模拟电梯程序概要设计说明书

小组成员：谢非 史柠玮 旦增次旦

联系方式（QQ号）：谢非（1420984249）

史柠玮（1044950475）

###### 总述：

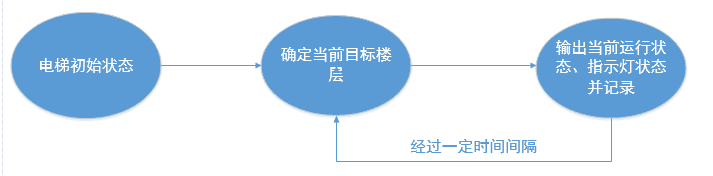
利用C语言编写一个在Windows环境下的可外部操作，并且将结果输出的模拟电梯程序。

###### 总体设计：

功能：

一台9层的电梯，用户可以在电梯内部或外部操作电梯，电梯具有若干种运行策略。

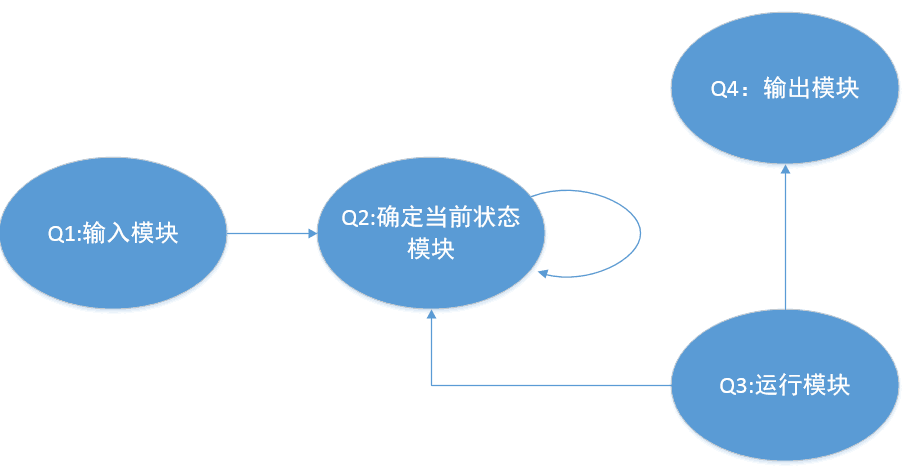
处理流程：



模块：

1. 确定当前状态模块;
2. 输入模块;
3. 运行模块;
4. 输出模块;

模块之间的关系：



###### 外部接口：

文件或控制台窗口或图形化界面窗口

###### 模块设计：

1. 确定当前状态模块(**State.c**)

定义全局变量

currentFloor表示当前楼层

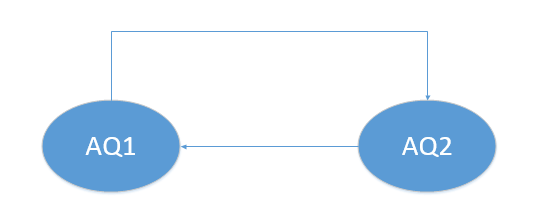
goalFloor表示目标楼层

定义结构

typedef struct CommandList{  
int floor\_Number;  
int up\_Down;  
struct CommandList \*next;  
}list;

|  |  |  |
| --- | --- | --- |
| 函数声明 | 作 用 | FSM代号 |
| void judgeUpDown\_State（int currentFloor，int goalFloor） | 判断上行还是下行并执行 | AQ1 |
| void judgeGoalFloor1\_State（struct cmdSet \*） | 策略1判断得到当前目标楼层 | AQ2 |
| void judgeGoalFloor2\_State（struct cmdSet \*） | 策略2判断得到当前目标楼层 | AQ2 |
| void judgeGoalFloor3\_State（struct cmdSet \*） | 策略3判断得到当前目标楼层 | AQ2 |
| void judgeGoalFloor4\_State（struct cmdSet \*） | 策略4判断得到当前目标楼层 | AQ2 |
| void timeRecord\_State() | 记录电梯运行时间和楼层数据,最终形成记录文件 |  |
|  |  |  |

FSM如下:



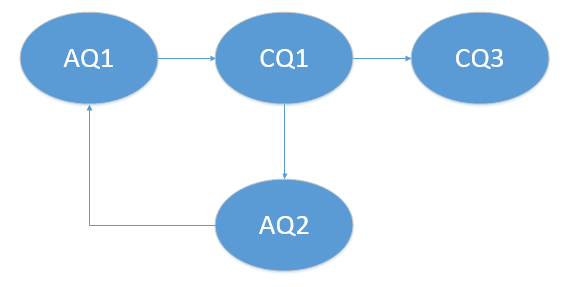
1. 输入模块(**In.c**)

|  |  |  |
| --- | --- | --- |
| 函数声明 | 作 用 | 状态转换图代号 |
| void file\_In(FILE \*) | 读取文件中的操作数据 |  |
| void win\_In() | 第N种策略的即时输入 |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. 运行模块(**Run.c**)

|  |  |  |
| --- | --- | --- |
| 函数声明 | 作 用 | 状态转换图代号 |
| void upper\_Run() | 每隔1秒移动上行一层 | CQ1 |
| void downer\_Run() | 每隔1秒移动下行一层 | CQ2 |
| void pause\_Run() | 电梯到达目标楼层后停止 | CQ3 |
| void vacant\_Run() | 电梯空闲（关门无人） | CQ4 |

FSM如下:



1. 输出模块(**Out.c**)

|  |  |  |
| --- | --- | --- |
| 函 数 声 明 | 作 用 |  |
| void file\_Out() | 在文件中输出当前位置 |  |
| void win\_Out () | 窗口输出当前位置 |  |
| void pilot\_Out () | 操作指示灯 |  |
| void timer\_Out () | 记录当前运行状态时间 |  |
|  |  |  |

###### 数据结构：

1. 先来先服务策略：用队列储存命令;
2. 顺便服务策略：指示灯储存;
3. …

###### 容灾设计：

1. 如果输入违反协议，提示错误并要求重新输入;
2. …

###### 制定规范：

1. 函数名称：功能名\_模块名;
2. 注释风格：在函数定义前一行/\*\*/注释，其他单句注释紧跟//
3. 函数定义：每个模块放在不同的.c文件中
4. 代码风格：…