嵌入式课程设计实验报告

学号：3170913022

姓名：杨春发

班级：计171

目录

[1. 选题理由 3](#_Toc28523242)

[2. 题目要求 3](#_Toc28523243)

[3. 可行性分析 3](#_Toc28523244)

[4. 需求分析 3](#_Toc28523245)

[5. 概要设计 4](#_Toc28523246)

[5.1 总体模块划分 5](#_Toc28523247)

[5.2 数据库设计 5](#_Toc28523248)

[5.3 核心算法设计 5](#_Toc28523249)

[6. 详细设计 7](#_Toc28523250)

[6.1 界面设计 7](#_Toc28523251)

[6.2模块设计 8](#_Toc28523252)

[7. 编码与测试 9](#_Toc28523253)

[8. 总结 13](#_Toc28523254)

1. 选题理由

随着3G技术的成熟和智能手机的不断普及,移动应用的需求与日俱增,移动应用开发成为当下最热门的技术之一-。在google和android手机联盟的共同推动下, android在众多移动应用开发平台中脱颖而出。Android操作系统是google最具杀伤力的武器之一。苹果以其天才的创新,使得iphone在全球迅速拥有了数百万忠实的“粉丝”，而android作为第一个完整 、开放、免费的手机平台，使开发者在为其开发程序时拥有更大的自由，。与windows mobile , symbian等厂商不同的是, android操作系统是免费向开发人员提供，这样可节省近三层得成本。我们知道,无论是产品还是技术，商业应用都是它最大的发展动力。android如此受到厂商与开发者的青睐，它的前景一片光明。国内对于安卓这方面的市场还没有完全开发出来。只要我们努力学习android,对理想、对生活都会有很大的补益。因此,开发这样一套很简单的音乐播放器软件对我们来讲是-件很有必要的事情,在下面的各章中我们将以开发安卓音乐播放器软件为例,谈谈其开发过程和所涉及到的问题及解决方法。

2. 题目要求

实现一个简易的MP3播放器，主要功能包括：

1. 自动读取SD卡的所有歌曲
2. 实现MP3的播放、暂停、上一曲、下一曲的功能
3. 通过进度条实时显示播放进度

3. 可行性分析   
1)技术可行性--- -本项目 仅需要一台装有Android Studio及配置好android开发环境的计算机即可,对计算机本身有一定的要求,它可以使用windows xp及其以上版本, macos, linux等操作系统;  
2)经济可行---由于本软件开发只需要一台电脑就行，基本不需要太大的成本。在经济上完全可行;  
3)操作可行性-- - -界面设计时充分考虑用户的习惯以及手机方面的局限性,程序必须要很简单;本程序只涉及到开发环境配置,安卓组件的熟悉，以及安卓平台上的数据存储，并没有太大的难度,所以,该项目完全有可操作

4. 需求分析

1. 功能需求

　　系统的功能需求，我们分成三个主要部分，即播放器的基本控制需求、播放列表管理需求、播放器友好性需求。以下分别描述：

* 1. 播放器的基本控制需求

　　播放：点击播放列表中的歌曲进行播放，播放时跳转播放界面，并显示歌曲信息，同时会匹配歌词，把相应的歌词显示在歌词面板中。

　　停止：当歌曲正在播放时，点击歌曲播放界面下方的停止键，停止当前播放的歌曲，同时歌词面板中的歌词清空。

　　暂停：当歌曲正在播放时，点击歌曲播放界面下方的暂停键，暂停当前播放的歌曲，同时歌词停止滚动。

　　上一曲：歌曲正在播放，点击上一曲键，播放列表中上一首歌曲。

　　下一曲：歌曲正在播放，点击下一曲键，播放列表中下一首歌曲。

　　播放列表：点击进入歌曲列表界面。

　　播放顺序：点击不同的播放顺序，歌曲就有不同的播放顺序。

1. 播放列表管理需求

　　添加歌曲：在菜单选项中，点击添加歌曲使程序进入手机内存、存储卡，选取要添加的歌曲。

　　移除歌曲：在菜单选项中，点击移除歌曲，界面跳转至歌曲列表，选取要移除的歌曲，点击确定，移除歌曲。

　　设置：在菜单选项中，选择设置，进入设置界面。

　　设为铃声：歌曲列表中，点击歌曲，弹出对话框，选择设为铃声。

　　音效设置：菜单选项中，点击进入音效设置界面。

　　歌曲收藏：歌曲选项中，选择收藏，收藏该歌曲。

1. 播放器友好性需求

　　视觉外观

　　使用播放器时，播放器根据歌曲文件的特性，通过视觉外观动态显示，外观优美。

　　系统稳定

　　使用播放器时能显示播放器的状态，是播放还是暂停，播放音乐时稳定。

　　后台播放

　　使用播放器时，能够进行后台播放，并且在手机桌面上显示 缩小化的播放器，歌词滚动显示。

　　播放模式

　　播放音乐时，更改播放模式能够顺利地、快速地、无差错地完成模式的转换。

1. 性能需求

　　播放器响应时间必须满足Android手机系统响应时间，因此 当要求歌曲播放、暂停、停止、上一曲、下一曲、进入列表等操作时，软件响应时间最长均不超过系统无响应时间。

1. [数据库](javascript:;)需求

　　Android自带的媒体库基于PacketVideo OpenCORE，该库支持多种常用的音频格式。编码格式MPEG4、MP3。

　　音乐文件的解析：对于常用的音乐文件格式进行解析，如：Mp3格式。

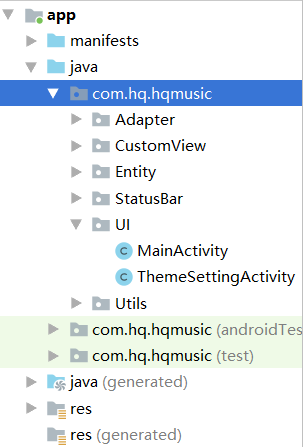
　　歌曲的存储：对于添加的歌曲进行存储，删除的歌曲进行移除。 错误信息的存储：对于系统检查到的错误进行存储记忆。

1. 可靠性要求

　　系统能持续运行，不影响其他程序的使用，不多占用内存，不会造成死机等问题。

5. 概要设计

## 5.1 总体模块划分



1. Manifest

AndroidManifest.xml:应用的配置文件，描述应用程序的名称、权限、能力和运行方式等。

1. Java

Com.ycf.ycfmusic下有以下几个包及其内容：

Adapter：MyAdapter适配器类

CustomView:下设顶部栏和底部栏用户视图的类

Entity： song歌手相关属性类

Status bar：下设播放进度类

Ui：下有mainActivity和ThemesettingActivity均为Activity,前者为播放器的主界面进行各种播放控制操作;后者用来播放器主题切换。;

Utils：设有MusicUtils类，扫描本地的音频文件

1. Res

外部资源文件夹主要为程序提供各种外部资源，png文件和xml文件

## 5.2 数据库设计

Android自带的媒体库基于PacketVideo OpenCORE，该库支持多种常用的音频格式。编码格式MPEG4、MP3。

　　音乐文件的解析：对于常用的音乐文件格式进行解析，如：Mp3格式。

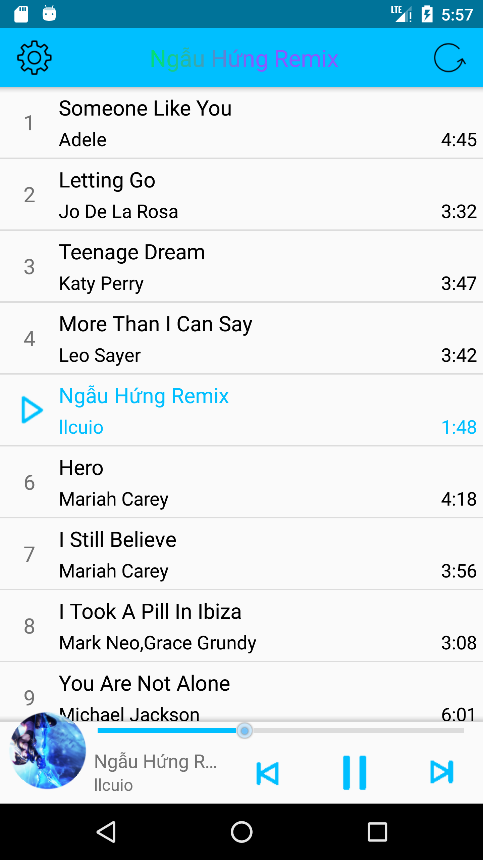
　　歌曲的存储：对于添加的歌曲进行存储，删除的歌曲进行移除。 错误信息的存储：对于系统检查到的错误进行存储记忆。

## 5.3 核心算法设计

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
  
 //获取权限  
 getAuthority();  
 sharedPreferences = getSharedPreferences("location", MODE\_PRIVATE);  
 // 主题设置  
 string\_theme = sharedPreferences.getString("theme\_select", "blue");  
 if (string\_theme.equals("blue")) {  
 setTheme(R.style.Theme\_blue);  
 } else if (string\_theme.equals("purple")) {  
 setTheme(R.style.Theme\_purple);  
 } else if (string\_theme.equals("green")) {  
 setTheme(R.style.Theme\_green);  
 } else {  
 setTheme(R.style.Theme\_red);  
 }  
 setContentView(R.layout.activity\_main);  
  
 // 顶部状态栏颜色设置  
 mTintManager = new SystemBarTintManager(MainActivity.this);  
 mTintManager.setStatusBarTintEnabled(true);  
 if (string\_theme.equals("blue")) {  
 mTintManager.setStatusBarTintResource(R.color.blue);  
 } else if (string\_theme.equals("purple")) {  
 mTintManager.setStatusBarTintResource(R.color.purple);  
 } else if (string\_theme.equals("green")) {  
 mTintManager.setStatusBarTintResource(R.color.green);  
 } else {  
 mTintManager.setStatusBarTintResource(R.color.red);  
 }  
  
 // 获得屏幕宽度并保存在screen\_width中  
 init\_screen\_width();  
  
 // 加载currentposition的初始数据  
 currentposition = sharedPreferences.getInt("currentposition", 0);  
  
 // 顶部视图控件的绑定  
 initTopView();  
  
 // 顶部和 底部操作栏按钮点击事件  
 setClick();  
  
 // 给textView1和textView2赋初值  
 initText();  
  
 // 加载popupwindow并设置相关属性  
 setpopupwindow();  
  
 // listview的绑定,数据加载,以及相关事件的监听  
 setListView();  
  
 // 设置mediaplayer监听器  
 setMediaPlayerListener();  
  
}

6. 详细设计

## 6.1 界面设计



1.实现扫描本地音乐

2.音乐的播放与控制

3.关联进度条seekbar，自定义seekbar

4.单曲循环，顺序播放，随机播放的实现

5.设置喜爱音乐

6.播放列表背景图设置与保存

7.实现APP主题换肤的功能

## 6.2模块设计

打开应用获取权限

是否获取成功？

提示并退出

否

是

顶部栏载入

扫描本地音乐

有无本地音乐？

无

有

播放列表及播放底部栏载入

音乐播放与控制

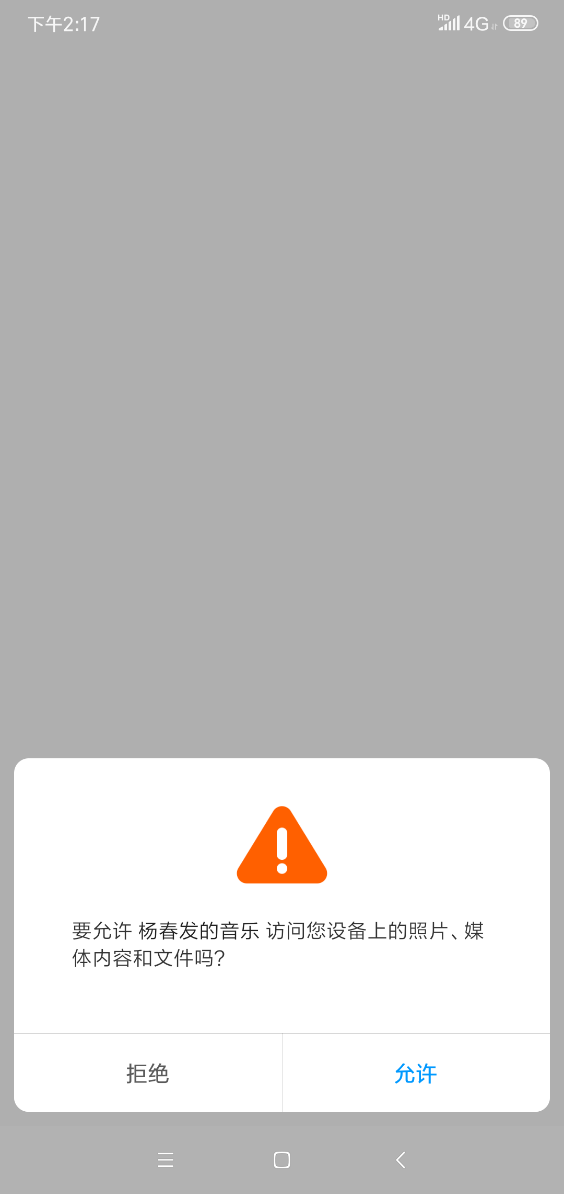
退出并提示

7. 编码与测试

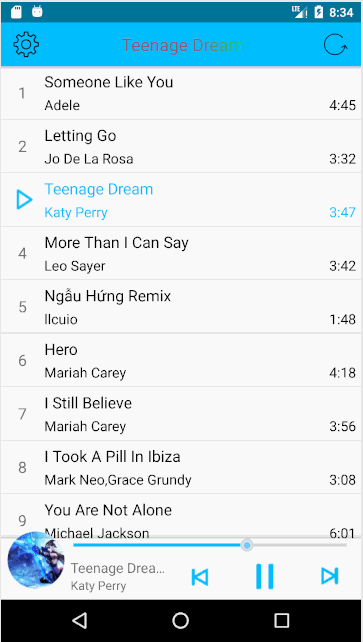
1. 安装效果：



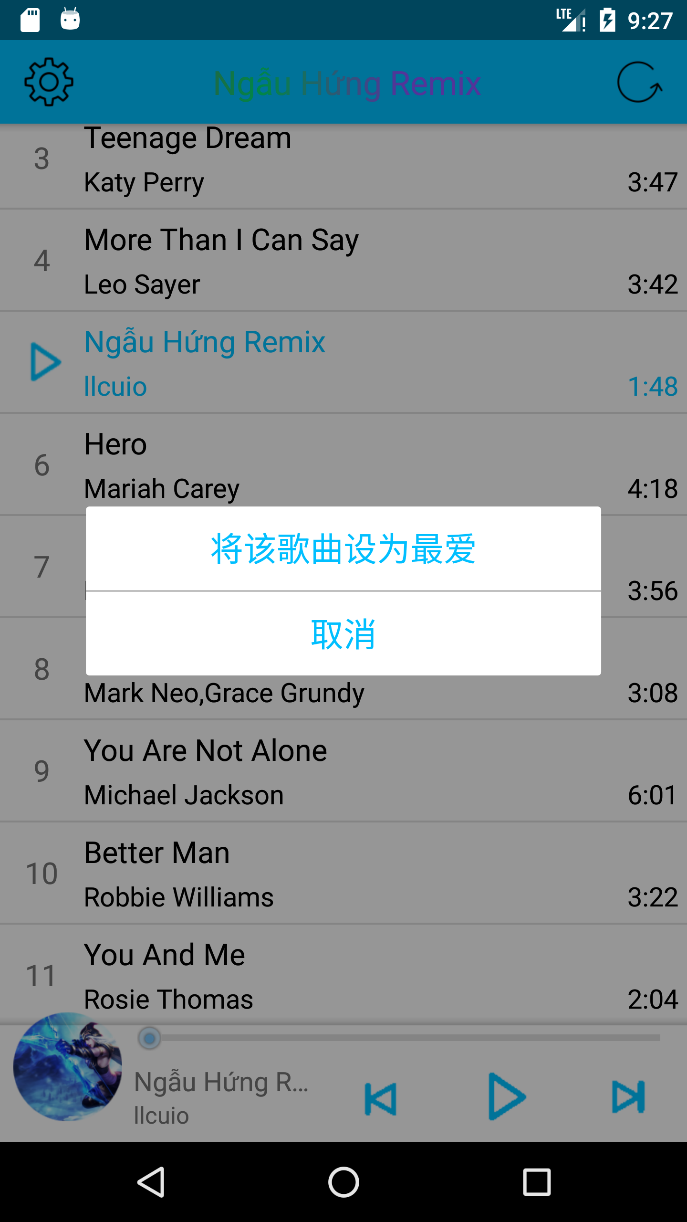
1. 权限获取：



1. 音乐播放：



1. 设置喜爱音乐：

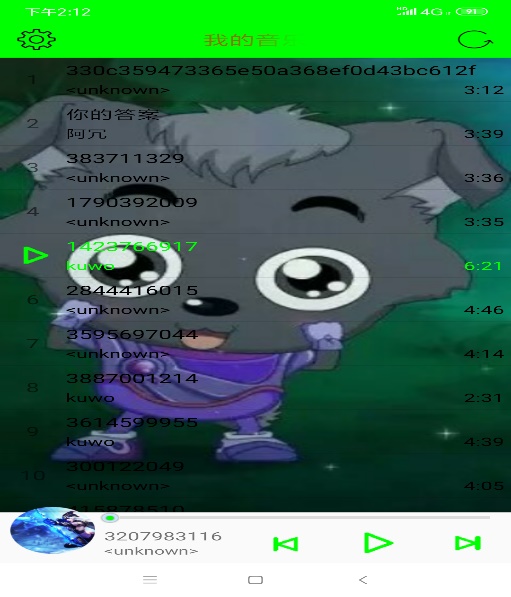


1. 设置栏实现



1. 更换列表背景：

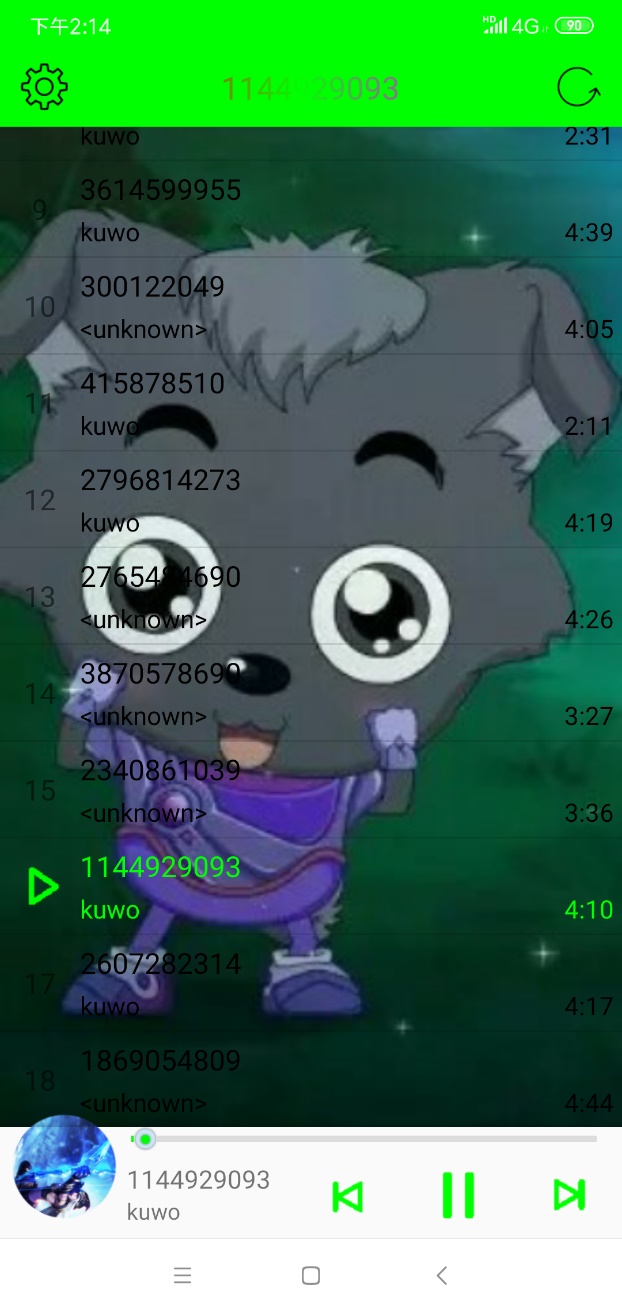




1. 更换主题效果：



1. 播放喜爱音乐



1. 退出应用：



8. 总结

嵌入式系统课程设计是一门紧跟时代步伐的课程，顺应了当下移动互联网的发展浪潮，这门课程使我们大学生将学习到的理论知识转化为社会所需要的才能，令我们获益匪浅。这次的作品在短短一周内，虽然并非一切顺利，但也算磕磕绊绊地完成了。途中遇到过难以解决的问题,虽然厌烦但还是逼着自己搞定了。说几个具体的吧，  通过这次的课程设计,主要得到了这么几个经验。首先,在开发之前，如果是比较生疏的领域最好将资料准备完备，否则在做扩展的时候将会后悔莫及;其次，效率永远是第一位的，不要勉强自己疲劳工作;最后, 询问别人并自思考永远是最快捷的成长途径。总而言之,通过这次的软件开发,通过这门课程,我们培养了自己独立思考和独立作业的能力，更加能够自信地面对即将踏入的社会。