# 碰撞预警系统详细设计说明书

时间：2018-7-13

版本：V1.0

|  |  |  |
| --- | --- | --- |
| 文件状态 | 文件标识 |  |
| [ ]草稿  [ ]正式发布  [ ]正在修改 | 当前版本 | V1.0 |
| 作者 |  |
| 完成日期 | 2018-7-18 |

**版本历史**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 作者 | 参与者 | 起止时间 | 备注 |
| V1.0 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**修改记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 日期 | 修订版本 | 参与者 | 修改描述 | 作者 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 一、系统总体设计

为了减少交通事故，保障车辆行驶安全，提高道路通行率，我们设计了基于RSU的碰撞预警系统，本系统主要用于路口有遮挡物或其他因素引起的视线受阻的场景，如：多雾、风沙、暴雨等恶劣天气。系统在司机视线受阻无法正常观测路面及其他车辆情况时，RSU根据收集到的信息处理后提前对车辆发出碰撞预警，能够有效减少交通事故的发生，保障司机的生命安全。

### 1.1系统架构图



Figure 1 系统架构图

如Fig.1 所示，本碰撞预警系统架构为A、B两辆车与RSU通过DSRC通信，向RSU传输车辆的位置L、速度V、方向D、加速度a、时间戳T等信息。RSU收集到两辆车采集的信息后对其进行处理、计算，得到A、B车将会发生碰撞的时间和两车的运行轨迹。最后RSU再通过DSRC将结果发送回车辆A、B，若碰撞时间小于阈值同时会发出碰撞警报。

### 1.2软件功能模块图



Figure 2 软件功能模块图

如Fig.2所示，本碰撞系统分为三个模块，分别为：OBU信息采集模块、RSU信息处理模块和DSRC通信模块。

**（1）OBU信息采集模块：**

本模块为车载终端，主要负责车辆行驶信息的采集、传输以及显示，由三个子模块：信息采集模块、UDP通信模块和显示界面模块组成。这些子模块在运行Android系统的手机上实现。为了完成信息的采集，手机需具有摄像头、陀螺仪、GPS等设备。

子模块的分工为：

* 信息采集模块，通过手机上的设备采集车辆的位置、速度、加速度与行驶方向等信息；
* UDP通信模块，将信息采集模块采集到的信息与手机MAC地址、采集时间点传输至OBU；同时还能将OBU接收到的信息传输至显示界面模块。
* 显示界面模块，将从OBU接收到的预警信息（碰撞时间、预测轨迹等）显示到屏幕上，让司机能够直观接收到路况信息，提前完成刹车、避让等措施，防止交通事故的发生。

**（2）RSU信息处理模块**

本模块主要负责接收信息采集模块传递的车辆行驶信息以及信息的处理，得到预计的碰撞时间以及碰撞路线，然后将得到的信息通过DSRC通信模块返回到信息采集模块。

因此，本模块共分为两个子模块：

* UDP通信模块，将信息采集模块采集到的信息传输至子模块信息处理模块；同时还能将信息处理子模块求得的信息传输至RSU上。
* 信息处理模块能够接受UDP通信模块传递的信息并通过一定的算法进行处理，提前判断两车是否会在接下来某一时间段发生碰撞，如果会发生碰撞将求得可能的碰撞时间和碰撞轨迹，并将这些信息以及碰撞警报通过UDP通信模块反馈给RSU。

**（3）DSRC通信模块**

专用短程协议，由路边单元RSU和车载单元OBU、控制中心以及一些辅助设备组成，能够实现RSU和OBU之间的无线短程通信，保证信息安全可靠传输。

## 二、程序设计说明

### 2.1 OBU信息采集模块

#### 2.1.1模块描述

该模块嵌入在行驶车辆的移动终端上，我们初步选择Android系统的智能手机作为终端，因为它具有摄像头、陀螺仪、GPS、显示屏等设备。模块程序在Android系统上开发实现。在后期可能会直接将模块嵌入汽车自带的系统上。

#### 2.1.2功能

负责车辆行驶信息的采集、传输以及显示。

#### 2.1.3性能

#### 2.1.4输入、输出项目

#### 2.1.5模块的交互

#### 2.1.6模块设计

#### 2.1.7程序逻辑

#### 2.1.8接口

#### 2.1.9测试要点

### 2.2 RSU信息处理模块

#### 2.2.1模块描述

该模块为路边单元RSU上对信息进行接收和处理的模块，实时监督RSU覆盖范围内的车辆行驶状态，能够提前发出碰撞预警，减少交通事故的发生。

初步目标是将该模块嵌入到与RSU连接的pc端，用pc的处理器处理信息；最终目标是将模块嵌入到RSU的ARM上直接处理接收到的信息得到碰撞时间和路线。

#### 2.2.2功能

本模块主要负责对信息进行处理，得到预计的碰撞时间以及碰撞路线，然后将碰撞时间以及路线显示到PC端屏幕，并将得到的信息通过DSRC通信模块返回到信息采集模块。

#### 2.2.3性能

PC为用户提供了高效的信息计算、处理、显示与传输的功能。

#### 2.2.4输入、输出项目

输入输出我们使用JSON格式来表示。

**(1)UDP通信子模块**

将RSU接收的信息转发给信息处理子模块。

输入输出项目均为：两辆车Android手机端采集到的信息，包含如表1中所含的各个参数。

表1：车辆信息

|  |  |  |  |
| --- | --- | --- | --- |
| 信息 | Item | Type | 备注 |
| MAC地址 | “id” | int |  |
| 时间戳 | “time” | int | 单位：秒（s） |
| 速度 | “speed” | double | 保留3位小数，单位（m/s） |
| 方向 | “direction” | char | “E””W””N””S” |
| GPS | “lat” | double | 保留3位小数 |
| “lon” | double | 保留3位小数 |
| 加速度 | “ace” | double | 保留1位小数，单位（m/s2） |

输入输出项目还包括如表2所示的RSU信息：

表2：RSU信息

|  |  |  |  |
| --- | --- | --- | --- |
| 信息 | Item | Type | 备注 |
| 时间戳 | “time” | int | 单位：秒（s） |
| GPS | “lat” | double | 保留3位小数 |
| “lon” | double | 保留3位小数 |
| 地图 | “map” |  |  |

**(2)信息处理子模块**

输入项目就是表1和表2的信息；

输出项目：

表3：碰撞信息

|  |  |  |  |
| --- | --- | --- | --- |
| 信息 | Item | Type | 备注 |
| MAC地址 | “id” | int |  |
| 碰撞时间 | “time” | int | 单位：秒（s） |
| 碰撞距离 | “distant” | double | 保留3位小数、单位（m） |
| 碰撞轨迹 | “lat\_array” | double | 保留3位小数 |
| “lon\_array” | double | 保留3位小数 |
| 地图 | “map” |  |  |
| 警报 | “warn” | boolean | T：碰撞；F：不碰撞 |

**(3)显示界面子模块**

输入项目：如表3所示；

输出项目：用图形界面将输入表示出来，如fig.3所示。



Figure 3：PC端显示界面

#### 2.2.5模块的交互



Figure 4: RSU信息处理模块

#### 2.2.6模块（算法）设计

#### 2.2.7程序逻辑



Figure 5： UDP通信子模块的程序逻辑



Figure 6： 信息处理子模块的程序逻辑

#### 2.2.8接口

**(1)UDP通信子模块**

该子模块分为服务器端和客户端，由两个函数分别实现；

**服务器端**:从客户端接收并处理信息，并向客户端发送碰撞信息。

* 调试环境：QT
* 程序名称：myUDP\_server.cpp
* 服务器IP地址：xx.xx.xx.xx
* 服务器端口：5050（自己配置）
* 程序功能：

①从端口5050接收客户端发送来的数据,接收成功后处理从客户端收到的数据, 客户端的IP地址和端口号;

②给客户端发送碰撞信息。

**客户端**:向服务器发送车辆信息，同时也接收从服务器发送的碰撞信息并进行显示。

* 调试环境：QT
* 程序名称：myUDP\_client.cpp
* 客户端IP地址和端口：yy.yy.yy.yy(与服务器端同一个网段)
* 客户端口：4040（自己配置）
* 程序功能：

从端口4040向服务器发送数据、接收数据并进行显示

**(2)信息处理子模块**

**(3)显示界面子模块**

#### 2.2.9测试要点

### 2.3 DSRC通信模块

#### 2.3.1模块描述

专用短程协议，由路边单元RSU和车载单元OBU、控制中心以及一些辅助设备组成，能够实现RSU和OBU之间的无线短程通信，保证信息安全可靠传输。

#### 2.3.2功能

负责OBU信息采集端与RSU信息处理端之间的可靠传输功能。

#### 2.3.3性能

#### 2.3.4输入、输出项目

#### 2.3.5模块的交互

#### 2.3.6模块设计

#### 2.3.7程序逻辑

#### 2.3.8接口

#### 2.3.9测试要点