C语言课程设计

16040310050姜汶岐

1.按下关门键（if (GetCloseDoorLight())

{

SetDoor(GetNearestFloor(), false);

SetCloseDoorLight(false);

\*state = DoorClosing；} ）

2.时间到了门就自动关闭（else if (IsDoorOpen(GetNearestFloor()))

{

SetDoor(GetNearestFloor(), false);

\*state = DoorClosing;

}）

电梯开门时按下开门键进行消费键的行为（开门到开门）

电梯开门

电梯关门

1.按下开门键（if (GetOpenDoorLight())

{

SetDoor(GetNearestFloor(), false);

SetOpenDoorLight(false);

\*state = DoorOpen; 电梯关门时按下关门键进行消费键 }） 的行为

判断门是否关闭，关闭进入静止状态

2.有障碍物（else if (IsBeamBroken())

else if (IsDoorClosed(GetNearestFloor()))

\*state = Idle;

{

SetDoor(GetNearestFloor(), true);

\*state = DoorOpen;

}）

1.在一楼按下开门键电梯开门if (GetCallLight(1, true)){

SetDoor(1, true);

SetCallLight(1, true,false);

\*state = DoorOpen;}

2.在其他楼层按下开门键或呼叫键也会开门SetDoor(floor, true);

\*state = DoorOpen;

静止状态

静止状态

静止状态

静止状态按下关门键else if (GetCloseDoorLight())

{SetCloseDoorLight(false);

printf("from idle to idle");

return;}

判断是否到达目标楼层，若是则开门if (fabs(GetFloor() - floor) < Lib\_FloorTolerance)

{ SetMotorPower(0);

SetDoor(floor, true);}

静止状态

电梯开门

电梯正在下降

判断是否到达目标楼层如果是则开门

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

{

SetMotorPower(0);

SetDoor(floor, true);

电梯正在上升

\*state = DoorOpen; }

读取门外呼叫按钮（与当前楼层进行判断）进入正在上升的状态

静止状态

if (floor > 0 && up)

{

SetMotorPower(1);

\*state = MovingUp;

printf("from idle to moving up/n");

return;

}

电梯正在上升

读取门外呼叫按钮（与当前楼层进行判断）进入下降状态

else if (floor > 0 && !up)

{

SetMotorPower(-1);

\*state = MovingDown;

printf("from idle to moving down");

return;

}

电梯正在下降

程序所需要的消键方式：  
门内按钮else if (GetOpenDoorLight())

{

SetDoor(GetNearestFloor(), true);

SetOpenDoorLight(false);

\*state = DoorOpen;

printf("from idle to DoorOpen");

return;

}

else if (GetCloseDoorLight())

{

SetCloseDoorLight(false);

printf("from idle to idle");

return;

}

门外按钮

SetCallLight(GetNearestFloor(), up, false);

编程的感想

经过这一次c语言课程设计，我发现了程序员需要一个必备的素质就是仔细，我刚开始编程的时候就没有注意到有些函数是需要给予他们参数的，导致我第一个程序完全运行不了，后面我发现了这个问题于是我开始了我的第二次编程，感觉还好但是程序就是会有bug--电梯门一直开关开关，我找了很久都没有发现自己错在哪里，后来在舍友的帮助下，我发现了自己的问题就是isbeambroken这个函数没有后面跟括号，，，，，，我竟然用了接近一下午的时间，，，，，，，

但是通过这一次的c语言课程设计，我觉得收获颇多，因为我第一次真正的用c语言解决了一个问题虽然当时找错的时候很苦恼，但是当程序做好的一瞬间，看到自己电梯可以正常工作的时候，感觉一切的努力是值得的。

同时我还真正了解到在用c语言编程的时候应该分模块来写函数，然后直接在主函数中调用即可，而且函数模块应该写完一部分就进行debug，这样容易找出自己的错误，而不应该一股脑的将程序输完后在进行改错，这样工作量太大。

程序代码：

#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)

{

bool up;

int floor;

floor=IdleWhatFloorToGoTo(&up);//获取目的楼层

if (GetCallLight(1, true))

{

SetDoor(1, true);

SetCallLight(1, true,false);

\*state = DoorOpen;

}

if (floor > 0 && up)

{

SetMotorPower(1);

\*state = MovingUp;

printf("from idle to moving up/n");

return;

}

else if (floor > 0 && !up)

{

SetMotorPower(-1);

\*state = MovingDown;

printf("from idle to moving down");

return;

}

else if (GetOpenDoorLight())

{

SetDoor(GetNearestFloor(), true);

SetOpenDoorLight(false);

\*state = DoorOpen;

printf("from idle to DoorOpen");

return;

}

else if (GetCallLight(floor, up))

{

SetDoor(floor, true);

SetCallLight(GetNearestFloor(), up, false);

\*state = DoorOpen;

printf("from idle to DoorOpen");

return;

}

else if (GetCloseDoorLight())

{

SetCloseDoorLight(false);

printf("from idle to idle");

return;

}

}

void StateMovingUp(int \*state)

{

int floor;

bool up;

floor=GoingUpToFloor();

up = GetCallLight(floor, true);

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

{

SetMotorPower(0);

SetDoor(floor, true);

if (up==true)

SetCallLight(floor,true,false);

else SetCallLight(floor, false, false);

SetPanelFloorLight(floor, false);

\*state = DoorOpen;

return;

}

if (GetOpenDoorLight||GetCloseDoorLight)

{

SetOpenDoorLight(false);

SetCloseDoorLight(false);

}

}

void StateMovingDown(int \*state)

{

int floor;

bool up;

floor = GoingDownToFloor();

up = GetCallLight(floor, true);

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

{

SetMotorPower(0);

SetDoor(floor, true);

\*state = DoorOpen;

if (up==1)

SetCallLight(floor, true, false);

else

SetCallLight(floor, false, false);

SetPanelFloorLight(floor, false);

return;

}

if (GetOpenDoorLight || GetCloseDoorLight)

{

SetOpenDoorLight(false);

SetCloseDoorLight(false);

}

}

void StateDoorOpen(int \*state)

{

if (GetCloseDoorLight())

{

SetDoor(GetNearestFloor(), false);

SetCloseDoorLight(false);

\*state = DoorClosing;

}

else if (IsDoorOpen(GetNearestFloor()))

{

SetDoor(GetNearestFloor(), false);

\*state = DoorClosing;

}

else if (GetOpenDoorLight())

SetOpenDoorLight(false);

}

void StateDoorClosing(int \*state)

{

if (GetOpenDoorLight())

{

SetDoor(GetNearestFloor(), false);

SetOpenDoorLight(false);

\*state = DoorOpen;

}

else if (GetCloseDoorLight())

{

SetCloseDoorLight(false);

}

else if (IsBeamBroken())

{

SetDoor(GetNearestFloor(), true);

\*state = DoorOpen;

}

else 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");

}

}

}