**<14j>小组<2016火车调度模拟系统>概要设计**

**版本号：14j-Design-03**

**编制时间：2016年5月29日**

**编制人员：罗暄澍 谢天 刘润程**

**1 用户界面设计**

**1.1文件方式**

**A．【文件名称】模拟火车调度系统**

**B．【文件格式】.c文件**

**【文件使用/产生说明】**程序运行时，先提示是否载入默认配置，否则首先出现初始化信息录入界面，输入初始化信息。然后程序根据这些参数绘制主画面。在主画面中，鼠标第一次点击小火车时，小火车启动运行，再次点击小火车时，火车暂停。火车申请进入公共轨道时，探测点点亮。

**1.2动画方式**

**2.有限状态自动机状态转换图**

**2.1【一级状态图】**

行驶

停止

加速/减速

**2.2【二级状态图】**

暂停（客户命令）

暂停（系统调度）

非公共轨道

公共轨道

收到客户命令

驶入灯信号

驶入灯信号

系统命令

**3.高层数据结构设计**

**3.1常量定义**

define \_TRAINHEADFILE\_H\_

**3.2全局变量定义**

TrainInfo train[4];

int i;

int run\_method;

FILE \*setting;

int temp;

char tempinput;

**3.3全局数据结构定义**

struct POSITION{

int x;

int y;

};

typedef struct POSITION Position;

struct Traininfo{

int train\_num;

int train\_type;

int train\_direction;

Position train\_datum\_point;

int train\_start\_1D;

Position train\_start\_2D;

int train\_detect\_1D\_1;

Position train\_detect\_2D\_1;

int train\_detect\_1D\_2;

Position train\_detect\_2D\_2;

int train\_detect\_1D\_3;

Position train\_detect\_2D\_3;

int train\_detect\_1D\_4;

Position train\_detect\_2D\_4;

int train\_start\_time;

int train\_speed;

int train\_pause\_time;

int train\_distance\_1D;

Position train\_distance\_2D;

int train\_next;

int train\_orbit\_L;

int train\_orbit\_W;

int train\_state;

int station1;

int station2;

struct Traininfo \*tptr1;

struct Traininfo \*tptr2;

};

**4.系统模块划分**

**4.1系统模块划分**

（1）本系统程序部分划分为Main.c,CreateOrbit.c,Length to Coordinate.c,TrainFile.c,EnsureInfo.c,CompleteInfo.c,Finite State Machine.c,File Write.c,Thread.c,TrainHeadfile.h 十个模块，功能如下：

1.模块名称：Main.c

功能： 主函数，运行火车信息的初始化，运行计算机控制的中央调度系统，运行计算机控制的火车运行线程函数，运行计算机控制的信息输出。运行计算机控制的小火车的运行情况和用户控制的小火车运行情况（接收人工输入）。

1. 模块名称：CreateOrbit.c

功能：输入轨道1，2，3的参数并作图，输出数据（轨道1，2，3的参数及其探测点），赋值。

1. 模块名称：Length to Coordinate.c

功能：将长度换为火车坐标。

1. 模块名称：TrainFile.c

功能：默认数据的输入，快车慢车的储存以及运行速度的判断，各车的运行速度，起始时间，运行方向，起始位置，停靠时间和类型。

1. 模块名称： EnsureInfo.c

功能：输出信息（车号/轨道号。快慢，方向轨道基准点，启动位置，探测点，坐标，启动时间，速度，停靠时间，所在轨道）

1. 模块名称： CompleteInfo.c

功能：指针指向重合轨道火车，确定正方向，初始位置确定，初始化状态

1. 模块名称： Finite State Machine.c

功能：修改状态，运行中，等待进站，在站台等待，进站，出站，执行操作，更新2维坐标，判断下一秒是否进入探测区。

1. 模块名称： File Write.c

功能：文件输入

1. 模块名称： Thread.c

功能：火车运行

10.模块名称：outputCli.c

功能：命令行输出：当前时间，当前策略，3辆火车运行状态，3辆火车当前位置，公共轨道的状态

11.模块名称： TrainHeadfile.h

outputCli.c

TrainFile.c

功能：头文件

CompleteInfo.c

CreateOrbit.c

Main.c

Finite State Machine.c

Length to Coordinate.c

Thread.c

TrainFile.c

EnsureInfo.c

（2）文件及函数组成

|  |  |  |  |
| --- | --- | --- | --- |
| 源文件 | 源文件说明 | 函数名 | 功能 |
| Main.c | 主函数，运行火车信息的初始化，运行计算机控制的中央调度系统，运行计算机控制的火车运行线程函数，运行计算机控制的信息输出。运行计算机控制的小火车的运行情况和用户控制的小火车运行情况（接收人工输入）。 | int main() | 定义变量  轨道参数  火车参数  完善参数  确认信息  执行策略 |
| CreateOrbit.c | 输入轨道1，2，3的参数并作图，输出数据（轨道1，2，3的参数及其探测点），赋值。 | void LoadInfo(TrainInfo\* train1,TrainInfo\* train2,TrainInfo\* train3) | 输入轨道参数  作图  输出数据  赋值 |
| Length to Coordinate.c | 将长度换为火车坐标。 | Position Length\_To\_Coordinate(TrainInfo a) | 长度换成坐标 |
| TrainFile.c | 默认数据的输入，快车慢车的储存以及运行速度的判断，各车的运行速度，起始时间，运行方向，起始位置，停靠时间和类型。 | void SetDefaultSet(TrainInfo set[]) | 默认数据输入 |
| void OutputSet(TrainInfo set[],int order) | 输出运行速度，起始时间，运行方向，起始位置，停靠时间和类型 |
| void InputSet(TrainInfo set[],int order) | 默认数据的输入，快车慢车的储存以及运行速度的判断 |
| EnsureInfo.c | 输出信息（车号/轨道号。快慢，方向轨道基准点，启动位置，探测点，坐标，启动时间，速度，停靠时间，所在轨道） | void EnsureInfo(TrainInfo\* train) | 输出信息（车号/轨道号。快慢，方向轨道基准点，启动位置，探测点，坐标，启动时间，速度，停靠时间，所在轨道） |
| CompleteInfo.c | 指针指向重合轨道火车，确定正方向，初始位置确定，初始化状态 | void CompleteInfo(TrainInfo\* train1,TrainInfo\* train2,TrainInfo\* train3) | 指针指向重合轨道火车，确定正方向，初始位置确定，初始化状态 |
| Finite State Machine.c | 有限状态自动机 | void ControlTrain(TrainInfo\* train) | 修改状态，运行中，等待进站，在站台等待，进站，出站，执行操作，更新2维坐标 |
| int ReadytoJudge(TrainInfo a) | 判断下一秒是否进入探测区。 |
| File Write.c | 文件输入 | void FileWrite(TrainInfo set[],FILE \*fp,int temp) | 文件输入 |
| outputCli.c | 输出模块 | void PrintTrainInfo(TrainInfo\* train,int OutputTime) | 命令行输出 |
| int IsInPubluic(TrainInfo a) | 判断是否占用公共轨道 |
| Thread.c | 线程 |
| TrainHeadfile.h | 火车系统的头文件 |