中兴Android系统界面软件设计与开发

**项目开发总结报告**

**版本：1.0**

编写： 张 翔

校对： 李凤彬

审核： 杨 帆

批准： 朱怡安

**西北工业大学－中兴通讯嵌入式系统联合开发实验室**

**2012年3月**

**目 录**

[1引言 3](#_Toc317842587)

[1.1文档标识 3](#_Toc317842588)

[1.2项目概述 3](#_Toc317842589)

[1.3文档概述 3](#_Toc317842590)

[1.4参考文档 4](#_Toc317842591)

[2技术难点 5](#_Toc317842592)

[3设计方案的优点 7](#_Toc317842593)

[4不足之处 8](#_Toc317842594)

[5下一步改进 8](#_Toc317842595)

# 1引言

## 1.1文档标识

中文名称：《项目开发总结报告》。

英文名称：“Project Development Summary Report（PDSR）”。

文档版本：“1.0”。

文档编号：“SSM-ZTE-AndroidUI-PDSR-1.0(E)”。

## 1.2项目概述

本文档适用于“中兴Android系统界面软件设计与开发”项目（以下简称“AndroidUI项目”）的开发过程。AndroidUI项目由中兴通讯股份有限公司（以下简称“中兴通讯”）提出，由西北工业大学－中兴通讯嵌入式系统联合开发实验室（以下简称“联合实验室”）负责实施，该项目标识号为“SSM-ZTE-AndroidUI”，其软件产品版本号为“1.0”，包括三个内部版本，分别是0.1版、0.2版和0.3版。

项目内容为：

基于Android 3.2系统版本，针对中兴通讯设计开发的平板电脑，联合实验室依据中兴通讯提出的用户需求和设计的软件界面，深度定制Android系统的人机界面及应用软件界面，完成中兴Android系统界面软件的设计与开发。

## 1.3文档概述

本文档依据国家标准[《GB/T 8567-2006计算机软件文档编制规范》](../../资料/GBT%208567-2006%20计算机软件文档编制规范.pdf)制定，属于技术文档，仅限于联合实验室和中兴通讯的项目相关人员阅读。

单击此处输入文字。

## 1.4参考文档

* [《GB/T 8567-2006计算机软件文档编制规范》](../../资料/GBT%208567-2006%20计算机软件文档编制规范.pdf)，国家标准
* [《高校合作项目要求说明书－中兴Android系统界面软件设计与开发（讨论稿）》，](../../资料/高校合作项目要求说明书－中兴Android系统界面软件设计与开发（讨论稿）.doc)中兴通讯提供
* [《SSM-ZTE-AndroidUI-SDP-1.0(E) 软件开发计划》](../1－项目计划阶段/SSM-ZTE-AndroidUI-SDP-1.0(E)%20软件开发计划.docx)，联合实验室编写
* [《SSM-ZTE-AndroidUI-FAR-1.0(E) 可行性分析报告》](../2－方案探索阶段/SSM-ZTE-AndroidUI-FAR-1.0(E)%20可行性分析报告.docx)，联合实验室编写
* [《SSM-ZTE-AndroidUI-PSS-1.0(E) 项目解决方案》](../2－方案探索阶段/SSM-ZTE-AndroidUI-PSS-1.0(E)%20项目解决方案.docx)，联合实验室编写
* [《SSM-ZTE-AndroidUI-SADD-0.1 软件结构设计说明》](../4－软件v0.1开发/SSM-ZTE-AndroidUI-SADD-0.1%20软件结构设计说明.docx)，联合实验室编写
* [《SSM-ZTE-AndroidUI-SPDD-0.1 软件物理设计说明》](../4－软件v0.1开发/SSM-ZTE-AndroidUI-SPDD-0.1%20软件物理设计说明.docx)，联合实验室编写
* [《SSM-ZTE-AndroidUI-SADD-0.2 软件结构设计说明》](../5－软件v0.2开发/SSM-ZTE-AndroidUI-SADD-0.2%20软件结构设计说明.docx)，联合实验室编写
* [《SSM-ZTE-AndroidUI-SPDD-0.2 软件物理设计说明》](../5－软件v0.2开发/SSM-ZTE-AndroidUI-SPDD-0.2%20软件物理设计说明.docx)，联合实验室编写
* [《SSM-ZTE-AndroidUI-SADD-0.3 软件结构设计说明》](../6－软件v0.3开发/SSM-ZTE-AndroidUI-SADD-0.3%20软件结构设计说明.docx)，联合实验室编写
* [《SSM-ZTE-AndroidUI-SPDD-0.3 软件物理设计说明》](../6－软件v0.3开发/SSM-ZTE-AndroidUI-SPDD-0.3%20软件物理设计说明.docx)，联合实验室编写
* [《SSM-ZTE-AndroidUI-PPR-1.0 项目进度报告》](../SSM-ZTE-AndroidUI-PPR-1.0%20项目进度报告.docx)，联合实验室编写

# 2技术难点

本项目开发过程中面临并解决了如下几个技术难点。

1. **ZTE小部件管理机制：**

**Launcher**

**LauncherZTEWidgetHost**

**LauncherZTEWidgetHostView**

**LauncherZTEWidgetInfo**

**Time Weather Widget**

**SNS Widget**

**MicroBlog Widget**

**Bookmark Widget**

项目需要开发ZTE定制的桌面小部件，这些内容与Android系统自带的桌面小部件的管理机制不同，为此，我们制定了如下的ZTE小部件管理机制：由Launcher通过LauncherZTEWidgetHost创建自定义小部件的视图LauncherZTEWidgetHostView，同时LauncherZTEWidgetHostView与每一个小部件的视图相关联，LauncherZTEWidgetInfo用于描述小部件的基本信息，如大小、位置等。

1. **滑屏**

Android提供的Animation技术只支持简单的缩放、旋转及平移操作，无法实现翻卷的效果。经过我们研究，实现3D的翻卷效果最理想的技术是OpenGL ES，但是Android的GLSurfaceView无法与其他基本控件进行交互，所以以上两种提及的技术都无法用于滑屏的实现，只能通过数学变换操纵图像变形来达到这种效果。具体而言，就是将实时获取的屏幕截图划分为2×25的网格，定位到每一个单元格，通过绘制贝塞尔曲线来实现翻卷效果，绘制采用drawBitmapMesh函数。

1. **SNS小部件**

折纸效果采用了OpenGL技术实现，由于Android的GLSurfaceView在GLThread中渲染，其他基本控件在UI Thread中渲染，因此一旦桌面上加入SNS小部件，滑屏时会出现不同步现象。所以实现时加入了精确的控制算法，正常情况下隐藏GLSurfaceView，并将GLSurfaceView中的内容传至基本控件中显示，一旦检测到用户在小部件上的滑动事件即显示GLSurfaceView，根据用户的滑动距离计算折纸效果。

1. **动态壁纸**

动态壁纸的实现采用了RenderScript技术，提供了一组高性能的3D图形渲染和计算API，底层代码使用C语言编写。RenderScript有三层代码实现，分别为本地代码层、反射层以及Android Framework层。本地代码层主要处理大量数据计算和图形渲染工作。反射层是本地代码反射类集合，对本地代码进行基本封装，使得Android Framework层可以和本地代码层进行交互。Android构建工具可以自动生成代码，减少了JNI代码的使用量。Android Framework层由Android Framework API组成，包含在android.renderscript包中。本层提供了更高级别的方法，如旋转视图，过滤位图等，本层API通过调用反射层代码来与本地层代码进行交互。

1. **任务管理器**

用户通过屏幕与模型进行交互需要将三维坐标系转换为二维坐标系，使用gluProject 函数，将模型在三维空间中的坐标，映射到屏幕的二维坐标系上，然后进行对比、变换、判断、完成交互，但无法直接得到作为函数参数的模型视图和投影矩阵，因为OpenGL ES本身不提供OpenGL2.0里面获得矩阵的接口，而且OpenGL ES 提供的诸如glTranslate、glRotate、glScale等方法是不保证性能的，如果要将一个由几万个三角形组成的模型进行几何变换，会造成很大的开销。解决方案是自定制矩阵，使用OpenGL ES 提供的一些矩阵操作接口直接对变换矩阵进行操作，并使用定制的矩阵实现几何变换，这样可以即时追踪矩阵信息，还可以保证几何变换的效率。

# 3设计方案的优点

下图是本软件的软件模块结构图。

**Launcher**

**LiveWallpaperPicker**

**锁 屏**

**任务管理器**

**ZTEWidgetManager**

**三 维 时 间 天 气**

**网 络 书 签**

**社 交 网 络**

**微 博**

**星 光 飘 移**

**实 时 时 间 天 气**

由上图可以看出，通过Launcher将各个模块整合到一起，Launcher分别与任务管理器、锁屏、LiveWallpaperPicker和ZTEWidgetManager进行交互。Android原生的LiveWallpaperPicker负责调用动态壁纸。参考AppWidgetManager实现机制，在Launcher中引入ZTE小部件管理机制，使得在小部件中可以实现各种动画效果，甚至可以嵌入GLSurfaceView。在Launcher的选项菜单以及应用程序菜单中分别调用任务管理器和锁屏应用。

该设计方案具备较好的可扩展性、可维护性。

# 4不足之处

1. 由于将所有的ZTE小部件的实现加入了Launcher中，对Launcher的整体性能影响较大，尤其是时间天气Widget动画效果复杂，社交网络Widget引入了openGL，在Launcher中开启了GLThread线程，因此一旦桌面上加入过多的ZTE小部件，Launcher可能会出现堆溢出异常。
2. 滑屏的实现需要实时获取多桌面的截图，该过程对堆的占用较大。
3. 任务管理器中正在运行程序的缩略图的获取需要系统级的权限，而在开发普通应用时无法获得该权限，因此程序缩略图用了其他图片代替。

# 5下一步改进

基于Android 4.0，对整套系统界面进行深度定制。具体而言，就是将ZTE小部件的实现与Launcher分开，每个ZTE小部件做成单独的应用并实现具体的功能，而不是简单的演示Demo，同时放开Android系统Framework层中RemoteView对控件类型的限制，使得在原有AppWidget机制中可以加入自定义控件，这样可以有效地避免ZTE小部件对Launcher性能的影响。