#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;

}

if (floor>0 && !up)

{

SetMotorPower(-1);//Transition

\*state = MovingDown;

}

if (GetOpenDoorLight())

{

SetDoor(GetNearestFloor(),1 ); //Transition

SetOpenDoorLight(0);

\*state = DoorOpen;

}

if (GetCallLight(GetNearestFloor(),1))

{

SetDoor(GetNearestFloor(), 1);//Transition

SetCallLight(GetNearestFloor(), 1, 0);

\*state = DoorOpen;

}

if (GetCallLight(GetNearestFloor(),0))

{

SetDoor(GetNearestFloor(), 1);//Transition

SetCallLight(GetNearestFloor(), 0, 0);

\*state = DoorOpen;

}

if (GetCloseDoorLight())

{

SetCloseDoorLight(0); return;//Transition

}

}

void StateMovingUp(int \*state)

{

int floor;

floor = GoingUpToFloor();//Event

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

{

SetMotorPower(0);//Transition

SetDoor(floor,1);

SetPanelFloorLight(floor, 0);

SetCallLight(floor, 1, 0);

if (floor==Lib\_FloorNum)

{

SetCallLight(floor, 0, 0);//Transition

}

\*state = DoorOpen;

}

if (GetOpenDoorLight() || GetCloseDoorLight())

{

SetOpenDoorLight(0);//Transition

SetCloseDoorLight(0);

}

};

void StateMovingDown(int \*state)

{

int floor;

floor = GoingDownToFloor();//Event

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

{

SetMotorPower(0);//Transition

SetDoor(floor, 1);

SetCallLight(floor, 0, 0);

if (floor == 1)

SetCallLight(floor, 1, 0);//Transition

SetPanelFloorLight(floor, 0);

\*state = DoorOpen;

}

if (GetOpenDoorLight() || GetCloseDoorLight())

{

SetOpenDoorLight(0);//Transition

SetCloseDoorLight(0);

}

}

void StateDoorOpen(int \*state)

{

int floor;

floor = GetNearestFloor();//Event

if (GetCloseDoorLight())

{

SetDoor(floor, 0);//Transition

SetCloseDoorLight(0);

\*state = DoorClosing;

}

if (IsDoorOpen(floor))

{

SetDoor(floor, 0);//Transition

\*state = DoorClosing;

}

if (GetOpenDoorLight())

{

SetOpenDoorLight(0);//Transition

}

}

void StateDoorClosing(int \*state)

{

int floor;

floor = GetNearestFloor();//Event

if (GetOpenDoorLight())

{

SetDoor(floor, 1);//Transition

SetOpenDoorLight(0);

\*state = DoorOpen;

}

if (GetCloseDoorLight())

{

SetCloseDoorLight(0);//Transition

}

if (IsBeamBroken())

{

SetDoor(floor, 1);//Transition

\*state = DoorOpen;

}

}

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

\* 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");

}

}

}

