## 三层电梯状态机设计报告

**16040310021 刘矫龙**

DoorOpen

MovingUp

S5

S4

S1

S3

S7

S8

DoorClosing

Idle

S9

S2

MovingDown

S6

Idle

S1：由idle到moving up。事件：非当前楼层门外calllight被呼叫，或者门内向上的楼层数被点亮。

代码：

int floor; bool up;

floor = IdleWhatFloorToGoTo(&up);to

if (floor > 0 && up)

{

SetMotorPower(1);

\*state = MovingUp;

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

}

S2：由idle到moving down。事件：非当前楼层门外calllight被呼叫，或者门内向下的楼层数被点亮。

代码：

if (floor>0&&!up)

{

SetMotorPower(-1);

\*state = MovingUp;

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

}

if (GetOpenDoorLight())

{

SetDoor(floor, true);

\*state = DoorOpen;

SetOpenDoorLight(false);

SetCallLight(GetNearestFloor(), true, false);

SetCallLight(GetNearestFloor(), false, false);

}

if (GetCloseDoorLight())

{

SetCloseDoorLight(false);

return;

}

S3：由idle到door open。事件：当前楼层门外呼叫或者门内开门按钮点亮。

代码：

if (GetOpenDoorLight())

{

SetDoor(floor, true);

\*state = DoorOpen;

SetOpenDoorLight(false);

SetCallLight(GetNearestFloor(), true, false);

SetCallLight(GetNearestFloor(), false, false);

}

S4：idle到idle。关门，无动作，灭掉门内该按键。

代码：

if (GetCloseDoorLight())

{

SetCloseDoorLight(false);

return;

}

MovingUp/Down

S5/S6：moving up/moving down到door open。事件：目的楼层到了，自然开门；若up/down的过程中途楼层同方向被呼叫，则在中途楼层停车，开门上客。

代码（Up）:

int floor;

floor = GoingUpToFloor();

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

{

SetMotorPower(0);

SetDoor(floor, true);

\*state = DoorOpen;

printf("Translation:from MovingUp to DoorOpen.\n");

SetPanelFloorLight(GetNearestFloor(), false);

if (floor = Lib\_Num)

{

SetCallLight(false);

}

}

if (GetOpenDoorLight() || GetCloseDoorLight())

{

SetOpenDoorLight(false);

SetCloseDoorLight(false);

}

代码（Down）:

int floor;

floor = GoingDownToFloor();

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

{

SetMotorPower(0);

SetDoor(floor, true);

\*state = DoorOpen;

printf("Translation:from MovingDown to DoorOpen.\n");

SetPanelFloorLight(GetNearestFloor(), false);

if (floor ==1)

{

SetCallLight(false);

}

D：moving up/moving down中不允许开关门。事件：按了开关门按键，无动作，同时灭掉按键。

代码：

if (GetOpenDoorLight() || GetCloseDoorLight())

{

SetOpenDoorLight(false);

SetCloseDoorLight(false);

}

DoorOpen

S7：door open到doorclosing。事件：门内乘客手动关门；开门动作结束后，门自动关闭；在开门状态下按门内开门按钮电梯无反应，只是灭掉开门按钮的灯。

代码：

if (GetCloseDoorLight())

{

SetDoor(GetNearestFloor(), false);

SetCloseDoorLight(false);

\*state = DoorClosing;

printf("Translation:from DoorOpen to DoorClosing.\n");

}

if (IsDoorOpen(GetNearestFloor()))

{

SetDoor(GetNearestFloor, false);

\*state = DoorClosing;

printf("Translation:from DoorOpen to DoorClosing.\n");

}

if (GetCloseDoorLight())

{

SetCloseDoorLight(false);

}

DoorClosing

S8：door closing到door open。事件：门内开门按键被点亮，电梯开门。

代码：

if (GetOpenDoorLight())

{

SetDoor(GetNearestFloor(), true);

SetOpenDoorLight(false);

\*state = DooOpen;

printf("Translation:from DoorClosing to DoorOpen.\n");

}

if (GetCloseDoorLight())

{

SetCloseDoorLight(false);

}

if (IsBeamBroken())

{

SetDoor(GetNearestFloor(), true);

\*state = DoorOpen;

Printf("translation:from DoorClosing to DoorOpen.\n");

}

S9：door closing到idle。关门结束后，电梯没有下一步动作，处于空闲的状态。

代码：

if (IsDoorClosed(GetNearestFloor()))

{

SetDoor(GetNearestFloor, false);

\*state = Idle;

}