c语言程序设计

16040310048

赵梅青

三层电梯状态机课程设计报告

1. **状态机图及其分解描述**。

(s4)

(s1) (s2)

(s3) (s9)

(s5)

(D) (D)

(s7)

（s8）

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:门内Up呼叫按钮（CallLight）

E5:门外Down呼叫按钮(CallLight)

|  |  |  |
| --- | --- | --- |
| 状态常数 | 状态函数 | 检查事件，执行动作，状态变迁 |
| Id1e | StateId1e | S1,S2,S3,S4 |
| MovingUp | StateMovingUp | S5,D |
| MovingDown | StateMovingDown | S6,D |
| DoorOpen | StateDoorOpen | S7 |
| DoorClosing | StateDoorClosing | S8,S9 |

二．状态函数。

**Id1e状态:Id1e MovingUp**

S1: if(Id1eWhatFloorToGoTo(&up)>0&&up)

SetMotorPower(1);

**Id1e MovingDown**

S2: if(Id1eWhatFloorToGoTo(&up)>0&&up)

SetMotorPower(-1);

**Id1e DoorOpen**

S3: if(GetPanelFloorLight()||GetCallLight())

SetDoor();

**Id1e Id1e**

S4: if(GetCloseDoorLight())

SetCloseDoorLight(false);

**MovingUp状态：MovingUp DoorOpen**

S5:if(fabs(GetFloor()-floor)<lib\_FloorTolerance)

SetMotorPower(0);

**MovingDown状态MovingDown DoorOpen**

S6: if(fabs(GetFloor()-floor)<lib\_FloorTolerance)

SetMotorPower(0);

**DoorOpen状态：DoorOpen DoorClosing**

S7: if(GetCloseDoorLight())

SetDoor();SetCloseDoorLight();

**DoorClosing状态：DoorClosing DoorOpen**

S8: if(GetOpenDoorLight())

SetDoor();SetOpenDoorLight();

**DoorClosing状态：DoorClosing Id1e**

S9: if(IsDoorClosed)

**MovingDown或MovingUp状态**

D: if(GetOpenDoorLight()||GetCloseDoorLight())

SetOpenDoorLight();SetCloseDoorLight();

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);

\*state = MovingUp;

}

if (floor>0 && !up)

{

SetMotorPower(-1);

\*state = MovingDown;

}

if (GetOpenDoorLight())

{

SetDoor(floor, true);

SetOpenDoorLight(false);

\*state = DoorOpen;

}

if (GetCallLight(floor, true) && floor>0 && up)

{

SetDoor(floor, true);

SetCallLight(floor,true,false);

\*state = MovingUp;

}

if (GetCallLight(floor, false) && floor>0 && !up)

{

SetDoor(floor,true);

SetCallLight(floor, false, false);

\*state = MovingDown;

}

if (GetCloseDoorLight())

{

SetCloseDoorLight(false);

return;

}

}

void StateMovingUp(int \*state)

{

int floor;

floor = GoingUpToFloor();

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

{

SetMotorPower(0);

SetDoor(GetNearestFloor(), true);

SetPanelFloorLight(floor, false);

\*state = DoorOpen;

}

if (GetOpenDoorLight())

{

SetOpenDoorLight(false);

}

if (GetCloseDoorLight())

{

SetCloseDoorLight(false);

}

if (Lib\_FloorNum)

{

SetCallLight(GetNearestFloor(), false,true);

}

}

void StateMovingDown(int \*state)

{

int floor;

floor = GoingUpToFloor();

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

{

SetMotorPower(0);

SetDoor(GetNearestFloor(), true);

SetPanelFloorLight(GetNearestFloor(), false);

\*state = DoorOpen;

}

if (GetOpenDoorLight())

{

SetOpenDoorLight(false);

}

if ( GetCloseDoorLight())

{

SetCloseDoorLight(false);

}

}

void StateDoorOpen(int \*state)

{

if (GetCloseDoorLight())

{

SetDoor(GetNearestFloor(), false);

SetCloseDoorLight(false);

\*state = DoorClosing;

}

if (IsDoorOpen(GetNearestFloor()))

{

SetDoor(GetNearestFloor(), false);

\*state = DoorClosing;

}

if (GetOpenDoorLight())

{

SetOpenDoorLight(false);

}

if (GetCallLight(GetNearestFloor(),true))

{

SetDoor(GetNearestFloor(),true);

SetCallLight(GetNearestFloor(), true, false);

}

if (GetCallLight(GetNearestFloor(), false))

{

SetDoor(GetNearestFloor(), true);

SetCallLight(GetNearestFloor(), false, false);

}

}

void StateDoorClosing(int \*state)

{

if (GetOpenDoorLight())

{

SetDoor(GetNearestFloor(), true);

SetOpenDoorLight(false);

\*state = DoorOpen;

}

if (GetCloseDoorLight())

{

SetCloseDoorLight(false);

}

if (IsBeamBroken())

{

SetDoor(GetNearestFloor(), true);

\*state = DoorOpen;

}

if (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");

}

}

}