



中国海洋大学  
OCEAN UNIVERSITY OF CHINA

# 硕士学位论文

MASTER DISSERTATION

基于 S5PV210 开发板的 Android

论文题目: 系统移植与应用开发

英文题目: Android system porting and application development  
based on the S5PV210 development board

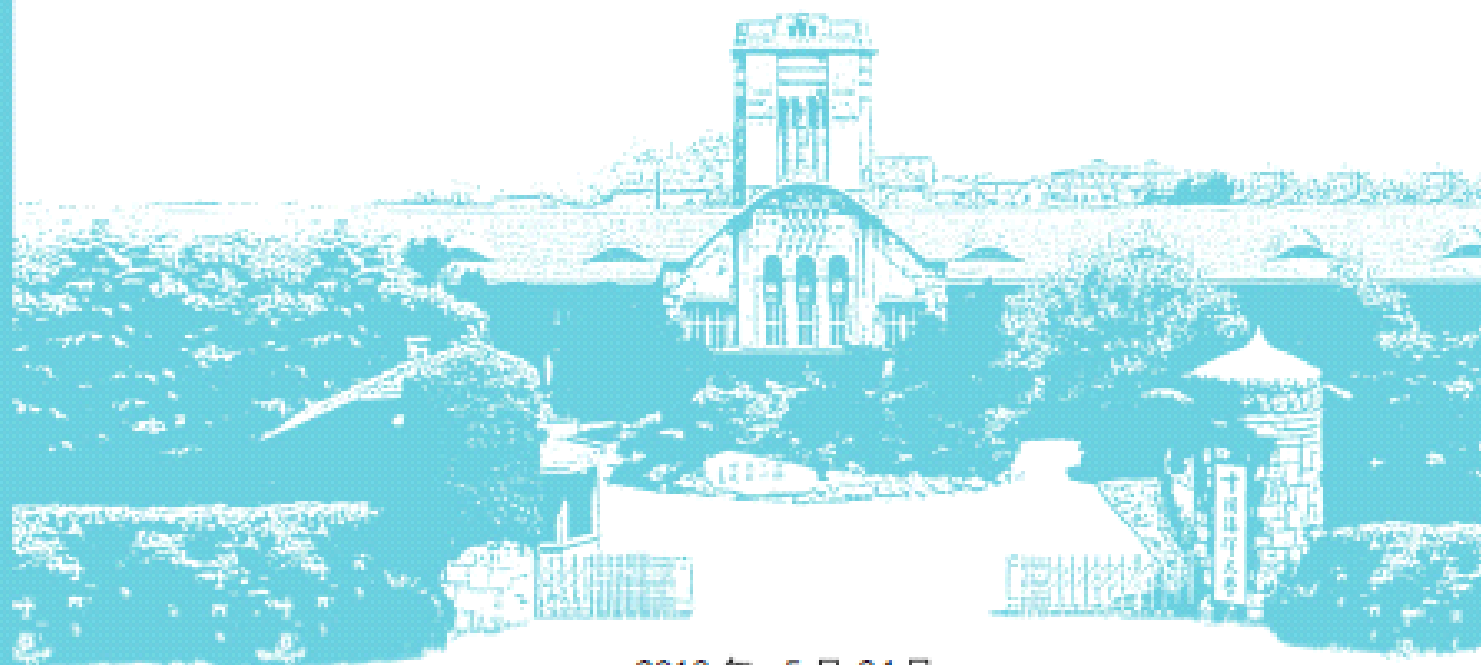
作者: 刘岩

指导教师: 王国宇 教授

学位类别: 全日制专业学位

专业名称: 电子与通信工程

研究方向: 智能检测



2013 年 5 月 24 日

谨以此论文献给所有关心我的人

----- 刘岩



基于 S5PV210 开发板的 Android 系统移植与应用开发

学位论文答辩日期: \_\_\_\_\_

指导教师签字: \_\_\_\_\_

答辩委员会成员签字: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_



## 独 创 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的  
研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其  
他人已经发表或撰写过的研究成果，也不包含未获得注：如没有其他需要特别声明的，本栏可空或其他教育机构的学位或证书使用过的材料。与我一同工作的  
同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：                    签字日期：      年    月    日

-----

## 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，有权保留并  
向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人  
授权学校可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用  
影印、缩印或扫描等复制手段保存、汇编学位论文。同时授权中国科学技术信息  
研究所将本学位论文收录到《中国学位论文全文数据库》，并通过网络向社会公  
众提供信息服务。（保密的学位论文在解密后适用本授权书）

学位论文作者签名：  导师签字：

签字日期：      年    月    日                            签字日期：      年    月    日



# 基于 S5PV210 开发板的 Android 系统移植与应用开发

## 摘 要

自从 1973 年,“手机之父”摩托罗拉公司工程技术人员马丁·库帕,发明了世界上第一部民用手机,手机逐渐成为人们日常生活中不可缺少的通讯工具。现在的手机在保证最基本的通讯功能下,还集其他各种功能于一身,全面发挥贴近人们日常生活的各种功能。功能全面、特色鲜明的手机必须有一套性能优越的系统作为强有力的支持,因此对手机系统的研究变得越来越重要。当前世界上的几大手机操作系统包括黑莓操作系统、Windows Mobile 操作系统、iPhone 操作系统、塞班操作系统、Android 操作系统。其中作为代码开源的 Android 操作系统,凭借其优越的系统性能、友好的用户界面以及打破专有权阻碍的开放性,都使 Android 系统这款新兴的手机系统在众多手机系统中占有一席之地并且其重要性变得越来越重要。HTC、三星、摩托罗拉等移动通讯行业的世界巨头都已经推出自己的 Android 手机产品,并且如今的 Android 系统不再仅仅局限于应用在手机上,很多公司已经把 Android 系统应用到平板电脑、手表、甚至是仪表设备上。

本文首先介绍了手机的发展历程,简单介绍了当前移动通讯几款最具特色的智能手机系统,从智能手机给人们生活带来方便的角度阐述了智能手机的优点。然后分析了 Android 系统成为几大智能手机系统之一不同于其他手机系统的优点,以及当前国内外对 Android 系统的研究和发展现状。

其次分析了 Android 系统架构和 Android 系统的组成,并对 Android 系统的四大组件及其生命周期进行详细的阐述。

然后针对本文的开发板硬件资源做了基本了解,阐述了如何把 Android 系统移植到 S5PV210 开发板上并运行成功,其中的工作包括,制作引导程序,在 Google 发布的 Android 内核进行适当的修改和裁剪,使之适用于 S5PV210 开发板,安装交叉编译器,配置编译内核制作成烧写使用的镜像,还有制作 Android 源码镜像。

第四,在 Windows 环境下,搭建 Android 应用程序开发环境,在详细研究 Android 开发程序的开发过程和组件的使用方法后,利用 Eclipse 工具,构建出 mp3 音乐播放器,并把应用程序烧写到移植后的开发板上,实现音乐播放的功能,间接验证了移植成果。

最后对本文做出总结并展望未来工作。



关键词：Android；S5PV210；移植；音乐播放器

# Android system porting and application development based on the S5PV210 development board

## Abstract

Since 1973, the father of the "mobile phone" Motorola Engineering Technician Martin Cooper invented the world's first civilian mobile phones, mobile has become indispensable communication tools in people's daily life. The cell phone has become an indispensable part of the daily life of contemporary people due to the fundamental telecommunication and other various services that it renders. Full-featured, distinct features of the phone must be supported by a set of superior performance of the phone system, so the research of the phone system is becoming increasingly important. Now several major mobile phone OS in the world includes the BlackBerry OS, Windows Mobile OS, iPhone OS, Symbian OS, Android OS. As a code open source OS, Android OS, which by virtue of its superior system performance, user-friendly interface, and the breaking down of the exclusive rights of OS, is becoming more and more important in a number of the Mobile phone OS. The world's giants of the mobile communications industry for example HTC, Samsung and Motorola, have released its own Android phone products. Toady, Android OS is no longer confined to be applied on the phone, many companied have applied Android system to Table PC, watches, and even the instrumental equipment.

This paper first introduces the development of mobile phones, briefly introduced several of the most unique phone systems, from the convenience that smart phone bring to people's lives perspective to explain the advantages of smart phone. And then the paper analyzed the Android's advantage which is different from the other smart phone systems, and the current research and development status at home and abroad on the Android system.

Second, the article analyzes the Android system architecture and the composition of the Android system, and elaborates on the four major components of the Android system and its life cycle.

Then the article describes the development board's hardware resources, explains how to port the Android system on S5PV210 development board and run successfully,

which includes making boot loader, modifying and tailoring Google's Android kernel, installing the cross compiler, compile the kernel and make it into a mirror, and making Android source mirror.

The article explains how to build Android application development environment in the Windows environment. And after the detail study of the use of the development progress and components of android developer program, I build a mp3 music player program by the use of Eclipse, and it realizes music player function on the development board which indirectly confirms the porting outcomes.

Finally, the article summarizes the work that has been done, and discusses the follow-up works.

**Keywords:** Android; S5PV210; porting; music player

# 目 录

1 绪论.....	1
1.1 课题研究背景.....	1
1.2 智能手机操作系统.....	2
1.3 智能手机的优点.....	3
1.4 Android 操作系统简介.....	4
1.5 国内外研究现状.....	5
1.6 本文内容和文章结构.....	6
1.7 本章小结.....	7
2 Android OS 简介.....	8
2.1 Android 操作系统的特点.....	8
2.2 Android 系统框架结构.....	9
2.2.1 Android 应用程序层.....	10
2.2.2 应用程序框架层.....	10
2.2.3 系统运行库层.....	11
2.2.4 Linux 核心层.....	12
2.3 Android 四大组件.....	12
2.3.1 Activity.....	12
2.3.2 Broadcast Receiver.....	14
2.3.3 Service.....	15
2.3.4 Content Provider.....	15
2.4 Android 系统启动过程.....	16
2.5 本章小结.....	16
3 Android 系统移植.....	18
3.1 S5PV210 开发板介绍.....	19
3.2 制作引导程序.....	20
3.3 Android 内核移植.....	21
3.3.1 Android 内核与 Linux 内核的区别.....	22
3.3.2 制作 Android 内核.....	24
3.4 制作 Android 源码镜像.....	28
3.4.1 获取 Android 源码.....	28
3.4.2 制作 Android 文件系统镜像.....	30
3.5 移植成果.....	32
3.6 本章小结.....	32
4 MP3 播放器的设计与实现.....	33
4.1 Android 开发环境搭建.....	33
4.2 Android 应用程序文件目录.....	34
4.3 MP3 播放器的设计与实现.....	34
4.3.1 需求分析.....	36
4.3.2 界面设计.....	36

4.3.3 功能实现.....	38
4.4 成果展示.....	43
4.5 本章小结.....	45
5 总结与展望.....	46
参考文献.....	49
致 谢.....	51
个人简历.....	52

# 1 绪论

## 1.1 课题研究背景

1902 年第一部手机诞生,随着人们对信息交流的需要不断增加,手机凭借着可以打破时间地域束缚的优势,慢慢被人们所接受。如今世界微电子技术日新月异,手机通讯业也随之发生了翻天覆地的变化,由最开始只有政府、上层社会才有能力使用的手机如今几乎人人必备,手机的方便性也由原来的大块头逐渐被现在小巧、携带方便的现代手机所代替,功能也从最开始的打电话、发短信逐步演变成人们随身携带的休闲娱乐工具。随着社会发展和技术进步,手机慢慢被人们所接受,人类进入手机时代。中国手机发展阶段基本可以分为 4 个时期:模拟手机时代、2G 时代、2.5G 时代和 3G 时代。其中 2.5G 和 3G 代表着中国手机的发展。

模拟手机把需要传送的声音转化为低频波加载进高频波中进行传输,然后在接收端过滤掉高频电波,人们就可以接受到来自远处的模拟的声音低频波。在模拟手机时代摩托罗拉曾经称霸整个手机行业,世界上第一部民用手机是 Motorola 8000X,如图 1-1。模拟手机有其致命的缺点:抗干扰能力差<sup>[1]</sup>,为了解决这个问题数字移动通讯技术应运而生,即 GSM 时代来临。在 2.5G 技术中,GPRS 起到了从 2G 时代到 3G 时代的过渡作用,其优越性不言而喻。GPRS 网络采用手机通话继续使用 GSM,而数据的传送则使用 GPRS。当仅仅具有通信功能的手机已经不能再达到市场人们对手机的要求时,3G 时代应运而生,这时的手机不仅仅是通信工具,更是集众多其他功能于一身,可以看电影、浏览网页、看电视、打游戏、当作车载、实时监控器等。2G 和 3G 的差别简单来说,类似于小马路和高速公路的差别。在 2G 时代,每当遇上打电话发短信的高峰期,经常会出现短信接收不成够或是电话业务繁忙这类的问题;而现在,手机可以完成的不仅仅是打电话、发短信,还可以传输文件、资料<sup>[2]</sup>。用户也可以随时随地用手机记录自己生活的点点滴滴,之后通过网络传输到自己的网站上,让亲友、同学一睹自己的风采。网络资料传输速度变快,随时随地地连接互联网已经不是天方夜谭。



图 1-1

在当今 2.5G 和 3G 交接的时代，简单的电话短信服务已不能满足人们对手机功能的要求，并且随着 3G 时代的来临，使用手机进行网络的数据交换已经成为快速方便的事情，把手机作为缩小版的个人电脑毫无疑问是势在必行而且目前来看这种思想是很成功的，因此，“智能手机”这个关键词正变得越来越重要。

## 1.2 智能手机操作系统

智能手机正常运行需要操作系统强有力的支持。一套好的手机系统是发挥手机自身优势的基本保证，只有好的操作系统才能充分支持应用软件的需求。世界各大手机厂商积极开发自己的手机系统，让用户有了多种选择。本文将对 Android 操作系统做详细论述，在此对目前其他主要智能手机操作系统进行简要的介绍。

塞班操作系统，该系统曾一度成为手机系统的领导者是因为移动通信业的领军人物 Nokia 专门采用该系统为自己的智能手机创造市场，其他设备厂商，如摩托罗拉和三星也都在塞班普及初期开发有塞班系统的智能手机。塞班智能操作系统基于微内核架构，支持抢占式多任务，其架构原理和机制在早期系统中是很出色的。该系统允许每个需求者定义适合自己风格的界面，就是说倘若应用程序运行在不同厂商的硬件上，应用程序供应商将不得不重新编写 UI 部分。塞班智能操作系统的热门软件平台包括诺基亚使用的 S60、摩托罗拉使用的 UIQ 等等。塞

班的主要发展环境是是 C++的一种变异体并以 C++可以被用来访问塞班系统服务的类库为基础的。

iPhone 操作系统，是 Mac OS 的衍生产物，苹果公司为自己公司的移动设备量身制定、设计开发的操作系统。它分为四个技术层次 - 核心操作系统，核心服务，媒体和可轻触层。其内核是基于 Mach 内核的一个变种。iPhone OS 自带各种预装的应用程序（例如：电子邮件，日历，iTunes 的应用程序商店，地图，YouTube 等）。iPhone OS 也支持通过 Safari 浏览器的 Web 应用程序交付。由于 iOS 使用目标 C 作为其编程语言，所以它也带有经典 C 编程语言的漏洞，比如整数易溢出和字符串易错。

Windows Mobile，它是微软为其移动设计和开发的操作系统，无论掌上电脑、袖珍电脑还是智能手机，都采用该智能操作系统。Windows 手机在使用和功能上的设计与桌面 Windows 操作系统相似。Windows Mobile 被大多数硬件制造商所接受，除了诺基亚、RIM、苹果这些厂商例外。该操作系统设计为四层结构，包括硬件层、OEM 抽象层、内核层和用户应用层，其中 OEM 抽象层决定了 Windows Mobile 和桌面 Windows 之间的差异性。第三方开发人员可以使用 VC++ 为 Windows Mobile 平台编写应用程序。与桌面版本一样，微软在 Visual Studio 开发环境中提供了软件开发工具包(SDK)。

黑莓操作系统，由加拿大的 Re-search in Motion(RIM)公司开发和设计，第一部黑莓智能手机于 2002 年发布。在此之前，1999 年 RIM 公司推出了寻呼机和可以连接电子邮件的 PDF。黑莓智能手机设计有一个完整的键盘、管理功能和 e-mail 访问，这使得它在商业人士中得到广泛青睐。黑莓智能手机操作系统是建立在 Java 和 J2ME 语言之上的，其安全体系结构是以签名和等级限制之间的结合为基础。

### 1.3 智能手机的优点

智能手机和以往的非智能手机不同，它拥有更优秀的操作系统、支持更多的硬件设备、更注重用户和机器交互的友好性。对普通手机稍有接触的人都很清楚普通手机只可以进行单线程操作，当打开一个应用软件再想使用其他软件就必须把当前的应用程序关闭掉，否则很可能连界面都不能切换。智能手机却可以向电



脑一样进行多线程任务，简单的来说，用户可以打开音乐播放器听着歌曲然后再开启看书软件看书，做到听音乐看书同时进行，用户的主观体验上了一个很大的档次。

作为可以比拟微型电脑的手机系统，其对用户的自由度根本不是以前封闭手机系统所能比拟的。用户可以通过任何途径获得支持自己手机的应用程序，安装使用，使用完后，也可以卸载掉应用程序。这使得手机对用户而言由原来的封闭空间成为一个开放性、包容性、自由性的空间，极大地丰富了用户对应用软件的选择性，制作或者下载到适合自身的应用软件，得到需要的服务，体现了电子行业的人性化发展趋势。

智能手机的功能与普通手机相比可以用天差地别进行比喻，普通手机由于其系统的封闭性，除了简单的打电话发短信功能以外，基本是没有其他功能可言，但是可以毫不夸张的说智能手机功能的强大已经使其成为人们不可缺少的左膀右臂。手机地图为人们搭乘公交、徒步旅行提供了莫大的帮助。办公软件让众多白领脱离了只能坐在电脑面前工作的烦恼。天气软件随时为人们的日常出行提供健康保障。众多精彩刺激的游戏软件让青少年游戏一族爱不释手。购物系统使得人们随时随地都可以浏览购买自己喜欢的物品。聊天软件不但让人们脱离了时间地点的束缚而且只需要消耗少量的流量即可任意沟通，比打电话打短信更加经济实惠。用智能手机上新浪、网易新闻随时随地关注国家大事<sup>[3]</sup>。

随着智能手机的出现，触摸屏的应用也随之成为手机屏幕的主流。去除众多手机按键后的智能手机，保证了在体积上不比普通手机大的前提下，手机的外观变得更加整洁，同时，让用户得到一个更宽的手机屏幕，获得更精彩的视听盛宴，也真是因为这个原因，智能手机得到广大青少年的追捧。

## 1.4 Android 操作系统简介

Android 是由 Google 为首的 OHA(Open Handset Alliance)推出的一款开放的嵌入式操作系统平台，从 2008 推出的 android SDK1.0 到现在，Android SDK 的最新版本已经升级到了最新的 4.0.3，每当 SDK 更新一次版本都会吸引一大批开发者的注意力并慢慢向 Android 系统开发上发展，这已经成为一个势不可挡的趋势，作为因果循环，Android 系统也得到广大开发者的滋养，引用的数量和质量

得到一个质的变化。目前 Android 开发最多的是在手机市场上，但 Android 系统作为一款全新的开放式系统，已经开始被移植到其他电子设备上，目前 Android 系统在电脑、电视上的应用势头正日益渐盛，这一切迹象都表明 Android 系统蕴含着无限的潜力，因此做好 Android 系统的嵌入式移植和上层开发，使 Android 系统能够被更广泛的使用，有着积极而深远的意义。世界上第一款 Google Android 系统的手机是 HTC G1，如图 1-2。



图 1-2

如今通过 Android 系统移植到电视机上，打破了传统电视机单一交互的方式，利用 Android 开发平台开发出的运行于电视的多媒体播放软件，使得用户可以从不同平台设备获得更加全面符合用户需求的信息，从而为电子终端设备更加深入人类生活和生产的各个领域起了强有力的推进作用。

## 1.5 国内外研究现状

自 2007 年 11 月 5 日 Google 公司公布开源手机系统平台——Android 系统之后，凭借其强有力的硬件兼容性和源码开放性，已成功征服众多世界顶尖的 IT 公司，像 ARM、ASUS、联想、索尼爱立信、三星、HTC 等，并由 Google 公司

宣布联合成立开放手机联盟<sup>[4]</sup>。

虽然目前 Android 系统移植在 IT 行业进行的如火如荼,但由于系统本身的复杂性、硬件厂商对移植过程的保密性、移植平台的多样性等等原因,使得 Android 移植方面的资料虽然众多,但在初学者学习过程中很大一部分都需要在网络上寻找需要的答案和解决办法,针对移植的书籍与上层开发相比较少之又少,这无疑是对国内 Android 移植学习者的一大打击。

国内外大厂家都已成功把 Android 系统移植到自己的产品上,IT 行业巨头不但早已推出自己的移植产品并且得到了用户的认可。

## 1.6 本文内容和文章结构

本文的工作重点可以归为两点,首先是通过在网络和书籍上的学习,掌握 Android 系统移植的流程和重点,根据 S5PV210 开发板的硬件设施进行适当的移植工作,解决移植过程中遇到的各种问题,把 Android 系统移植到 S5PV210 开发板上,使 Android 系统成功运行在这个开发板上。其次,根据自身掌握的 Android 上层开发知识,自主编写一款音乐播放器软件,调试其中的 BUG,通过编译链接产生 apk 文件,最终使音乐播放器的 apk 应用程序在 S5PV210 开发板上成功运行实现相应的功能。

以下几点是本文在研究和撰写过程中最关键的几个问题:

- 1、掌握 Android 移植的大体流程。
- 2、针对 S5PV210 开发板,对内核进行相应的修改,形成能运行在这个开发板上的内核。
- 3、获得 Android 源码并把源码制作成可烧写的文件系统镜像。
- 4、对上层开发进行研究,设计出一个完整的 Android 工程,编写源码,实现音乐播放器的功能,使其最终在 S5PV210 开发板上成功运行。

本文章节结构如下:

第一章讲述了手机的发展历程,以此作为切入点,然后简要了解了塞班、iPhone、Windows Mobile、黑莓等当前主流智能手机操作系统,并从用户的角度论述了智能手机的优点。最后对 Android 系统进行全面分析并陈述了当前国内外的研究现状。

第二章详细讲述了 Android 系统的特性、Android 系统的框架结构、四大组件以及系统启动流程。

第三章讲述了 Bootloader 的作用，Android 内核与 linux 内核的区别以及如何对 U-boot 和内核进行适当的修改以适用于 S5PV210 开发板。

第四章介绍了 Eclipse 工具以及如何使用 Eclipse 进行音乐播放器的设计和功能实现的具体步骤，并把应用程序安装到开发板上，实现相应功能。

本文的最后对所做的工作做了总结，通过分析本论文的不足之处，对下一步工作做出展望。

## 1.7 本章小结

本章首先对手机的发展历史做了简要了解。论述了智能手机操作系统的特点以及给人们日常生活带来的帮助，然后介绍了 Android 系统以及国内外对 Android 系统的研究现状，最后对本文各章节做了大体的总结。

## 2 Android OS 简介

Google 公司认识到当前手机市场的发展方向，其构建的 Android 系统的得到了很好的发展。Android 操作系统是站在 linux 操作系统这个巨人的肩膀上的，并且 Android 源码是开放性的，这就使得很多之前热衷于 Symbian 操作系统的商家在把目光投向了异军突起的 Android 操作系统，从此手机市场发生了翻天覆地的巨变，从之前的 Symbian 一手遮天转瞬形成了 Android、IOS、Symbian 三足鼎立的境况。

Android 之所以发展的如此顺利，其中一个很重要的原因是其使用了 Java 语言进行开发，Java 语言的语法风格和 C\C++ 很相似，并且 Java、C、C++ 这三大软件语言，其背后的研发力量不可估计，于是大量的 Java、C、C++ 开发者把目光投向了 Android 的开发，从此保证了对 Android 系统强大的研究力量。

### 2.1 Android 操作系统的特点

Android 操作系统的特点主要体现在五个方面：开放性、平等性、无界性、方便性、硬件的丰富性。

开放性：Android 在最初设计的时候就定位于标准化开放性的操作平台，它是建立在 linux 开放平台的基础上<sup>[5]</sup>。

平等性：在 android 操作系统上，不论是系统自带的应用程序还是用户自行开发的应用程序，用户都可以根据自己的喜好自由选择，而不是只能使用一家独大的系统程序。比如用户不喜欢 android 系统自带的音乐播放器，允许用户自行开发一款属于自己特色的音乐播放器或者使用购买的音乐播放器软件，比如当前盛行的酷狗音乐盒等。

无界性：指的是其应用程序相互间能够进行相互组合嵌套。用户自行开发的多个程序能够比较容易得进行程序之间的交互。比如，某个游戏程序需要视频播放功能，恰巧你的手里有视频播放的模块，这样只需要进行简单适当的配置，就可以把视频播放模块嵌套进游戏程序里，而无需再进行多余的视频播放器模块的开发。另外，多个用户进程之间能够相互访问，比如 android 系统中的 ContentProvider，就是专门负责 android 系统内部数据的存储与共享。

方便性：这里 android 平台的方便性特指应用程序的开发，android 平台集合了大量的实用数据库和方便的工具集，而且集合了很多实用的强大功能，比如 Google Map 等等。对于 android 的开发人员，如果对 android 平台比较熟悉，只需要用很简单的调用代码就可以把这些强大的功能整合到一起制作成一个模块嵌套到需要的应用程序中。

硬件的丰富性：由于 Android 平台的开放性，很自然的吸引了众多厂商的注意力。各家厂商推出特色各异、功能不同的多种产品。但是功能与特色上的差别并没有影响到数据同步、软件兼容。比如原来在 Nokia 手机上的应用程序能够很轻松的移植到三星手机上使用，并且保证其中的文件资料更加方便安全的转移。

## 2.2 Android 系统框架结构

Android 系统框架结构如图 2-1 所示。



图 2-1 Android 系统框架结构

Android 系统框架采用分层结构，共有四层，从低层至高层依次是应用程序层、应用程序框架层、系统运行库层、Linux 核心层<sup>[6]</sup>。Google 公司在开发 android 系统时，采用分层框架结构，有显著特点。首先，明确各层分工，使 android 的

系统开发、移植开发、应用程序开发所关注的重点变得一目了然。其次，减小各层之间的依赖性、从而提高系统的稳定性并减轻改版系统所付出的代价，比如写文章，条理清晰确的文章，其修改起来会轻松不少并且可以提高整个系统的可移植性。

### 2.2.1 Android 应用程序层

应用程序层是由 Java 语言编写的可以运行在虚拟机上的程序集合，图 2-1 最顶层的蓝色部分代表的就是应用程序层。在开发 android 系统时，Google 公司就已经在 android 系统中捆绑了若干比较重要的应用程序，比如文件管理器、SIM 卡应用、任务管理器、流量统计器等。

Android 系统是一个开放的系统。Android 系统开发分为三种开发：应用程序开发、移植开发和系统开发。其中的应用程序开发的工作就是进行在应用程序层这个平台之上的，开发者开发出各种各样的适合市场、用户需求的应用程序，然后将这些应用程序投入 Android 商店进行交易，方便用户购买使用。

### 2.2.2 应用程序框架层

应用程序框架层是建立在核心层和系统运行库之上的，在图中所处的位置是在应用程序层之下的蓝色部分。应用程序框架层集合了应用层所用到的最重要的基类，基类为应用程序层提供 API 接口，众多的 API 接口的集合形成了 API 框架，如图 2-2，开发者通过应用程序框架层提供的 API 框架来实现所编写的应用程序的功能，因此就简化了程序开发的架构设计，但是必须遵守 API 框架的安全限制条件。

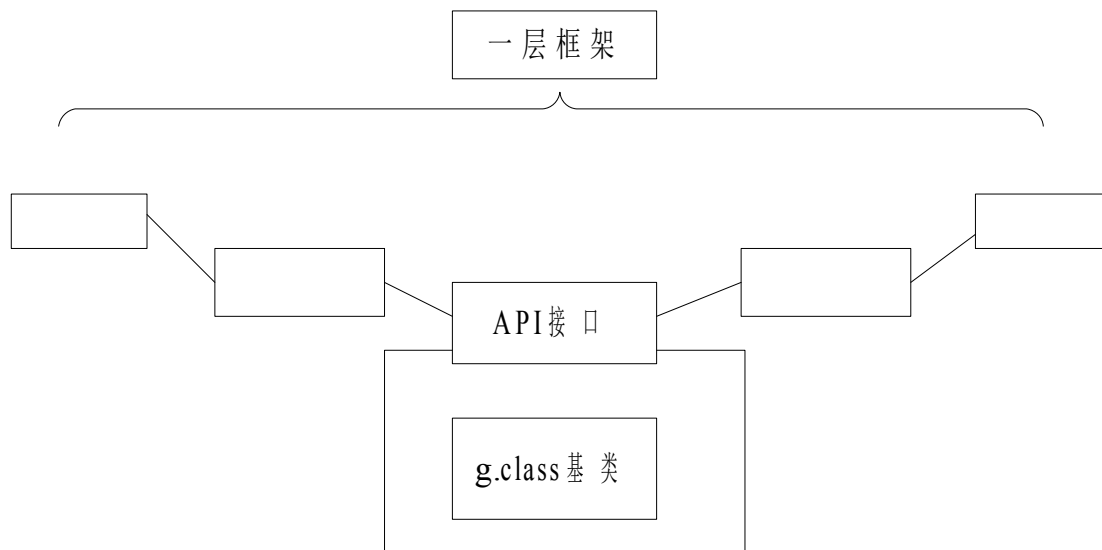


图 2-2 API 与框架关系

应用程序框架层为应用程序开发者提供的几种组件：视图系统、内容提供者、资源管理器、包管理器、活动管理器和硬件服务<sup>[7]</sup>。这些组件包含了应用程序运行时需要的所有基本功能支持，可以说这些组件是组成应用程序的基本单位。

### 2.2.3 系统运行库层

Android 的系统运行库分为 Libraries 运行库层和 Android Runtime 层。其中 Libraries 运行库层对应于图 2-1 的绿色部分，Android Runtime 层对应于图 2-1 黄色部分。

系统运行库层是由 C/C++ 语言编写的，包含了 Android 系统平台底层的不同功能的组件库和 android 系统的运行时环境。Google 公司在开发 android 系统时为什么要把系统运行库划分成一个独立的层，而不是把它归并到上一层呢？其中原因有两条：

首先，应用程序框架层是由 Java 语言编写的，Java 源码执行速度慢，C 程序源码执行快，这就注定要把 android 系统中核心的部分用 C/C++ 来编写。

更重要的一点是，许多既有的程序模组都是由 C/C++ 编写，并且通过系统运行库和应用程序框架层的 JNI，能轻易将 Java 程序与 C 模组衔接起来。

Libraries 运行库是由 C/C++ 编写的各种函数库，应用程序实现框架层的 API 接口，API 接口所处的基类会通过 JNI 层调用 Libraries 运行库的函数。所以，上层的应用程序其实是最终调用众多的 Libraries 运行库的核心库函数来支持应用



程序的运行。

Libraries 运行库包含各种功能的组件，例如系统 C 库，它是由标准 C 系统库衍生而来的。媒体库，可以对主流的音视频格式进行播放，比如 MPEG4、MP3 格式，还有图像文件，比如 JPG、PNG 格式等等。用于矢量字体和位图渲染的 FreeType。轻型的数据库 SQLite，遵守 ACID 的关联式数据库管理系统，占用资源非常的低，仅需要几百 K 的内存即可<sup>[8]</sup>。

Android Runtime 层主要是 Dalvik 虚拟机，它是 Android Java 应用程序运行的基础，在虚拟机上进行程序的调试极大地缩短了调试占用的时间。该虚拟机有对内存高效使用，以及在低速 CPU 上表现出色的优点。Android 系统可以简单地完成进程隔离和线程管理。任何一个 android 应用程序都对应一个独立的 Dalvik 虚拟机实例，并且 Dalvik 虚拟机可以同时运行多个 Android 应用程序实例<sup>[9]</sup>。

#### 2.2.4 Linux 核心层

Android 的 Linux 内核部分可以看成处在硬件层和系统软件组之间的一个抽象层。这个抽象层定义了各种硬件设施的驱动程序，比如，Display Driver（显示器驱动程序）、Camera Driver（照相机驱动程序）、Flash Memory Driver（闪存驱动程序）、Wifi Driver（Wifi 驱动程序）。但是严格来说，它并不算是 Linux 操作系统。

### 2.3 Android 四大组件

开发者对 Android 系统的开发大部分是集中在应用程序的开发上，因此对大部分开发者而言 Android 系统的主体部分就是应用程序框架。Google 在设计 Android 系统时，为使应用程序开发变得简单易行，吸引更多开发者加入其中，在应用程序框架中设置了四个重要的组件。它们分别是 Activity（活动）、Service（服务）、Broadcast Receiver（广播接收器）、Content Providers（内容提供者），四大组件形成了整个 Android 框架的灵魂<sup>[10]</sup>。

#### 2.3.1 Activity

Activity 处理 Android 应用程序 UI 互动的事情，是构建应用程序界面的组件，

其中包含许多的控件，可以说，Activity 就是 Android 应用程序的脸面，毫无疑问的，Activity 是四大组件的重中之重。它不仅需要显示当前窗口状态，还要对外部的触发事件做响应，并且要实现 Activity、Service 等之间的跳转。一般来说，开发者使用 Activity 时必须继承 Activity 基类，才能编写其他代码实现应用程序需要的功能。

Activity 的生命周期，可以用图 2-3 来概括：

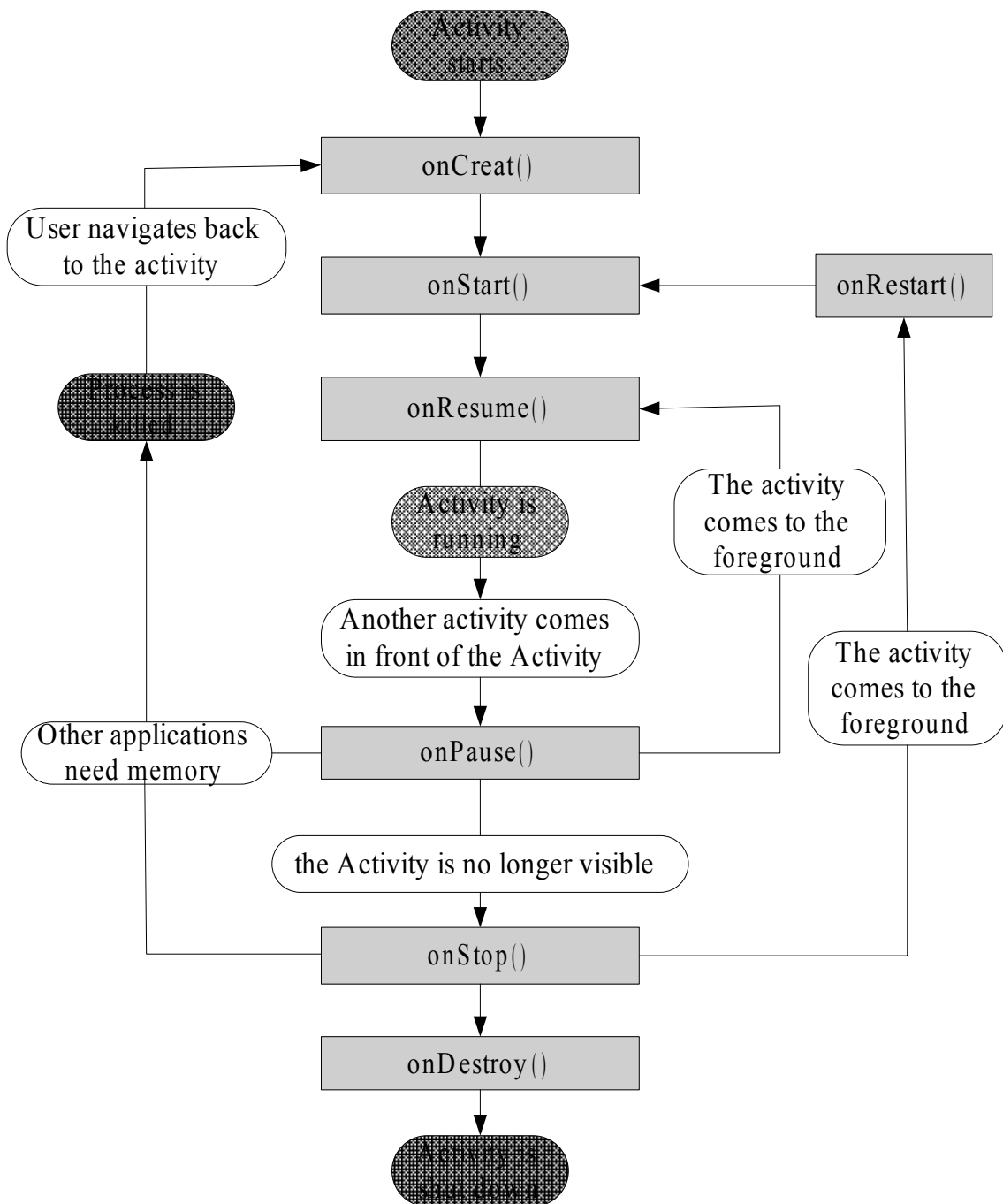


图 2-3 Activity 生命周期

图 2-3 是 Google 官方的 Activity 生命周期。Activity 的生命周期函数包含有 onCreate()、onStart()、onResume()、onPause()、onStop()、onDestroy()、onRestart() 七个函数<sup>[11]</sup>。和其他大部分组件一样，都是从创建生成到销毁的过程，只不过 Activity 和其他组件相比，生命周期得到细化，以此也可以看出 Activity 在众多组件中凸显的复杂性和重要性。

onCreat(), 用来创建 Activity。

onStart()方法，当 Activity 开始被执行时调用此方法，一般情况下，该方法紧随在 onCreate()函数之后。

onPause()方法，从一个应用界面调用另外一个用用界面时，前一个 activity 就会调用该函数，进入暂停状态并被新调用的界面在屏幕上替换掉。当调用了 onPause()函数的 activity 因为 android 系统的内存管理机制，它的进程被杀死。当在重新被调用运行，该 Activity 就会重新被创建，也就是重新调用它的 onCreate()方法。

onResume()方法，调用了 onPause()方法的 activity，如果它的进程没有被杀死，该 activity 再次被调用，重新获得用户焦点时，会直接调用它的 onResume()函数运行该 activity。所谓的用户焦点，可以这样理解，程序的界面有时会弹出一个窗口或者其他程序，此时其背后的应用界面虽然可以被看到却不能接受用户的输入指令，即没有获得用户焦点<sup>[12]</sup>。

onStop()方法，当 activity 很长时间没有被再次运行，即对用户很久不可见，会调用该方法，activity 被停止。

onRestar()方法，调用 onStop()函数的 activity 再次被运行时，调用该方法。

onDestroy()方法，可以彻底销毁 activity。

### 2.3.2 Broadcast Receiver

在 Android 应用程序中，通常会应用到一些触发机制，当 Broadcast Receiver 能够对外部事件进行滤掉，只对感兴趣的外部事件（例如接收到短信或者有电话接入时）进行接受并做出相应的处理<sup>[13]</sup>。广播接收器是在程序后台运行的，用户是无法观察到的。它对感兴趣的外部事件做出的反应，视情况而定，可以是开启一个 activity 或者 service 来响应其接收到的感兴趣事件并且通知用户，吸引用户

注意力的方法有多种，比如 LCD 闪烁、设备振动或者播放音乐等等<sup>[14]</sup>。

Broadcast Receiver 在应用程序中的注册方法有两种：

第一、通过使用 Activity.this.sendBroadcast(intent) 方法，在 AndroidManifest.xml 当中对 Broadcast Receiver 进行注册。该方法的特点是，不论程序是否打开，监听一直是进行着的，其缺点也是显而易见的，在程序运行的时间内，广播接收器一直在占用内存和消耗设备电量。

第二、通过使用 registerReceiver(receiver,filter)方法，在应用程序的代码中对其进行注册，该方法的特点是，当应用程序关闭时，监听也随之停止，省电省内存。

### 2.3.3 Service

Service 可以应用到多种场合，它没有可视化界面，为整个 Android 应用程序提供后台服务。比如，下载资源时，使用者只关心下载是否完成，下载过程中更希望去做别的事情，因此在下载资源时 Activity 会调用 context.startService()方法建立一个后台服务，利用 Service 进行后台服务，这样不会影响到用户的正常使用，并且系统会保持 Service 一直在运行，直到下载任务完成。通过 startService()方法建立的 Service，建立完成后它就和开启它的 Activity 脱离关系了。不论 Activity 是否继续运行，服务都可以继续进行，直到完成当前任务。如果要提前停止该服务需要调用 context.stopService()方法，这样系统会调用 onDestroy()杀掉该服务。

和 Activity 一样，Service 在使用时也要继承其基类。Service 是在后台运行的服务，因此它和用户之间没有交互的应用组件。和 Broadcast Receiver 相似，每个 Service 要在 AndroidManifest.xml 配置清单文件中<application>节点里对其进行配置，例如<service name=".SMSService"/>。

### 2.3.4 Content Provider

内容提供者实现了应用程序之间数据共享<sup>[15]</sup>。利用内容提供者得到特定数据的应用不能直接从内容提供者取得希望的数据，而是经过内容分解器这个类间接地从内容提供者中获取或者存入数据。内容提供者只有在多个应用程序共享同一

份数据时，才会被用到<sup>[16]</sup>。

四大组件之间是如何相互沟通，协调工作的呢？这就不得不提到 Intent（意图）。Intent 本身是定义为一个类别，Intent 中包含的最重要的两个参数是 Component Name（组件名称）和 Action（动作）。前者代表意图所指向的组件，后者是意图期望所指向的组件发生的动作。Intent 物件表达一个目的或期望，叙述其所希望的服务、动作、与动作有关的资料等，Android 则根据 Intent 的描述，负责配对，找出配对的组件，把意图传递过去<sup>[17]</sup>。

## 2.4 Android 系统启动过程

开机上电，bootloader 进行底层初始化，加载 Linux kernel 部分，随后跳转到内核的 boot 程序，进行 Kernel-Space 的初始化，然后载入硬件驱动，至此启动了 Linux 系统<sup>[18~19]</sup>。随后切换到 User-Space 的初始化，创建 init 初始化进程，读取 init.rc 文档，在初始化完成之后 init 进程仍会存在。然后启动系统底层的核心服务，这些服务不同于应用程序中的服务，而是支持系统运行的服务。首先启动本地服务再启动 Android Service，完成核心服务的启动<sup>[20]</sup>。本地服务是系统运行库层提供的 C/C++编写的服务，Android Service 是应用框架层提供的 Java 编写的服务。

启动 Native Service 和 Android Service 时，首先 Core Service 逐一登记到 Binder Driver，随时准备为 app 调用。其中 ServiceManager(SM)会优先登记到 BD 里，方便其他核心服务透过 IBinder 接口远距离呼叫 SM，进而更多核心服务呼叫 SM 的 addService() 函数去登录到 BD 里，例如 ActivityManagerService、MediaPlayerService。系统在启动服务完成后，会通知各个服务系统已经完成服务的启动，同时调用 ActivityManagerService.systemReady() 启动 HomeActivity，Android 系统启动完成<sup>[21]</sup>。

## 2.5 本章小结

本章主要介绍了 Android 的基本特性：开放性、平等性、无界性、方便性、硬件的多样性，从架构方面分析了 Android 框架各层的作用，从宏观上对 Android 有了一个基本的认识。重点介绍了 Android 的四大组件以及它们的功能和之间的

相互联系。整个系统的功能是通过从高层到底层的调用实现的，应用程序实现其底层的 API 接口；应用框架层功能的实现需要调用 Libraries 库函数；Linux 核心层为 Libraries 提供底层硬件的请求数据。

### 3 Android 系统移植

自从 google 公司发布 Android 系统以来，以 Android 系统的开放性和易移植的特性，吸引了众多个人、小团体以及大公司的注意力，他们纷纷把 Android 系统移植到不同的硬件平台，甚至把用于 Android 系统的 apk 移植到其他系统上去至，如今已取得不错的成果。

Android 系统移植的途径大致分成两种，第一种是先制作出针对硬件设备完整的 linux 系统，然后再通过网络等手段得到 Android 系统的补丁，把这个补丁打到 linux 系统上，让它具有可以运行 Android 系统的特性，最后这个系统就可以使特定的硬件上运行 Android 系统了。另外一种办法是先制作出合适版本的 Android 系统，然后从硬件厂商那里得到针对开发板的系统补丁，把这个补丁打到 Android 系统上，就可以得到针对这个硬件设备的 Android 系统。本文采用的是第二种方法，首先得到标准的 Android 系统内核，然后对该内核进行相应的处理，最终得到在 S5PV210 开发板上运行的 Android 系统。

移植 Android 系统可以大致分成三步：bootloader 的移植、kernel 移植和 Android 源码移植。

系统移植是一项复杂、仔细、工作量相当大的工程，在移植过程中，很多内在获外在的因素都会影响到移植成果，下面是移植系统过程中最重要的几点因素：

1、毅力。在移植过程中，可以说百分之百的会碰到各种各样的棘手问题，感觉自己在这个过程中碰的头破血流，很容易放弃这条道路另寻它途。但是这时必须也只有坚持下去，不断地碰到问题锲而不舍地解决它，是贯彻整个移植过程的主线。

2、前人的指引：移植 android 是一项工作量庞大的工程，在这项工程中，如果能得到前辈的指引，对移植的方向和方法有一个正确的认识，不但能够避免走弯路而且在面对棘手问题时能够找到最有效的解决办法。

3、对移植过程的客观认识：以前是否进行过 android 内核与源码的移植，对 Linux device model、Linux kernel、java 命令、linux 操作系统、嵌入式系统、android 文件系统、上层开发等的熟练程度。

### 3.1 S5PV210 开发板介绍

S5PV210 开发板是一款高性能的 Cortex-A8 核心板，采用三星 S5PV210 作为主处理器，运行主频高达 1GHz。S5PV210 内部集成了 PowerVR SGX540 高性能图形引擎，支持 3D 图形流畅运行，并可播放 1080P 大尺寸高清视频，流畅运行 Android、Linux 和 WinCE6 等高等级操作系统。图 3-1 是 S5PV210 核心板。

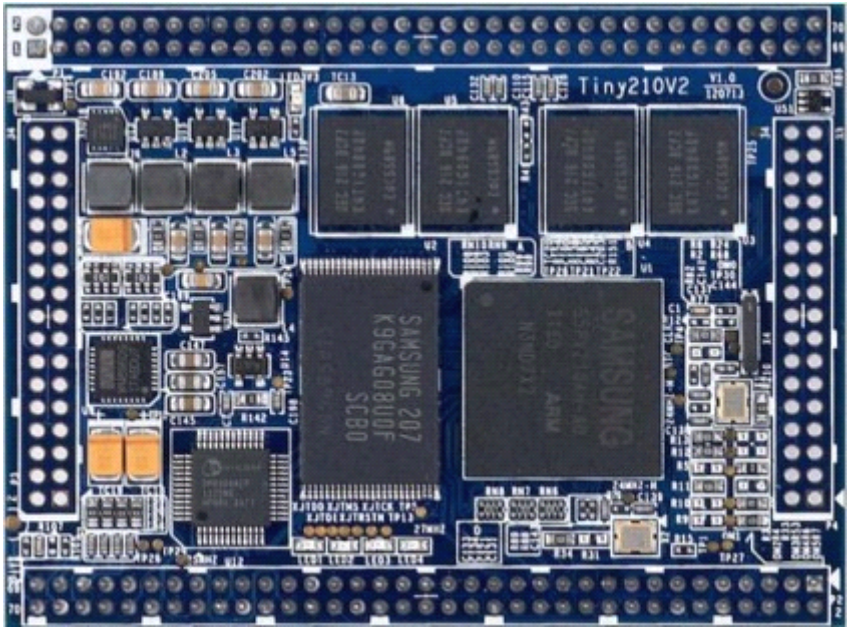


图 3-1 S5PV210 核心板

核心板资源特性如表 3-1 所示：

Item	Description
CPU 处理器	Samsung S5PV210，基于 Corex™-A8，运行主频 1GHz 内置 PowerVR SGX540 高性能图形引擎 支持流畅的 2D、3D 图形加速
DDR2 RAM 内存	Size: 512MB 32bit 数据总线，单通道 运行频率：200MHz
FLASH 存储	SLC NAND Flash:512M
在板资源	4 x User Leds(Green) 1 x Power Led(Red) HDMI 接口



电源供电	Supply Voltage from 2V to 6V（支持睡眠唤醒模式）
------	--

表 3-1 S5PV210 核心板资源特性

## 3.2 制作引导程序

每一套系统都需要自己的引导程序，缺少了这些引导程序，系统无法进行正常的启动。最常接触的例子，比如 Windows 的 BIOS，从开机上电开始，最先执行的就是 BIOS 引导程序，windows 的 BIOS 功能相当强大，可以在系统运行过程中进行系统自检和诊断。对应于 Window 的 BOIS，在 Linux 基础上运行起来的 Android 系统拥有的是 Bootloader，它负责初始化系统的硬件设施，硬件初始化完成后把 kernel 镜像和 yaffs2 格式的文件系统镜像从 FLASH ROM 映射到 RAM 中去，完成系统的加载并设置 kernel 的启动参数，然后执行 kernel，直到把 kernel 运行起来，就结束自己的程序，让出 CPU 给 kernel，然后 kernel 对系统做初始化设置，完成后把 CPU 交给文件系统<sup>[22]</sup>。

Bootloader 的种类有很多，LILO、GRUB、Loadlin、ROLO、Etherboot、LinuxBOIS、BLOB、U-boot、RedBoot，Bootloader 的适用范围受到两方面的限制：第一是 CPU 架构，第二是开发板的硬件设施。以上类型的 Bootloader 全部支持 x86 架构的开发板，但是只有一部分支持 ARM 和 PowerPC，目前应用的最广泛的是 U-boot。

本文采用的是基于 S5PV210 微处理器的嵌入式开发板，针对开发板的硬件资源，对 U-boot 进行适当的修改，使其能够完成 S5PV210 开发板的程序引导功能。

1、打开根目录下的 boarts.cfg 文件，找到 smdk210，然后在 smdk210 下面添加：smdk210 arm S5PV210 — Samsung s5pv210。

2、对看门狗定时器的设定，设置 WTCON=0x0，屏蔽看门狗定时器，不允许其复位。

3、为在调试和运行中，排除其他干扰，需要屏蔽掉所有中断信号，使 U-boot 在执行的时候不用处理其他中断，设置 INTMSK=0x7fffff。

4、为保证在 U-boot 运行过程中，处理器能正常工作，开发板的工作主频需要和 U-boot 一致，S5PV210 的工作主频为 1GHz，因此需要根据 S5PV210 的工作主频修改宏 CONFIG\_S5PV210\_CLOCK\_SPEED，保证开发板可以正常运行。

5、在/include/config 文件里可以对 S5PV210 开发板嵌入式头文件进行相应的更改，为了保证移植过程中可以在终端看到相应的启动信息，需要根据开发板的串口传输波特率修改波特率信息。

6、为 U-boot 添加 FLASH 分区信息：

```
+{0x00000000, 0x000041FF, FLAG_PROTECT_SET,    0, "Bootstrap"},
+{0x00004200, 0x000083FF, FLAG_PROTECT_CLEAR,   0, "Environment"},
+{0x00008400, 0x00041FFF, FLAG_PROTECT_SET,    0, "U-Boot"},
+{0x00042000, 0x00251FFF, FLAG_PROTECT_CLEAR,   0, "Kernel"},
+{0x00252000, 0xFFFFFFFF, FLAG_PROTECT_CLEAR,   0, "FS"},
```

7、配置编译 u-boot 得到需要的 u-boot.img 文件

3.3 Android 内核移植

内核移植涉及到几个相似的内核，如表 3-2 所示。

Mainline kernel	这种内核可以从主线下下载(即 <a href="http://www.kernel.org">http://www.kernel.org</a> )，它是没有经过任何更改的内核。
Reference kernel	这是针对开发板的内核，一般是把主线内核和针对开发板的内核补丁结合在一起形成的。
Merged kernel	该内核是把 Reference kernel 和 Android 内核补丁结合形成的内核，这种内核应该支持 Android 系统
Android patches	该补丁是由 google Android 内核提取出来的，是主线内核版本导入到 Android 内核树中所用到的补丁。参考 <a href="http://android.qit.kernel.org">http://android.qit.kernel.org</a>
Android kernel	这种内核可以从谷歌的 Android 开源项目资源库中下载得到。其中包含了谷歌为支持 Android 系统对主线内核进行的修改。下载地址： <a href="http://android.qit.kernel.org/kernel/common.qit">http://android.qit.kernel.org/kernel/common.qit</a> 。

表 3-2

把主线纯净的内核移植到一个具体的开发板上可用图 3-2 表示，大体的步骤分为三步：

- 1、主线内核和针对开发板设备内核相融合，获得参考内核。
- 2、Mainline kernel 和实现 Android 系统功能的内核相融合，得到 Android 系统的内核，它是正常运行 Android 系统的基础。
- 3、把第一步和第二步中得到的内核进行融合，得到支持 Android 系统并且适合该开发板的内核。

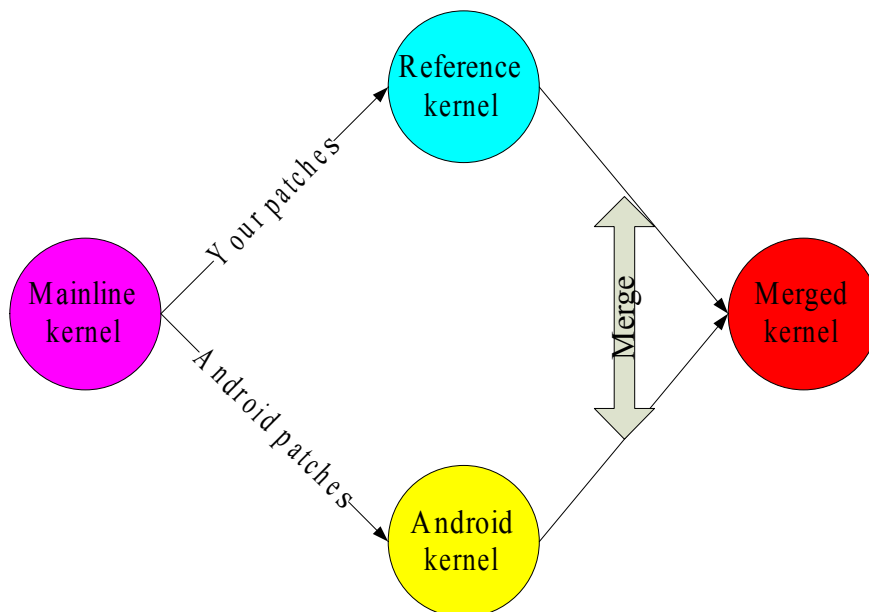


图 3-2 内核关系

本文采用的是直接从谷歌网站上下载 Android 内核，然后根据开发板对内核进行适当的修改，使之成为包容针对开发板特性的内核，支持 Android 系统在 S5PV210 开发板上运行。

移植内核的过程是一项庞大的工程，本文仅讲述其中的重点移植方面，其他的硬件的移植可根据以下内容进行相应的修改。

### 3.3.1 Android 内核与 Linux 内核的区别

在了解 Android 内核对 Linux 内核的变动之前，首先明确 Android 和 Linux 之间的关系：

- 1、Android 内核是建立在 Linux 内核的基础之上的。
- 2、Android 对 Linux 内核进行修改，目的是为了支持 Android 系统运行的设备以及其他一些 Android 不同于 Linux 的功能。
- 3、最初 Android 是以 Linux 分支的形式产生的，后来因某些原因无法并入 Linux 主开发树，被内核组删除<sup>[23]</sup>。

由此可见，Android 是运行在 Linux 基础上的操作系统，但它并不是 Linux 系统的某个版本。

Android 对 Linux 内核的改动可以概括为以下几点：

### 1、Android Binder

Android 系统每个应用都运行在一个独立的进程，Android 的 Binder 驱动是 google 公司为了保证进程间的通信，开发出一款基于 OpenBinder 框架的驱动。Binder 驱动替换掉了原有的支持 Linux 系统上层应用进程之间通信的 D-Bus(Desktop Bus)，即依靠消息总线的方式实现 IPC 机制。

### 2、Android 电源管理

Android 系统广泛的作为移动设备的操作系统，如何尽可能减少能量的消耗，达到省电的目的，对移动设备上的操作系统显得尤为重要。

### 3、低内存管理器

低内存管理器相比 Linux 标准的 OOM 显得更灵活，并且低内存管理器的代码非常简单，其中最重要的函数 `lowmem_dhrinker()` 会在初始化时注册一个 `lowmem_dhrinker`，在内存紧张时，被 vm 调用，寻找一个合适的进程杀死并释放它所占的内存，从而优化内存管理。

### 4、匿名共享内存

Android 系统提供了独特的匿名共享内存子系统 Ashmem，在 Kernel Space 中，它以驱动的形式实现。它通过 Binder 进程间的通信机制为进程间提供大块的内存共享。

### 5、Android Logger

Android Logger 是一种轻量级日志设备，可以抓捕系统的日志。其字体不同的颜色代表着不同的严重级别。

### 6、Android Alarm

Android Alarm 为 Android 系统提供了一个定时器，它可以把设备从睡眠状态唤醒，并且即使设备处于睡眠状态，它提供了始终基准依然可以正常运行。

### 7、Android RAM Console

应用程序开发的大部分时间都是花费在程序的调试上，因此，Android 通过 RAM Console 把调试日志进行输出。

## 8、Yaffs2 文件系统

Yaffs2(Yet Another Flash File System,2nd edition)文件系统是 Android 使用的 MTD nand flash 文件系统。Yaffs2 文件系统为 Linux 内核提供了一个高效访问 nand flash 的接口。同时，Yaffs2 文件系统有诸多优点：可以快速稳定的应用于 nand flash 和 nor flash；通其他 flash 文件系统相比占用的内存更小；在垃圾回收方面表现优异；更加适合大容量的 Flash 存储。然而对 nand flash 的支持并不包含在标准 linux2.6 内核中，因此 Android 系统只能添加 yaffs2 文件系统<sup>[24]</sup>。

### 3.3.2 制作 Android 内核

1、安装 arm-linux-gcc，并添加系统环境变量。

2、下载 Google 网站下的 Linux-3.0.8 内核源码，使用命令：`$ git clone https://android.googlesource.com/kernel/common.git`，然后选择适合自己的分支，即可。

3、为了使移植成功，让 Linux-3.0.8 可以在 S5PV210 开发板上运行，还需要把 Linux-3.0.8 的默认目标平台设置为 ARM 的平台，如图 3-3 所示。需要把总目录下的 Makefile 文件中的

```
export KBUILD_BUILDHOST:=%(SUBAECH)
```

```
ARCH ?=%( SUBAECH)
```

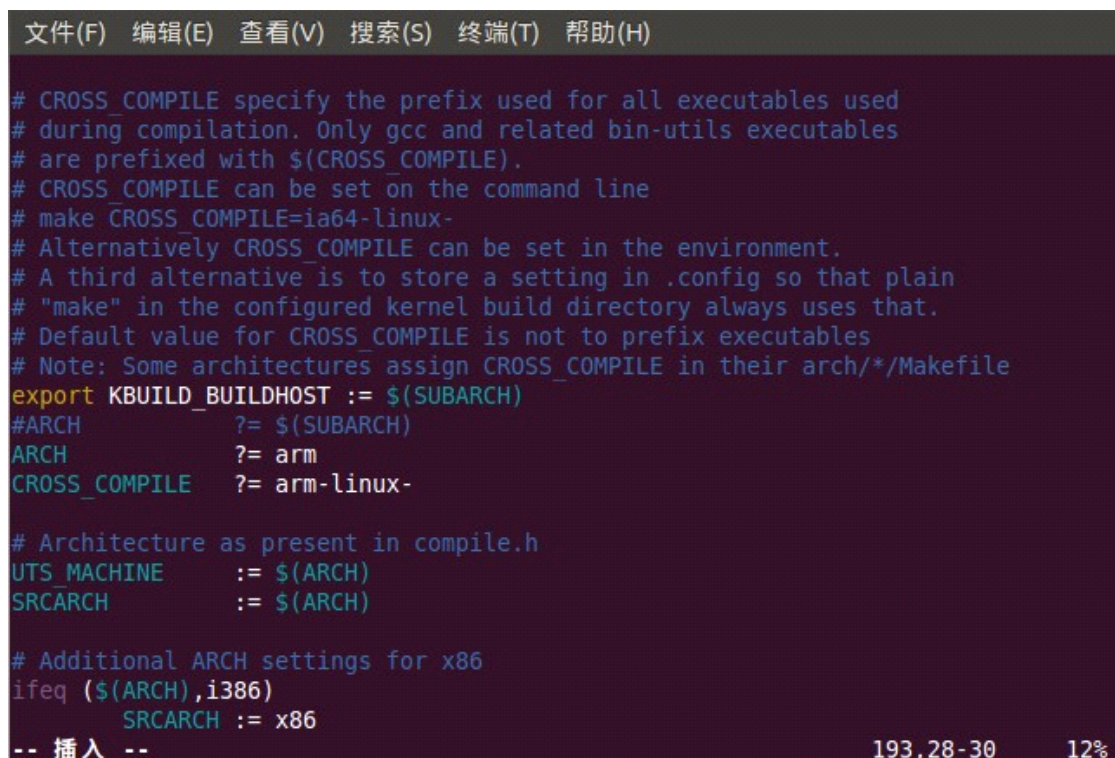
```
CROSS_COMPILE ?=
```

修改成：

```
export KBUILD_BUILDHOST:=%(SUBAECH)
```

```
ARCH ?=$ arm
```

```
CROSS_COMPILE ?=arm-linux-
```



```

文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

# CROSS_COMPILE specify the prefix used for all executables used
# during compilation. Only gcc and related bin-utils executables
# are prefixed with $(CROSS_COMPILE).
# CROSS_COMPILE can be set on the command line
# make CROSS_COMPILE=ia64-linux-
# Alternatively CROSS_COMPILE can be set in the environment.
# A third alternative is to store a setting in .config so that plain
# "make" in the configured kernel build directory always uses that.
# Default value for CROSS_COMPILE is not to prefix executables
# Note: Some architectures assign CROSS_COMPILE in their arch/*/Makefile
export KBUILD_BUILDHOST := $(SUBARCH)
#ARCH      ?= $(SUBARCH)
ARCH       ?= arm
CROSS_COMPILE ?= arm-linux-

# Architecture as present in compile.h
UTS_MACHINE := $(ARCH)
SRCARCH      := $(ARCH)

# Additional ARCH settings for x86
ifeq ($(ARCH),i386)
    SRCARCH := x86
-- 插入 --
193,28-30 12%
    
```

图 3-3

修改的作用是把 arm 平台作为 ARCH 指定的默认目标平台。CROSS\_COMPILE 指定交叉编译器的类型，如果使用的不是系统缺省的交叉编译器，一定要把编译器的全部绝对路径指定给 CROSS\_COMPILE<sup>[25]</sup>。

#### 4、修改 NAND Flash 分区。

在上文中引导程序的制作中对 Nand Flash 进行分区。这里的分区是为了把内核装载到 Nand Flash 上，当 U-boot 完成该操作时，CPU 的控制权转交给内核去处理，这时 U-boot 中只有一部分内核参数还需要被内核使用，被 U-boot 占用的大部分内存空间都要被回收，供内核利用，因此内核是不会用到 Bootloader 上的分区，即内核需要有自己的分区，否则无法加载系统。

首先，在 arch/arm/mach-s5pv210/devS.C 中指明分区信息。

其次，建立 Nand Flash 分区表：

/\*本文使用的目标板的 Nand Flash 为 512M，根据引导程序、内核和文件系统的占用空间进行分区\*/

```

Static struct mtd_partition partition_info[] = {
    {
        name: "bootloader",
        size: 0x00100000,
    }
}
    
```

```

        offset: 0x0,
    }, {
        name: "kernel",
        size: 0x00600000,
        offset: 0x00700000,
    }, {
        name: "bootloader",
        size: 0x0c800000,
        offset: 0x0cf00000,
    }, {
        name: "bootloader",
        size: 0x06400000,
        offset: 0x13300000,
    }
};

```

其中分配给 U-boot 分配 1M 空间，分配给内核分配 6M 空间，分配给文件系统分配 200M 空间，分配给用户数据分配 100M 空间。

第三，加入 Nand Flash 分区

```

struct s5pv210_nand_set nandset={
    nr_partiton:4,          //指明分区表信息中分区数目
    partition:partition_info, //分区表信息
};

```

第四，在 Nand Flash 驱动中建立 Nand Flash 芯片支持，分别是通过 struct s5pv210\_platform\_nand superllplatform{} 和修改 s5pv210\_device\_nand 结构体变量实现。

接下来实现启动时对 Nand Flash 分区的初始化。修改 arch/arm/mach-s5pv210/mach-smdk2410.c 文件中的结构体变量 smdk2410\_device[]，在其中加入&s5pv210\_device\_nand 语句即可。

最后因为本文是通过 U-boot 烧写内核到 Flash 上，U-boot 自身会通过 ECC 算法产生 ECC 校验码，这种校验码与内核校验的 ECC 不一样，因此需要把对内核的 ECC 校验禁止掉。方法是修改 drivers/mtd/nand/s5pv210.c 文件中的 s5pv210\_nand\_init\_chip() 函数，在函数的最后加上语句：chip->eccmode =

NAND\_ECC\_NONE；保存退出即可。

5、LCD 移植

LCD 的移植主要需要考虑两个方面，一个是系统缓存 frambuffer，它模仿显卡功能，从用户角度可以把 frambuffer 当做内存显示的一个映像。另外一个就是修改 LCD 的参数配置。

首先修改 LCD 驱动的相关参数，根据表 3-3 中 LCD 参数的阈值，修改 /arch/arm/mach-s5pv210/mach-smdk110.c 文件中的相关参数，修改初始化参数。修改上述 C 文件中的 static struct s3cfb\_lcd lte480wv 变量中的液晶屏的行列点数、行结束/开始等待周期、行同步脉宽、列结束/开始等待周期、列同步脉宽等参数。

参数	符号	最小值	推荐值	最大值	单位
DCLK	fcclk	—	30	40	MHz
	tcclk	—	33.3	—	ns
HSD	th	1114	1344	1440	tcclk
	thd	—	1024	—	tcclk
	thpw	1	—	140	tcclk
	thb	—	160	—	tcclk
	thfp	16	160	216	tcclk
VSD	tv	624	635	750	th
	tvd	—	600	—	th
	tvpw	1	3	20	th
	tvb	—	23	—	th
	tvfp	1	12	127	th

表 3-3

其次，设置 LCD 时钟频率，LCD 的时钟频率一般设定在 30~40SHZ 之间。S5PV210 芯片中有关时钟配置的相关参数如表 3-4 所示。

CLKVAL	60MHz/X	VCLK
2	60MHz/3	20.0MHz
3	60MHz/4	15.0MHz
:		



63	60MHz/64	937.5kHz
----	----------	----------

表 3-4

其中，VCLK 控制着像素时钟，这个值变大，帧频就会越大，屏幕闪烁的程度就会越小，但是这不代表着 VCLK 可以无限大，因为芯片本身对于 VCLK 有一个可接受范围。

由 LCD 手册中的公式： $RGB\_VCLK(Hz) = HCLK/(CLKVAL+1)$ ，计算出  $CLKVAL=4$ 。再由 S5PV210 的 Datasheet 中找到计算帧频的公式：

$$FrameRate = 1 / [ \{ (VSPW+1) + (VBPD+1) + (LINECAL+1) + (VFPD+1) \} \times \{ (HSPW+1) + (HBPD+1) + (HFPD+1) + (HOZVAL+1) \} \times \{ (CLKVAL+1) / (FrequencyofClocksource) \} ]$$

其中，HSPW 是行同步脉冲宽度，HBPD 是行可视范围前肩，HFPD 是行可视范围后肩，调节这三个参数负责控制 LCD 屏幕的左偏或者右偏。VSPW 是场同步脉冲宽度，VBPD 是帧可视前肩，VFPD 是帧可视后肩，调节这三个参数负责控制 LCD 屏幕的上偏或者下偏。LINEVAL 和 HOZVAL 是由 LCD 屏的分辨率来定的。计算出 Frame Rate = 41.56。因此把 s3cfb\_lcd.Freq 设置为 42。

最后，移植完 LCD 后发现每隔一段时间会自动黑屏，经过探查了解到需要注释掉 drivers/char/vt.c 中的 blank\_screen\_t(unsigned long dummy)函数，黑屏问题得到解决。

## 6、内核配置和编译

内核源码在烧写到目标开发板之前需要进行配置和编译，运行指令：`make menuconfig`，对内核进行编译之前的配置工作，配置内核有几点原则：首先要取消与 Goldfish 相关的所有选项，其次需要选择所有与 S5PV210 开发板有关的内核选项，还需要选择上必选项<sup>[26]</sup>。

配置完内核运行指令：`make`，进行内核编译，最后生成一个 Image 镜像文件，大约在 4.5M 左右，这就是最后可以烧写进开发板的内核镜像了。

## 3.4 制作 Android 源码镜像

### 3.4.1 获取 Android 源码

本文的开发环境为 Ubuntu10.10，在 linux 环境下下载 Android 源码大致分为以

下四步:

### 1、搭建 linux 开发环境

本文的 Android 移植是进行在 linux 环境下的，因此需要安装相应的包，具体操作命令如下:

```
sudo apt-get install git-core gnupg flex bison gperf libssl-dev build-essential zip curl  
libncurses5-dev zlib1g-dev
```

然后需要手动添加 JDK 库，执行命令如下:

```
sudo apt-get install sun-java6-jre sun-java6-plugin sun-java6-fonts sun-java6-jdk
```

在移植过程提取文件系统时，需要编译 SDK，make SDK，经过实际操作会发现 sun-java6-jdk 在编译 SDK 时出现错误，经过查找资料和实验发现，要下载 1.5 版本来兼容 “make doc” 这个命令，其命令如下:

```
sudo apt-get install sun-java5-jdk
```

另外还需要修改 javadoc 的链接对象，删除原有指向 1.6 版本的链接，建立指向 1.5 版本的链接。

### 2、设置环境变量

使用 vim 命令打开 .bashrc 文件，在其中新增和整合和 Java 开发运行相关的系统环境变量。

最后使新添加的系统环境变量生效:

```
source ~/.bashrc
```

### 3、安装编译工具

本文是在 Ubuntu10.10 环境下进行的开发工作，因此 git 和 repo 是开发 Android 必须安装的编译工具。git 负责有效地管理 android 项目版本，repo 的功能是更新 android 源码，是一种调用 git 的封装工具。

安装执行 repo 脚本的方式如下:

```
curl http://dl-ssl.google.com/dl/googlesource/git-repo/repo >repo
```

```
chmod a+x ~/bin/repo
```

### 4、获取 android 源码

首先初始化 repo:

```
~/bin/repo init -u https://android.googlesource.com/platform/manifest -b android-4.0.3_r1
```

下载源码：

repo sync

Android 源码非常庞大，由于网络的原因，在下载过程中经常会出现下载中断，此时需要重新输入 repo sync 命令即可，一般整个下载过程会持续 5、6 个小时。

### 3.4.2 制作 Android 文件系统镜像

把文件系统烧写进开发板，需要把文件系统制作成 yaffs2 格式的系统镜像。文件系统的制作有很多种办法，比如 make 后自动生成一个 out 文件夹，提取文件系统需要的文件都包含在这个文件夹下。比如提取文件系统需要的 root 文件夹和 system 文件夹都在包含在 out 文件夹下。这就是一个基本的文件系统，不过这个文件系统有很多的缺陷，在实际操作中，往往不提直接利用 out 文件夹制作烧写镜像，而是采取通过模拟器和 SDK 的方法提取文件系统。

通过 make 命令编译源码后，生成 out 文件夹，该文件夹包含了最重要的 root 文件夹。Android 源码中包含了完整的交叉编译工具和可发工具，因此只要能够正确的把它们编译出来就可以直接使用，这种做法相当有人性化，连工具版本问题也不需要考虑，不论是上层开发还是移植都需要用到这些脚本。

编译 SDK 使用命令：make sdk，在生成的/out/host/linux-x86/sdk/android-sdk\_eng.root\_linux-x86 文件夹里基本包含了 Android 模拟器和所有开发调试工具，比如 emulator、adb、mksdcard、android 等脚本。

通过模拟器和 SDK 提取文件系统时，需要把 busybox 推送进模拟器的 shell 中，因此需要事先准备好编译好的 busybox。配置编译后，会生成一个 busybox 可执行文件，这就是后面需要用到的文件。

下面需要创建一个虚拟 sdcard，再创建一个附加 sdcard 的模拟器，并运行这个模拟器，通过修改 system 目录的读写权限，把 busybox 推送进模拟器的 shell，利用 busybox 把需要的文件拷贝出来，完成文件系统的提取。

创建虚拟 sdcard，使用命令：

```
$/mksdcard -l 200M /home/sdcard.img
```

这个 sdcard 可以用来进行模拟器和外界数据之间的交换，因文件系统容量的原

因，这个 sdcard 不应该小于 100M，因此本文把它定位 200M。否则在后期打包文件系统的时候会出现空间不足的情况。

创建名为 android4.0.3 并附带有上述 sdcard 虚拟设备的模拟器：

```
$/android create avd -n android4.0.3 -t -c /home/sdcard.img 1
```

启动 android4.0.3 这个带有 sdcard 虚拟设备的模拟器：

```
$/emulator -avd android4.0.3 -sysdir /home/android/out/target/product/generic  
-data /home/android/out/target/product/generic/userdata.img -ramdisk /home/andr  
oid/out/target/product/generic/ramdisk.img -kernel /home/android/prebuild/android-  
arm/kernel/kernel-qemu
```

其中，-sysdir 指定模拟器系统路径，-data 指定模拟器加载的数据路径，-ramdisk 指定模拟器的虚拟内存，-kernel 指定模拟器的内核。

使用 adb 中的 remount 命令修改模拟器中的 system 目录的读写权限为可读写：

```
$/adb remount
```

利用 adb 工具中的 push 命令把之前编译出来的 busybox 可执行文件推送到模拟器的/system/bin 下，并进入到 shell 中：

```
$/adb push busybox /system/bin
```

```
$/adb shell
```

可执行文件把 system 和 data 中的所有文件进行大包并拷贝到前面创建的虚拟 sdcard 中：

```
$/tar xvf /sdcard/system.tar /system
```

```
$/tar xvf /sdcard/userdata.tar /data
```

把虚拟 sdcard 挂载到 mnt 目录下，并退出 shell：

```
$mount -o loop sdcard.img /mnt
```

```
$exit
```

要把目标文件系统全部写到开发板中，一般还需要先把目标文件系统目录制作成单个镜像文件以便烧写，Linux 内核启动时，会根据命令行的参数不同挂载不同格式的系统，例如 yaffs2、ubifs、ext2 等，本文使用 mkyaffs2image-128M 来制作文件系统镜像，这个工具适用于 SLC NAND Flash。使用命令如下：

```
$mkyaffs2image-128M rootfs_dir rootfs_android.img
```

这样得到最后需要的 rootfs\_android.img 这个 yaffs2 格式的 android 文件系统镜像。

### 3.5 移植成果

把内核镜像和源码镜像烧写到开发板上，重启系统，运行效果如图 3-4 所示，虽然已实现初步移植，但仍有很多功能不能实现，需要进一步研究和完善。



图 3-4

### 3.6 本章小结

本章通过对引导程序和 Google Android 内核的修改初步实现了 Android 系统移植到 S5PV210 开发板上的目标，但是距离全面完整的移植还有一段距离，需要进一步的研究完善。

## 4 MP3 播放器的设计与实现

成功得在 S5PV210 开发板上启动 Android 系统后，为了验证第三章的成果，在对 Android 上层开发进行了深入研究后，自主设计了一个 Andoird 项目，用来做成果测试。本章将介绍如何在 Eclipse 平台下设计 MP3 播放器，包括相应的界面显示并实现相应的播放音乐功能。

### 4.1 Android 开发环境搭建

Android 的上层开发是基于 Eclipse 平台进行的。Eclipse 是一个基于 Java 语言的可扩展平台。最初是 IBM 的一款软件，慢慢形成了当前主流的 Java 开发平台。Eclipse 在设计思路采用小核心多插件的思想，其功能大多是以插件的形式附加到一个小核心上实现的。Eclipse 有三个很吸引人的地方：1、创新性的图形 API；2、插件机制，Eclipse 平台中的每样东西都是插件；3、利用其插件机制开发出的众多功能强大的插件<sup>[27]</sup>。Eclipse 安装好之后，运行 hello world 程序，效果如图 4-1 所示。

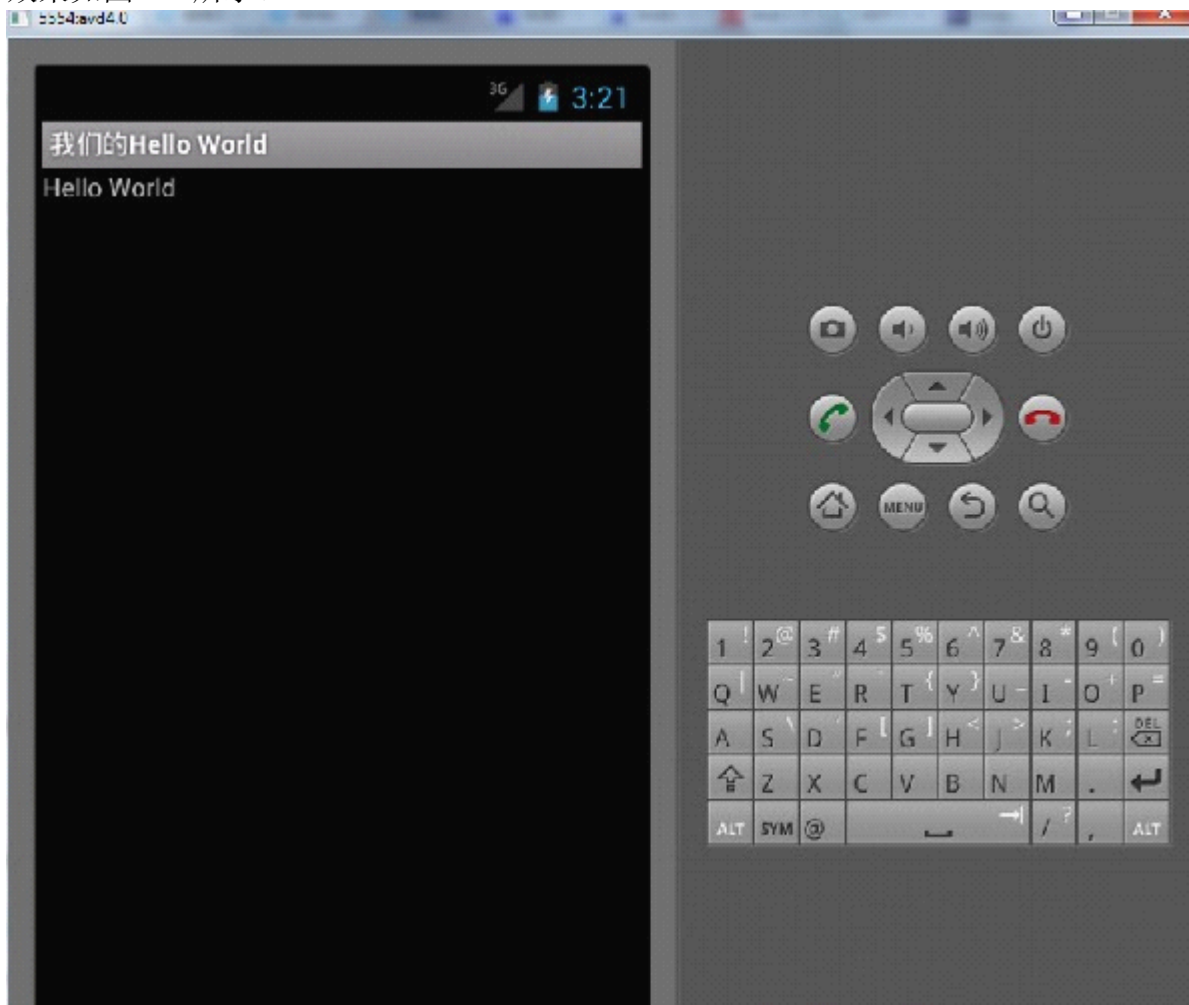


图 4-1

## 4.2 Android 应用程序文件目录

Android 应用程序的开发是基于 Eclipse 平台之上的，每个 android 应用程序都有以下几个文件基本文件组成：src、gen、assets、bin、res 5 个基本文件夹和 AndroidManifest.xml 文件。

src 目录下存放着应用程序的 Java 文件。

gen 目录下存放的是由 Android 开发工具自动生成的 R.java 文件。存放在 res 目录下的 xml 界面文件、图片、界面控件等等会通过 Android 开发工具自动更新修改其 ID 并保存到 gen 目录下的 R.java 文件中。R.java 在整个应用程序中起到资源索引的作用，编写应用程序时，开发者使用的资源和系统上是别的 ID 是一一对应的，极大地方便了开发者进行应用开发<sup>[28~29]</sup>。

assets 目录用来保存用户的资源，默认为不会更新索引到 R.java 文件，但是它可以结合 SQLite 数据库使用<sup>[30]</sup>。

res 文件夹保存着应用程序中所有的资源。layout 保存着应用程序中的 Activity 的布局文件，大部分的界面控件都需要在该文件中进行注册获取对应的 ID。values 文件夹下的文件是保存了应用程序中变量的值，以键值对的形式出现。在 values 目录下的 string.xml 文件中存放的是 string 对象的键值对，每个 string 对象可以对应多个值，大部分应用程序都可以设置不同版本的语言风格，主要依靠的是 string.xml 文件的作用。

bin 目录存放着应用程序的二进制文件，程序源码在编译链接后就在该目录下生成对应的 apk 文件<sup>[31]</sup>。

AndroidManifest.xml 是功能清单文件，所有的 Activity、Service 都需要在此进行注册，并且注册的 Activity 和布局文件中的 activity 是一一对应<sup>[32]</sup>。默认的启动 Activity 为 Main\_Activity。

## 4.3 MP3 播放器的设计与实现

上层开发是基于应用框架层之上的，因此必须要了解下层的应用程序框架提供了什么功能的基类和接口。Android 框架中的应用程序框架层为上层开发应用程序提供了大量的、丰富的、功能各异的基类，其中的 MediaPlayer 基类为开发



者开发多媒体播放类得应用程序提供了极大地方便<sup>[33]</sup>。

本文播放器的设计注重人性化、简易化和功能多样化。

人性化：每个按钮在按下和释放两种不同状态下显示不同的效果，当按钮按下时，按钮周围出现光晕效果，按钮被释放时，恢复原来的状态，光晕效果消失，体验者会体验到一种奇妙很奇妙的感觉。

简易化：播放器的所有供用户体现的功能几乎都包容在最核心的播放界面，实现各种功能的按钮以播放/暂停按钮为中心左右对称排列，其排列的顺序和位置能够帮助用户更容易理解每个按钮的功能。用手拖动位于播放界面的进度条上的浮标，位于进度条左侧显示当前播放时间的控件会随着浮标的显示出相应的时间。

功能多样化：播放器所能实现的功能，除了最基本的播放音乐的功能外，还包括快进、后退、播放上一首、播放下一首、随机播放、单曲循环播放，以此更加符合用户的需求。

MP3 播放器的实现大致分为两个步骤：第一、设计播放器的界面，比如程序需要涉及到几个 Activity，每个 Activity 里面界面控件的设置以及界面背景等等。第二、编写源码实现播放器的相关功能，这一步涉及到源码主要都是存放在 src 文件夹里的 Java 程序。程序的整体文件结构如图 4-2 所示。

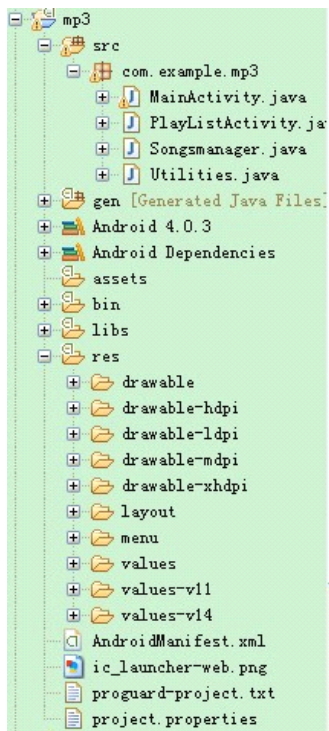


图 4-2



### 4.3.1 需求分析

播放器实现功能：

- (1)、基本音乐播放控制（播放、暂停）。
- (2)、上一曲、下一曲控制。
- (3)、快进、快退控制。
- (4)、获取 SD 卡中所有的 mp3 文件，并显示在音乐列表界面。
- (5)、单击音乐文件列表，播放音乐。

程序能够自动识别存入 SD 卡的 mp3 文件，把检测出来的 mp3 文件名称形成列表显示在一个 Activity 界面上。选中其中的任何一个 mp3 文件名称点击播放按钮实现播放功能，播放按钮转为暂停按钮。

点击快进、快退按钮分别实现快进 5 秒和后退 5 秒。点击上一首和下一首按钮分别实现播放上一首和下一首。播放界面中的进度条可以显示播放进度，并且拖动进度条的位置能调整对应的播放进度。随机播放按钮和单曲循环按钮分别实现随机播放音乐和单曲循环的功能。点击文件列表按钮进入 mp3 文件列表，并且显示相应的文件信息。

### 4.3.2 界面设计

本项目虽然只有两个界面，但是设计时还是花费了很多的心思。

#### (1) 运行界面

运行界面有两个：音乐播放界面和音乐列表界面。

音乐播放界面实现了音乐播放功能，该页面为主界面，由 7 个播放控制按钮、一个播放进度条和一个音乐列表显示按钮组成。

#### (2) 布局控件

依据设计的界面选取对应的布局方式和控件。主界面的布局框架选用的是 RelativeLayout，其中镶嵌了多个播放控制按钮、进度条和播放时间显示控件。上一首、快退、播放/暂停、快进、下一首 5 个 ImageButton 播放控制按钮镶嵌其中，位于整个界面的下端，这 5 个按钮采用 LinearLayout 布局一字排开<sup>[34]</sup>。

#### (3) 细节完善

Activity 的布局有 LinearLayout、RelativeLayout、TableLayout 三种布局方式，

巧妙地利用这三种布局方式、实现布局方式的相互嵌套以及利用“盒子模型”，合理布置各种控件和父控件之间的绝对和相对位置，可以让程序呈现出耳目一新的感觉。

该音乐播放器的主界面采用 `LinearLayout` 和 `RelativeLayout` 相互嵌套的方式，整体设计结构紧凑，美观大方。布局方式的嵌套如下：

< `RelativeLayout`

    <`LinearLayout`>——→排列随机播放、单曲循环播放

    <`LinearLayout`>——→排列显示歌曲名、和播放列表按钮

    <`LinearLayout`>——→排列快进、快退、上一首、下一首、播放/暂停等控制按钮

    <`LinearLayout`>——→排列进度条、已播放时间、总播放时间

>

每个 `ImageButton` 使用的背景资源都来自 `drawable` 资源文件，他们实际上是一个 `xml` 文件，利用这些 `XML` 文件可以非常轻松的处理按钮的点击反馈，对应用程序的开发起到很大的促进作用。

控件需要规范的基本设置包括：长度、宽度、背景图片、颜色、字体大小，对于 `RelativeLayout` 中的控件，还需要设置水平方向居中、竖直方向居中、与相邻控件的外边距、和父控件的相对位置等等。

设计好播放器的主界面后，还需要设计播放列表界面，用来显示所有歌曲名称，并显示歌曲的相关信息。播放器歌曲列表的实现采用 `Listview` 控件，该控件的布局文件位于 `res/drawable/playlist.xml`。`Listview` 可以包含所有歌曲的信息，单首歌曲的界面显示则通过 `Listview` 中的 `item` 实现，其布局文件位于 `res/drawable/playlist_list.xml`。`Item` 的布局方式采用 `LinearLayout` 的相互嵌套，界面被两个 `LinearLayout` 成左右排列分割，每个 `LinearLayout` 里都嵌套了一个小的 `LinearLayout`，分别用来显示歌曲名称、歌手信息、总播放时间和正在播放图标。嵌套方式如下：

<`LinearLayout`

    <`LinearLayout`>——→显示歌曲名称

>

在 ListView 的 Item 中有一个 Switch 控件，当 Item 过多屏幕装不下时，上下滑动使未显示的 Item 显示，这时开始和结尾的几个 Item 中的 Switch 控件会自动改变状态，并响应状态改变的事件，因此不用担心 Item 过多的问题。

布局设计的主要代码分布在 mp3/res 文件夹下，该文件夹的主要目录及作用如表 4-1：

draeable 文件夹		该文件夹保存了 button_back、button_forward、button_next、button_pause、button_playlist、button_privious、button_repeat、button_shuffle、进度条等控件的资源图片
layout	activity_main.xml	播放器住播放界面的布局文件，规定了播放按钮、进度条、歌曲信息、播放列表按钮的排列位置
	playlist_item.xml	设置了单首歌曲信息的排列规则
	playlist.xml	囊括了所有 playlist_item，并规定它们的显示方式
values/strings.xml		定义多组键值对，方便了语言版本的更换

表 4-1

4.3.3 功能实现

在 android 原生的媒体播放器中是由 mediaplayerservice 来控制媒体播放器的，在 mediaplayerservice 中创建了 mediaplayer。该类用来实现音乐播放、视频播放等多媒体功能。MeidaPlayer 的状态有游离态（Idle）、初始态（Initialized）、准备态（Prepared）、运行态（Started）、完成态（Playback Completed）、停止态（Stopped）、装载态（Preparing）、暂停态（Paused）、异常态（Error）和结束态（End）<sup>[35]</sup>。一些常用状态中允许的操作如表 4-2：

状态 \ 操作	prepare()	Start()	pause()	stop()
Initialized	是	否	否	否
Prepared	否	是	否	是
Started	否	是	是	是
Pause	否	是	是	是
Stopped	是	否	否	是

Playback Completed	否	是	否	是
--------------------	---	---	---	---

表 4-2

在搭建完界面后，编写 java 源码搭建音乐播放界面和歌曲列表界面并实现相应的功能。该部分的设计大致分为四个步骤：

首先，自定义一个 SongManager 类，通过该类获取设备中所有 mp3 文件。其次，自定义 Utilities 工具类，用于额外工作的处理，比如时间处理。然后，自定义一个 Activity 界面 PlayListActivity 用于显示播放列表。最后，搭建播放器的主播放界面 MainActivity，包含各种播放控制控件并实现其相应的功能。其中 PlayListActivity、MainActivity 是继承四大组件中的 Activity，需要在 Android.manifest.xml 中进行注册，分别声明这两个 Activity 的 name、label 两项属性，并在 MainActivity 中添加 Intentfilter 标签，标签中声明该 Activity 是启动后的第一个 Activity，声明语句如下所示：

```
<action android:name="android.intent.action.MAIN" />
```

以上四个步骤主要的代码文件在总目录中的位置如下：

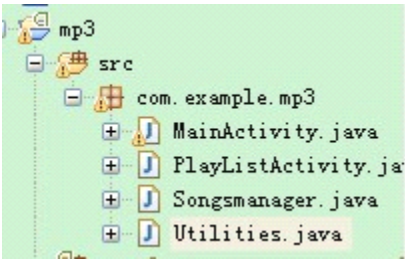


图 4-3

1、SongManager类

本音乐播放器在设计时采用遍历文件，筛选出其中后缀名称为mp3的文件，用户使用时，所有可播放的mp3文件全部显示在playlistactivity界面，一目了然。该类所用到的头文件如表4-3：

java.io.File	指定扫描设备路径时，file home =new File(media_path);
java.io.FilenameFilter	获得设备中所有文件名称后，筛选后缀为mp3的文件
java.util.ArrayList	把所有HashMap中的文件名和文件路径打包进Arraylist
java.util.HashMap	把音乐文件的路径和文件名分别作为一对键值对，打包进HashMap

表 4-3

具体实现方法：首先定义 `Media_Path` 字符串，用来定义扫描设备的路径，此处路径为 “/sdcard/”。再定义一个 `songList` 变量，该变量的类型为 `ArrayList<HashMap<string,string>>`，用来存放返回的播放列表的键值对。`SongManager` 类很重要的一个步骤是在设备文件名中筛选出后缀为 `mp3` 的文件。`Android` 平台为开发者提供了一个十分实用的接口：`FilenameFilter`，该接口被称作文件名称过滤器，用来过滤不合格的文件名并返回合格的文件名，实现这个接口需要指定文件的当前目录和文件名。指定文件目录和名称后，把后缀为“.mp3”的文件返回到 `FileExtensionFilter` 子类，过滤掉其他格式的文件。如果在扫描设备所有文件中遍历筛选出来的音乐文件的个数不为空 `home.listFiles(new.FileExtensionFilter()).length>0`，则进入循环，每次循环都把扫描设备中的所有文件名称和文件路径保存到 `HashMap<String,String>` 类型的 `song` 变量中，然后通过 `songlist` 中的 `add` 方法把 `song` 变量添加进 `songlist`，执行语句为 `songList.add(song)`，该循环一直被执行，直到所有音乐文件都被扫描一遍。扫描到的文件数目通过 `file.getName().length()` 来控制。然后把得到所有文件名称和文件路径键值对的 `songList` 作为返回值，返回 `getPlayList()` 子类。到此，程序得到了扫描设备中所有以 `.mp3` 为后置名的文件名称和文件路径并全部保存在 `getPlayList()` 内部类中，后面就可以直接调用 `SongManager` 方法得到设备文件中所有音乐文件的名称和路径了。

## 2、PlaylistActivity 类

通过 `SongMannager.java` 获得文件设备中所有以“.mp3”为后缀的音乐文件，接下来需要把这些音乐文件的信息显示在屏幕界面上，方便用户查看与选择，这项任务交给 `PlaylistActivity` 来实现。

该类继承于 `ListActivity` 这个基类，在该类的入口函数处有 `Activity` 子类标志性的两个函数，一个是 `onCreat()` 方法，该方法在 `Activity` 第一次被调用时用于创建 `Activity`，另外一个为 `setContentView()` 方法，把 `Activity` 与资源文件中的布局文件绑定在一起。

因为在 `SongManager` 中声明的变量其有效地范围只局限在该类内部，`PlaylistActivity` 和 `SongManager` 是两个独立平等的类，因此在 `SongManager.java` 中定义的所有变量并不适用于 `PlaylistActivity` 类。要把 `SongManager` 类中扫描筛选出的文件设备中的“.mp3”音乐文件相关的数据应用到 `PlaylistActivity` 类中，只

能在播放列表界面的 Activity 类中声明一个 `ArrayList<HashMap<String,String>>` 类型的 `songlist` 变量，通过调用 `SongManager` 类中的 `getPlaylist()` 方法，把其中文件设备中扫描到的音乐文件信息传递到 `songlist` 变量。使用 `for` 循环，其循环次数用 `songList.size()` 来控制，通过循环把刚刚得到的 `songlist` 变量中的歌曲信息数据传递给类型为 `HashMap<String, String>` 类型的 `song` 变量并通过该变量把所有歌曲的歌曲名称赋值给 `songlistData` 变量，至此文件设备中所有以“.mp3”为后缀的音乐文件信息都存储到 `HashMap` 类型的 `songlistData` 变量中，该变量是在 `PlayListActivity` 类中声明的，因此可以在该类中任意使用。为把 `songlistData` 变量中歌曲信息显示到视图上，需要通过使用 `SimpleAdapter` 实现。该方法是一个适配器功能，通过指定数据支持的列表，例如由 `ArrayList` 组成的 `Hashmap`，能将静态数据映射到 XML 文件中定义好的视图上。

把音乐文件的信息显示到界面之后，用户需要点击单手歌曲并且播放，因此还需要对单手歌曲的 `listView` 设置监听器，用以监听用户点击 `listView` 事件，做出相应的处理。当用户点击单首歌曲，程序可以获得这首歌曲的相关信息，并把这首歌曲的 `position` 传递给 `int` 类型的 `songIndex` 变量，并把该变量打包进 `Intent` 变量，同时指定 `Intent` 所要发送的对象，即播放界面 `MainActivity`，由它去处理该事件。

### 3、MainActivity 类

`MainActivity` 是播放器的主界面，接受从 `PlaylistActivity` 发送到的 `Intent` 并处理 `Intent` 中包含的数据，得到歌曲信息，并播放该歌曲。`MainActivity` 还需要实现播放/暂停按钮、快进按钮、后退按钮、上一首按钮、下一首按钮、进度条跟进、播放列表按钮、随机播放按钮、单曲循环按钮等众多按钮的功能，除此之外，该 Activity 还要处理歌曲播放完成后进入下一首播放事件。

在对界面上的按钮进行操作之前，需要首先在 `MainActivity` 中对它们进行声明，然后通过资源标识符中每个控件唯一拥有的 ID 找到该按钮资源，然后才可以对该按钮设置事件监听器进行功能实现。以播放/暂停按钮为例，首先声明一个 `ImageButton` 变量：`private ImageButton btnplay;` 然后通过资源标识符中的 ID 找到对应的按钮资源：`btnPlay = (ImageButton) findViewById(R.id.btnPlay);` 之后可以编写代码实现播放/暂停按钮的功能了。

在该 Activity 中地应了一个很重要的内部类 `playSong()`，该内部类是实现播放音乐功能的一个基础，它主要作用是在程序调用其他控制播放音乐的操作时进行初始化，包括音乐播放时间复位、获取音乐名称使用 `songTitleLable.setText()` 方法把当前播放的音乐名称显示在界面上、准备音乐、播放音乐、进度条最大值和最小值的初始化并调用更新进度条的函数 `updataProgressBar()`。

当用户点击播放列表中任何一首歌曲时，`PlayListActivity` 会发送指向播放器主播放界面的 `Intent`，触发 `MainActivity` 接收到 `Intent` 中有关的音乐信息后，获得该音乐文件的 `Index`，然后调用 `playSong()` 方法播放该音乐文件。

播放/暂停按钮功能的实现比较简单，当有事件触发监听器时，通过 `mp.isPlaying()` 方法，判断当前是否正在播放音乐，如果返回值为 `true`，则暂停音乐同时更换该按钮的图片资源文件，把图标转到暂停显示状态；否则播放音乐并更换按钮图片资源，在界面上显示播放状态。

快进按钮，监听器被触发时，首先得到当前音乐播放的进度 `currentPosition`，如果当前播放进度和快进进度超过了音乐文件总播放时间，则直接跳转到音乐的最后面；否则从当前播放进度加上快进进度的位置开始播放。

快退按钮，监听器被触发时，首先得到当前音乐播放的进度 `currentPosition`，如果当前播放进度小于后退进度，则直接跳转到音乐的最开始播放；否则从当前播放进度减去后退进度的位置开始播放。

下一首按钮，外部事件触发该按钮的监听器时，首先得到当前播放音乐文件在音乐列表中的位置 `currentSongIndex`，如果该变量的值小于所有音乐文件总和减去 1，则按照音乐文件的排列顺序播放下一首歌，否则播放第一首歌曲文件。

上一首按钮，外部事件触发该按钮的监听器时，首先得到当前播放音乐文件在音乐列表中的位置 `currentSongIndex`，如果该变量的值大于 0，则按照音乐文件的排列顺序播放上一首歌，否则播放最后一首歌曲文件。

单曲循环播放按钮，当该按钮的监听器被触发后，首先判断当前的播放模式是否为循环播放，如果返回值为 `true`，则取消循环播放模式并把循环播放按钮的图片资源更换成非循环播放状态的图片；否则设置播放模式为循环播放并弹出对话框提示用户当前已取消随机播放模式，同时更换随机播放按钮和循环播放按钮的图片资源。

随机播放按钮，当该按钮的监听器被触发后，首先判断当前的播放模式是否为随机播放，如果返回值为 `true`，则取消随机播放模式并更换随机播放按钮的图片资源；否则设置播放模式为随机播放并弹出对话框提示用户当前已取消单曲循环模式，同时更换随机播放按钮和循环播放按钮的图片资源。

当一首音乐播放完成时，程序会自动检查播放模式，如果是单曲循环播放，则重新播放这首歌曲；如果是随机播放模式，则在播放列表中随机选一首歌曲播放；如果没有指定播放模式，则播放下一首歌曲。

#### 4、Utilities.java 类

这是一个辅助工具类，其中包含三个内部类，实现三方面的功能：

内部类 `milliSecondsToTimer(long)`：把毫秒转化为 `hours: minites: seconds` 的标准格式，返回该时间格式字符串显示在界面上。

内部类 `intgetProgressPercentage(long,long)`：获得进度条进度百分比，通过计算当前播放时间占总播放时间的百分比，获得浮标在进度条上的位置，从而直观的显示播放进度，该方法的核心部分是计算百分比：`percentage = (((double)currentSeconds)/totalSeconds)*100`。

内部类 `intprogressToTimer(int,int)`：把进度条显示的进度转换成当前播放时间，核心代码为 `currentDuration = (int) (((double)progress) / 100) * totalDuration`。

### 4.4 成果展示

开发程序或者系统最耗时的事不是编写代码，而是程序或者系统调试，经过反复的调试，最终成功把播放器运行在开发板上，效果如图 4-4：





图 4-4

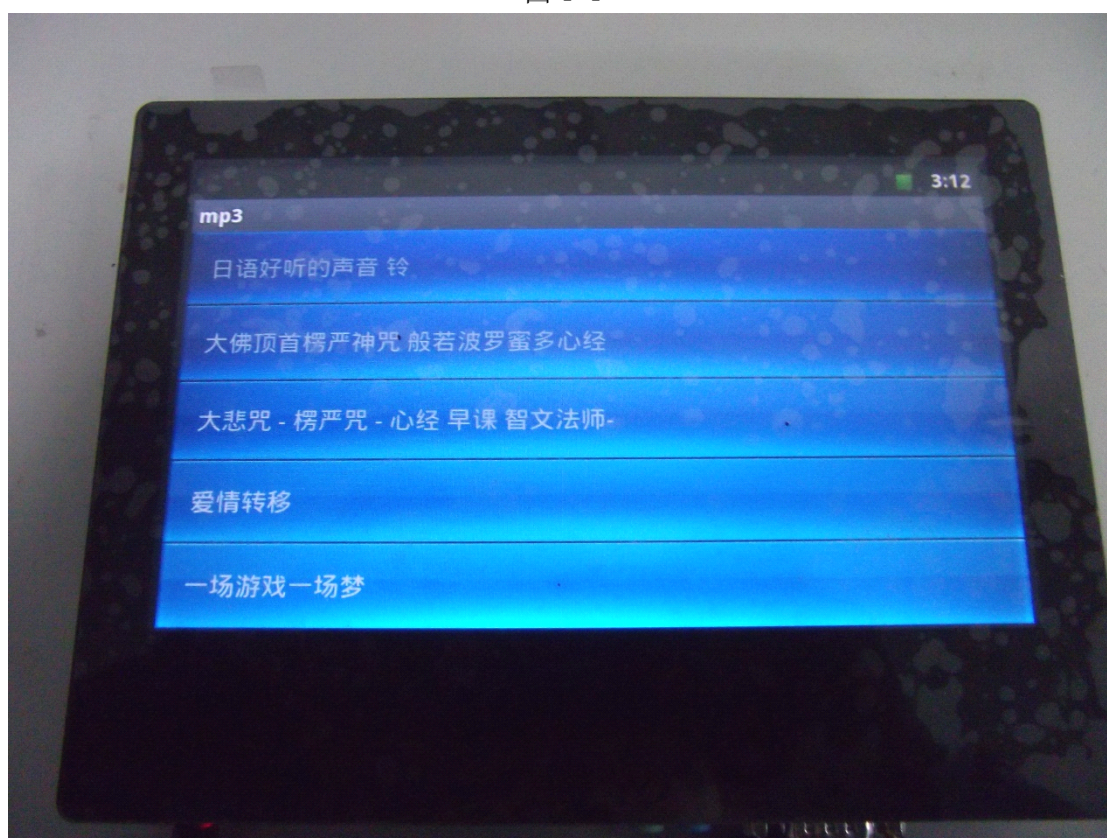


图 4-5

## 4.5 本章小结

本章介绍了 Android 开发工具 Eclipse 的安装，详细分析了自主设计的音乐播放器实现的功能，界面设计的原理和播放器涉及到的四个大类 Songsmanager 类、PlaylistActivity 类、MainActivity 类和 Utilities 类的设计原理与实现，可以作为 Android 项目的一个典型实例进行理解。

## 5 总结与展望

伴随着通信技术的发展，手机逐渐普及，手机的款式从原来“大哥大”式的四四方方的笨重逐渐演变成现在外观小巧、风格多样的现代手机，功能从原来简单的打电话发短信逐渐发展到集各种功能于一身、随时随地为人们提供娱乐、资讯等多种服务，手机系统也从原来的封闭单一演变成现在开放式百家争鸣式的系统。

当封闭、功能单一的手机已不能满足当前人们对手机的需求时，智能手机应运而生，并逐渐成为手机行业的发展趋势。手机系统是智能手机的灵魂，因此，当前智能手机系统呈现出激烈的竞争态势，主流手机系统有黑莓手机系统、Windows Mobil 手机系统、Android 手机系统、iPhone 手机系统和塞班手机系统。Android 手机系统凭借其源码开放性和易移植的特点，在众多手机系统中显得尤为特殊和重要，不但在手机市场中深受广大用户的喜好，而且正逐渐被应用到其他电子产品中。

现对本文工作进行总结如下：

1、介绍了手机发展历程，从用户体验的角度阐述了智能手机系统的特点，介绍了当前国内外几种主流智能手机系统，详细介绍了 Android 手机系统的特点与当前国内外对 Android 系统移植的研究现状。

2、研究了 Android 系统架构设计，它采用分层结构：应用程序层、应用程序框架层、系统运行库层、Linux 核心层，Android 系统四大组件以及系统的启动过程。

3、分析了影像系统移植成败的客观因素和 Android 内核与 Linux 内核的区别，简要介绍了 S5PV210 开发板核心板资源。Android 系统移植分为三大步骤：制作引导程序、修改 Android 内核、制作 Android 源码镜像。其中最重要的是内核移植部分，本文采用在 Google 的 Android 内核基础上进行修改和裁剪，最终得到适用于 S5PV210 开发板的 Android 内核，制作镜像，烧写到开发板上成功运行 Android 系统。

4、介绍了 Android 开发平台 Eclipse 工具的安装和完整 Android 项目所包含的文件结构，自主设计了 mp3 播放器界面并介绍了 Songsmanager 类、

PlaylistActivity 类、MainActivity 类和 Utilities 类四个大类,实现音乐播放器功能,并安装到开发板上,进行测试,达到预期目的,间接证明移植成果。

对本文内容的展望:

虽然本文初步实现了 Android 系统在 S5PV210 开发板上的移植,但目前的系统运行状态仍不稳定,并且有很多功能和设备没有进行移植,比如网卡、USB 等,因此,把 Android 系统完全移植到开发板上仍需要做进一步的研究和完善,这需要更多的时间和精力。

另外,因为对 Android 上层开发的理解不深刻,本文中的音乐播放器只实现了简单的功能,需要完善的功能有很多,比如同步下载显示歌词、显示歌曲的专辑和歌手名称等信息、制作进入和退出界面以及改变播放器界面的单一风格,实现多种界面任意选取的功能。



## 参考文献

- [1]中国信息产业园. 手机的发展史 [EB/OL]. 2007.07.06, <http://www.cnii.com.cn/20070520/ca419965.htm>
- [2]大河网. 2G 真的 OUT 了 [EB/OL]. 2012.08.03, [http://newpaper.dahe.cn/dhb/html/2012-08/03/content\\_758779.htm?div=-1](http://newpaper.dahe.cn/dhb/html/2012-08/03/content_758779.htm?div=-1)
- [3] 邵艳杰.Android 操作系统移植及应用研究 [D].湖南: 湖南大学, 2011.
- [4] Android developer. What is Android. [EB/OL]. [http://develop.android.comnyud.net/guide/basics/what-is -android.html](http://develop.android.comnyud.net/guide/basics/what-is-android.html), 2010-12-28
- [5] 吴亚峰, 苏亚光. Android 2.0 游戏开发实战宝典 [M].北京: 人民邮电出版社, 2010.07.01
- [6] Gasolin. IntroAndroid [EB/OL]. 2010-02-01, <http://code.google.com/p/androidbmi/wiki/IntroAndroid>
- [7] 李苑. 基于 Android 移动平台的景物查询系统[D]. 南昌: 南昌大学, 2009.
- [8] 马志强. 基于 Android 平台即时通信系统的设计与实现[D]. 北京: 北京交通大学, 2010.
- [9] 刘必刚.Android 通信模块的设计与优化 [D]. 武汉: 武汉理工大学, 2010.
- [10] 韩超, 梁泉.Android 系统原理及开发要点详解 [M]. 北京: 电子工业出版社, 2010.
- [11] Marko Gargenta. Learning Android[M]. O'Reilly Media, 2011
- [12] Frank Ableson. Introduction to Android development [EB/OL]. <http://www.ibm.com/developerworks/opensource/library/os-android-devel/index.html>, 2009.6.12.
- [13] Ed Burnette. Hello , Android: Introducing Google's Mobile Development Platform [M]. Progmatic Bookshelf, 2010: 76-85
- [14] 马越.Android 的架构与应用 [D].北京: 中国地质大学, 2008.
- [15] Enck W, Ongtang M, McDaniel P. Understanding Android Security [C]. IEEE Computer Society, 2009.
- [16] 叶炳发.Android 操作系统移植及关键技术研究 [D].广东省: 暨南大学, 2010.
- [17] Android developer. Intents and Intent Filters [EB/OL]. <http://developer.android.android.com.nyud.net/guide/topics/intents/intents-filters.html>, 2010-12-28
- [18] 郭宏志.Android 应用开发详解 [M]. 北京: 电子工业出版社, 2010:46-125.
- [19] 靳岩, 姚尚朗. Google Android 开发入门与实战 [M]. 北京: 人民邮电出版社, 2009: 289-314.

- [20] Shu Xianhua, Du Zhenjun, Chen Rong. Research on Mobile Location Service Design Based on Android [C]. Beijing:IEEE Xplpre, 2009.
- [21] 杨丰盛.Android 应用开发揭秘[M].北京:机械工业出版社, 2010: 46-58
- [22] Vasileeios Laganakos. From Zero to Boot: Porting Android to your ARM platform [EB/OL].2011-06-02 ,  
<http://blogs.arm.com/software-enablement/498-from-zero-to-boot-porting-android-to-your-arm-platform/>
- [23] Greg Kroah-Hartman. Android and the Linux kernel community [EB/OL]. 2010-02-02 ,  
<http://www.kroah.com/log/linux/android-kernel-problems.html>
- [24] 王振丽.Android 底层开发技术实战详解[M]. 北京:电子工业出版社, 2012-08: 6-8.
- [25] Mark L. Murphy. Beginning Android 2[M]. Apress, 2010: 94-103
- [26] 张利国, 龚海平, 王植萌.Android 移动开发入门与进阶[M].北京:人民邮电出版社, 2009.9:178-231
- [27] 高昂. 支持动态语言的 Android 平台 [M]. 北京:人民邮电出版社, 2009-05-07:156-109.
- [28] 余志龙, 陈昱勋, 郑名杰.Android SDK 开发范例大全[M].北京:人民邮电出版社, 2010.
- [29] Reto Meier. Professional Android Application Development [M].北京:清华大学出版社, 2010:120-238
- [30] 高焕堂.Android 应用框架原理与程式设计 36 技 [M].台湾:广悦文化事业有限公司, 2009:56-83
- [31] 高焕堂.Google Android 设计招式之美 [M].台湾:广悦文化事业有限公司, 2009:103-135
- [32] W.Frank, Ableson Charlie, Collins Robi. Google Android 揭秘 [M].北京:人民邮电出版社, 2010:68-75
- [33] 杨文志. Google Android 程序设计指南[M]. 北京:电子工业出版社, 2009:351-373.
- [34] J.F. DiMarzio. Android: A Programmers' Guide [M]. McGraw-Hill Osborne Media , 2008:320-291.
- [35] 何建明.Android 精髓 [M]. 北京:清华大学出版社, 2008: 209-265.



## 致 谢

值此论文完成之际，向支持、帮助我的导师、家人、朋友表示最衷心的感谢与诚挚的敬意，在两年的研究生求学生涯中，是你们的肯定和鼓励给予我面对困难的勇气，是你们为我提供了学习钻研的温床。

首先，感谢王国宇教授两年来对我的工作和学习的大力支持和帮助，为论文的完成提供了良好的研究与工作环境。

本文的大部分工作是在研发中心进行的，在此我衷心的感谢曾小光工程师和周全工程师，感谢两位工程师在论文选题、研究方法、论文撰写等方面给予的大力帮助，曾小光工程师敏锐的洞察力、朴实的研究作风和周全工程师认真的工作态度、对新事物的理解能力以及对我的悉心教导都给我留下了深刻的印象，他们不仅为论文研究提供了莫大的帮助，更是指引我人生方向的指明灯。同时，衷心的感谢在研究撰写论文期间一直支持鼓励我的同事、朋友，正是他们的帮助和支持才能让我顺利完成论文。

最后，感谢我的母校中国海洋大学多年来对我的培育，感谢所有曾经教授过我知识的老师和帮助过我的同学。



## 个人简历

1988 年 4 月 14 日出生于河北省沧州市。

2007 年 9 月考入中国海洋大学信息科学与工程学院电子信息科学与技术专业，2011 年 7 月本科毕业并获得理学学士学位。

2011 年 9 月考入中国海洋大学信息科学与工程学院电子与通信工程专业攻读硕士学位至今。