight(false); //消费门内开门灯

}

void StateDoorClosing(int \*state)

{

if (GetOpenDoorLight())

{

SetDoor(GetNearestFloor(),true); //开门

SetOpenDoorLight(false); //消费内门开门等

\*state = DoorOpen;

}

else if (GetCloseDoorLight())

{

SetCloseDoorLight(false); //消费内门关门灯

}

else if (IsBeamBroken())

{

SetDoor(GetNearestFloor(), true); //当红外线检测到有物体未通过，停止关门并开门

\*state = DoorOpen;

}

else (IsDoorClosed(GetNearestFloor()))

\*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");

}

}

}