**碰撞预警系统概要设计说明书**

时间：2018-7-13

版本：V2.0

一、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、速度、方向等信息，RSU通过收集到车辆采集的信息，并对其进行处理后得到将会发生碰撞的时间，RSU再通过DSRC将结果发送回车辆A、B。

二、场景图



Figure 2 场景图

A、B两车在有视线障碍的十字路口可能会发生碰撞。

三、功能模块图



Figure 3 功能模块图

四、碰撞预警系统项目时间安排

总计划时间：7月12日至8月16日（五周）

Table 1 项目时间安排表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 周数 | 星期四 | 星期五 | 星期六 | 星期天 | 星期一 | 星期二 | 星期三 | Linux | Android |
| 1 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | UDP通信模块 |  |
| 2 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 信息处理模块 | 信息采集模块 |
| 3 | 26 | 27 | 28 | 29 | 30 | 31 | 1 | 信息显示模块 | UDP通信模块 |
| 4 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |  | 界面展示模块 |
| 5 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 测试、debug | |

红色虚线框：许新操，周易

蓝色虚线框：许新操，张浪

红色虚线框为连接RSU的Linux PC端程序设计时间，主要为UDP通信模块与信息处理模块，其中的重点为信息处理模块，考虑如何设计信息处理模块处理第二节中的三种场景，首先完成场景一中的数据处理功能，然后再进行添加功能。当Linux PC端的程序设计完成时，可考虑移植到RSU中的ARM Linux中，此时需重写里面的程序，可能要花费2周时间，若能重写，那RSU就不需要外联PC做逻辑处理。

蓝色虚线框为连接OBU的Android端程序设计时间，主要为信息采集模块、UDP通信模块与界面展示模块，以及最后的测试、debug时间。信息采集模块与UDP通信模块已有实现代码，主要需要对原有代码进行模块化重构。

Table 2 详细计划表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **周次** | **星期四** | | **星期五** | | **星期六** | | **星期天** | | **星期一** | | **星期二** | | **星期三** | |
| 1 | **12** | **讨论具体数据** | 13 |  | 14 |  | 15 |  | 16 |  | **17** | **完成PC端UDP通信模块** | **18** | **完成详细设计说明书** |
|  |  |  |  |  |  |  |
| 2 | 19 |  | **20** | **模拟数据生成** | 21 |  | 22 |  | 23 |  | **24** | **信息处理模块实现** | **25** | **信息采集模块**  **重构实现** |
|  |  |  |  |  |  |  |
| 3 | 26 |  | **27** |  | 28 |  | 29 |  | 30 |  | **31** | **信息显示模块完成** | **1** | **Android端UDP通信模块重构实现** |
|  |  |  |  |  |  |  |
| 4 | 2 |  | **3** | **界面展示模块实现** | 4 |  | 5 |  | 6 |  | 7 |  | **8** | **测试、写测试文档** |
|  |  |  |  |  |  |  |
| 5 | 9 |  | 10 |  | 11 |  | 12 |  | 13 |  | 14 |  | **15** | **Debug** |
|  |  |  |  |  |  |  |