C语言课程设计报告

1.状态说明：

(S5)

DoorOpen

MovingUp

Idle

(D)

(s3) (s7) (s8)

(s1)

(S4)

DoorClosing

(s9)

(s2)

Moving Down

(s6)

(D)

S1：

floor=IdleWhatFloorToGoTo(&up);

If(floor>0&&up){SetMotorPower(1);}//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,电梯开始上升。

S2：

floor=IdleWhatFloorToGoTo(&up);

If(floor>0&&!up){SetMotorPower(-1);};//1电梯停于2F或3F时，按1F向上呼叫按钮；2电梯停于3F时，按1，2F向下向下呼叫按钮；电梯下降

S3：if(GetOpenDoorLight()){SetDoor(GetNearestFloor(),true);SetOpenDoorLight(false);};

If(GetCallLight()){SetDoor(GetNearestFloor(),true);SetCallLIght(GetNearestFloor(),up,false);};空闲状态，门是关闭的，按下门外呼叫按钮灯（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，电梯开门。

S4：

if(GetCloseDoorLight()){SetCloseDoorlight(false);rturn;};空闲状态，门是关闭的，按关门按钮失效。

S5：

floor=GoingUpToFloor();

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

{SetMotorPower(0);SetDoor(GetNearestFloor(),true);SetCallLight(GetNearestFloor(),true,false);SetCallLight(GetNearestFloor(),false,false);if(floor==3){SetCallLight(GetNearestFloor(),true,false);SetpanelFloorLight(GetNearestFloor(),false);}};电梯上升，到达停止的最近楼层(目标楼层)，电梯停止运行，开门；如果电梯在3楼，此时，门内门外向上按钮灯均失效。

D//(MovingUp):if(GetOpenDoorLight()){SetOpenDoorLight(false);};

If(GetCloseDoorLight()){SetCloseDoorLight(false);};电梯正在上升，开关门按钮均失效。

S6：

floor=GoingDownToFloor();

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

{SetMotorPower(0);SetDoor(GetNearestFloor(),true);SetCallLight(GetNearestFloor(),true,false);SetCallLight(GetNearestFloor(),false,false);if(floor==1){SetCallLight(GetNearestFloor(),false,false);SetpanelFloorLight(GetNearestFloor(),false);}};电梯下降，到达停止的最近楼层(目标楼层)，电梯停止运行，开门；如果电梯在1楼，此时，门内门外向下按钮灯均失效。

D//(MovingDown):if(GetOpenDoorLight()){SetOpenDoorLight(false);};

If(GetCloseDoorLight()){SetCloseDoorLight(false);};电梯正在下降，开关门按钮均失效。

S7：1.if(GetCloseDoorLight()){SetDoor(GetNearestFloor(),false);SetCloseDoorLight(false);};

电梯处于开门状态，按下关门按钮，电梯门关闭。

If(IsDoorOpen(GetNearestFloor())){SetDoor(GetNearestFloor(),false);};电梯处于开门状态，10s无动作，电梯门关闭。

if(GetOpenDoorLight()){SetOpenDoorlight(false);};电梯处于开门状态，开门按钮失效。

S8：

if(GetOpenDoorLight()){SetDoor(GetNearestFloor(),true);SetOpenDoorLight(false);};

电梯处于关门状态，按下开门按钮，电梯开门。

If(GetCloseDoorLight()){SetCloseDoorLight(false);};

电梯处于关门状态，关门按钮失效。

If(IsBeamBroken()){SetDoor(GetNearestFloor(),true);};

红外探测到遮挡物，电梯开门。

S9:if(IsDoorClosed(GetNearestFloor())){\*state=Idle;};电梯门关闭，电梯处于空闲状态。电梯在2F以上，10s无动作，自动降到1楼。

2.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)

{

int floor; bool up;

floor = IdleWhatFloorToGoTo(&up);//Event

if (floor > 0 && up)

{

SetMotorPower(1);//Transition

\*state = MovingUp;

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

}

if (floor > 0 && !up)//Event

{

SetMotorPower(-1);//Transition

\*state = MovingDown;

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

}

if (GetOpenDoorLight())//Event

{

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

SetOpenDoorLight(false);

\*state = DoorOpen;

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

}

if (GetCallLight(GetNearestFloor(),true))//Event

{

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

SetCallLight(GetNearestFloor(),up,false);

\*state = DoorOpen;

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

}

if (GetCloseDoorLight())//Event

{

SetCloseDoorLight(false);//Transition

return;

}

}

void StateMovingUp(int \*state)

{

int floor;

floor = GoingUpToFloor();//Event

if (fabs(GetFloor()-floor)<Lib\_FloorTolerance)//Event， 动态监测，电梯正在上升时，检测将要到达的目标楼层

{

SetMotorPower(0);//Transition

SetDoor(GetNearestFloor(),true);

\*state = DoorOpen;

SetCallLight(GetNearestFloor(),true,false);

SetCallLight(GetNearestFloor(), false, false);

if (floor == 3){ SetCallLight(GetNearestFloor(),true,false); };//Event

SetPanelFloorLight(GetNearestFloor(),false);//Transition

}

if (GetOpenDoorLight()){ SetOpenDoorLight(false); };

if (GetCloseDoorLight()){ SetCloseDoorLight(false); };//无动作，消费按键行为

}

void StateMovingDown(int \*state)

{

int floor;

floor = GoingDownToFloor();

if (fabs(GetFloor() - floor) < Lib\_FloorTolerance)//Event，动态监测，电梯正在下降时，检测将要到达的目标楼层

{

SetMotorPower(0);//Transition

SetDoor(GetNearestFloor(),true);

\*state = DoorOpen;

SetCallLight(GetNearestFloor(),false,false);

SetCallLight(GetNearestFloor(), true, false);

if (floor == 1){ SetCallLight(GetNearestFloor(),false,false); };//Event

SetPanelFloorLight(GetNearestFloor(),false);//Transition

if (GetOpenDoorLight()){ SetOpenDoorLight(false); };

if (GetCloseDoorLight()){ SetCloseDoorLight(false); };//无动作，消费按键行为

}

}

void StateDoorOpen(int \*state)

{

GetNearestFloor();//获取当前层

if (GetCloseDoorLight())//Event

{

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

SetCloseDoorLight(false);

\*state = DoorClosing;

}

if (IsDoorOpen(GetNearestFloor()))//Event

{

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

\*state = DoorClosing;

}

if (GetOpenDoorLight()){ SetOpenDoorLight(false); };

}

void StateDoorClosing(int \*state)

{

if (GetOpenDoorLight())//Event

{

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

SetOpenDoorLight(false);

\*state = DoorOpen;

}

if (GetCloseDoorLight()){ SetCloseDoorLight(false); };

if (IsBeamBroken())//Event，红外探测到遮挡物体

{

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

\*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");

}

}

}

C语言课程设计报告

16040310039

付金双

2017年5月18日