# C

# 语

# 言

# 课

# 程

# 设

# 计

——侯琳杰（16040310062）

#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); //上升至目标楼层，消费按键，并转移状态

\*state = MovingUp;

}

else if (floor > 0 && !up){ SetMotorPower(-1); //下降至目标楼层，消费按键，并转移状态

\*state = MovingDown;

}

if (GetOpenDoorLight()){ SetDoor(floor,true); //开门，消费按键，并转移状态

SetOpenDoorLight(false);

\*state = DoorOpen;

}

if (GetCloseDoorLight()){ SetCloseDoorLight(false); //消费关门按键

return;

}

}

void StateMovingUp(int \*state)

{

int floor; bool up, s;

floor = GoingUpToFloor();

if (fabs(GetFloor() - floor) < Lib\_FloorTolerance){ SetMotorPower(0);//到达目标楼层，，开门，并转移状态。消费按键

SetDoor(floor,true);

\*state = DoorOpen;

if (fabs(GetFloor() - 3) < Lib\_FloorTolerance)

SetCallLight(floor, false, false);

SetCallLight(floor,true,false);

SetPanelFloorLight(floor,false);

}

else {

GetOpenDoorLight(); SetOpenDoorLight(false); //运行状态的安全设置

GetCloseDoorLight(); SetCloseDoorLight(false);

}

}

void StateMovingDown(int \*state)

{

int floor; bool up;

floor = GoingDownToFloor();

if (fabs(GetFloor() - floor) < Lib\_FloorTolerance){ //到达目标楼层，停止，开门，并转移状态，消费按键

SetMotorPower(0);

SetDoor(floor, true);

\*state = DoorOpen;

if (fabs(GetFloor() - 1) < Lib\_FloorTolerance)

SetCallLight(floor, true, false);

SetCallLight(floor, false, false);

SetPanelFloorLight(floor, false);

GetNearestFloor();

}

else { //安全设置

GetOpenDoorLight(); SetOpenDoorLight(false);

GetCloseDoorLight(); SetCloseDoorLight(false);

}

}

void StateDoorOpen(int \*state)

{

int floor;

floor=GetNearestFloor();

if (GetCloseDoorLight()){ //正在开门，按了关门灯，转而关门，消费关门按键，并转移状态

SetCloseDoorLight(false);

\*state = DoorClosing;

}

else if (IsDoorOpen(floor)){ //自动关门，并转移状态

SetDoor(floor, false);

\*state = DoorClosing;

}

else if (GetOpenDoorLight())

SetOpenDoorLight(false);

}

void StateDoorClosing(int \*state)

{

int floor;

floor = GetNearestFloor();

if (GetCloseDoorLight()){

SetDoor(floor, true); //正在关门，按了开门灯，转而开门，消费按键并转移状态

SetOpenDoorLight(false);

\*state = DoorOpen;

}

else if (GetCloseDoorLight()) SetCloseDoorLight(false);//

else if (IsBeamBroken()){

SetDoor(floor, true);

\*state = DoorOpen;

}

else if (IsDoorClosed(floor)) \*state = Idle; //如果红外探测到遮挡，转而开门，消费按键，并转移状态

}

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

\* 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:

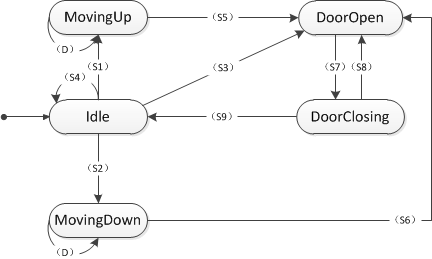
printf("没有这种状态!!!\n");

}

}

}

# ——流程图



if (GetCloseDoorLight()) SetCloseDoorLight(false);

if (GetOpenDoorLight())

SetDoor(floor,true);

SetOpenDoorLight(false);

if (floor > 0 && !up)

SetMotorPower(-1);

if (floor > 0 && up)

SetMotorPower(1);

Movingdown

DoorOpen

Idle

MovingUp

if (fabs(GetFloor() - 3) < Lib\_FloorTolerance)

SetCallLight(floor, false, false);

SetCallLight(floor,true,false)

SetPanelFloorLight(floor,false);

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

SetMotorPower(0);

SetDoor(floor,true)

if (GetCloseDoorLight())

SetCloseDoorLight(false) if (IsDoorOpen(floor))

SetDoor(floor, false);

;

if (GetOpenDoorLight())

SetOpenDoorLight(false)

DoorOpen

DoorClosing

GetOpenDoorLight(); SetOpenDoorLight(false);

GetCloseDoorLight(); SetCloseDoorLight(false);

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

SetMotorPower(0);

SetDoor(floor, true);

MovingDown

DoorOpen

DoorOpen

MovingUp

if (IsDoorClosed(floor))

if (IsBeamBroken())

SetDoor(floor, true);

Idle

if (GetCloseDoorLight())

SetDoor(floor, true);

SetOpenDoorLight(false);

DoorClosing

DoorOpen