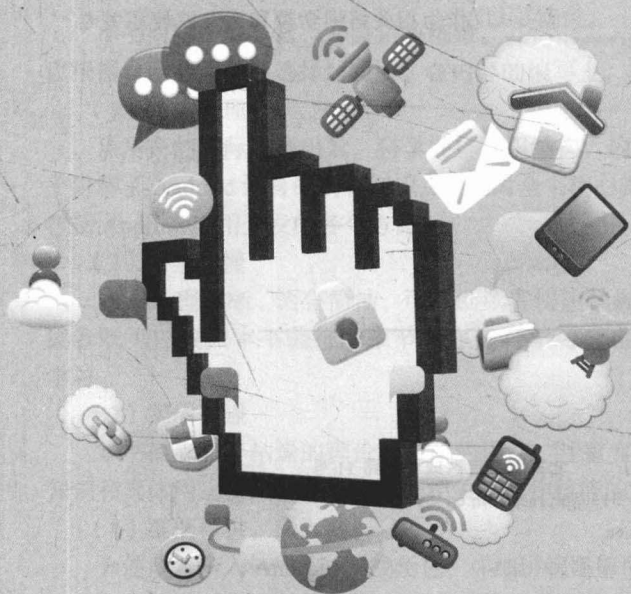




# Android Studio 应用开发

实战  
详解

王翠萍◎编著



人民邮电出版社  
北京

## 图书在版编目(CIP)数据

Android Studio应用开发实战详解 / 王翠萍编著

— 北京 : 人民邮电出版社, 2017.2

ISBN 978-7-115-43673-3

I. ①A… II. ①王… III. ①移动终端—应用程序—  
程序设计 IV. ①TN929.53

中国版本图书馆CIP数据核字(2016)第289787号

## 内 容 提 要

全书共分18章,依次讲解了Android开发基础、搭建Android开发环境、Android Studio集成开发环境介绍、Android Studio常见操作、分析Android应用程序文件的组成、Gradle技术基础、UI界面布局、Material Design设计语言、核心组件介绍、Android事件处理、图形图像和动画处理、开发音频/视频应用程序、GPS地图定位、Android传感器应用开发、编写安全的应用程序、Google Now和Android Wear、Android应用优化以及Android TV开发。本书几乎涵盖了Android Studio应用开发所能涉及的所有领域,在讲解每一个知识点时,都遵循了理论联系实际的讲解方式,用具体实例彻底剖析了Android Studio开发的每一个知识点。本书讲解方法通俗易懂,特别有利于初学者学习并消化。

本书适合Android初级读者、Android应用开发人员、Android爱好者、Android Studio开发人员、Android智能家居、Android可穿戴设备研发人员学习,也可以作为相关培训学校和大专院校相关专业的教学用书。

---

◆ 编 著 王翠萍  
责任编辑 张 涛  
责任印制 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
固安县铭成印刷有限公司印刷

◆ 开本: 787×1092 1/16  
印张: 25  
字数: 734千字  
印数: 1-2500册

2017年2月第1版  
2017年2月河北第1次印刷

---

定价: 69.00元

读者服务热线: (010)81055410 印装质量热线: (010)81055316  
反盗版热线: (010)81055315

# 前 言

因为从 Android 6.0 开始，谷歌公司宣布不再对 Eclipse 环境提供任何技术支持，而是主推 Android Studio 开发环境。为了顺应当今技术潮流，笔者编写了这本 Android Studio 应用开发的书籍，详细讲解了使用 Android Studio 工具开发 Android 应用程序的基本知识。

## 本书的版本

Android 系统自 2008 年 9 月发布第一个版本 1.1 以来，截止到 2016 年的最新版本 7.0，一共存在十多个版本。由此可见，Android 系统升级较快，一年之中最少有两个新版本诞生。如果过于追求新版本，会造成力不从心的结果。所以在此建议广大读者：不必追求最新的版本，我们只需关注最流行的版本即可。据官方统计，截至 2016 年 4 月 14 日，占据前 3 位的版本分别是 Android 4.4、Android 5.0 和 Android 6.0。其实这 3 个版本的差别并不是很大，只是在某领域的细节上进行了更新。

## 本书特色

本书内容十分丰富，并且讲解细致。我们的目标是通过一本图书，提供多本图书的价值，读者可以根据自己的需要有选择地阅读。在内容的编写上，本书具有以下特色。

### (1) 结构合理

从用户的实际需要出发，科学安排知识结构，内容由浅入深，叙述清楚，具有很强的实用性，几乎讲解了 Android 语言的所有知识点。同时全书精心筛选的具有代表性、读者关心的典型知识点，几乎包括 Android 应用开发的各个方面。

### (2) 易学易懂

本书条理清晰、语言简洁，可帮助读者快速掌握每个知识点。每个部分既相互连贯又自成体系，读者既可以按照本书编排的章节顺序进行学习，也可以根据自己的需求对某一章节进行针对性地学习。

### (3) 实用性强

本书彻底摒弃枯燥的理论和简单的操作，注重实用性和可操作性，详细讲解了各个知识点的基本知识和具体用法实例，使用户掌握相关的操作技能的同时，还能学习到相应的知识。

### (4) 案例精讲，深入剖析

为使读者步入 Android 实战角色，详细讲解了每一案例的详细实现流程。通过学习本书，读者不但对基本知识点进行了系统的学习，而且能够从实战中轻松掌握各个知识的综合运用技巧，为读者将来更深层次的学习打下坚实的基础。

## 读者对象

- 初学移动开发的自学者
- Android Studio 开发人员

- Android智能家居开发人员
- Android可穿戴设备研发人员
- 大中专院校的老师和学生
- 从事移动开发的程序员
- 编程爱好者
- Android开发人员
- 相关培训机构的老师和学员

## 特别注意

本人在编写过程中，得到了人民邮电出版社工作人员的大力支持，正是各位编辑的求实、耐心和效率，才使得本书在这么短的时间内出版。另外也十分感谢我的家人，在我写作的时候给予的巨大支持。由于本人水平有限，纰漏和不尽如人意之处在所难免，诚请读者提出意见或建议，以便修订并使之更臻完善。另外为本书提供了售后支持网站（<http://www.toppr.net/>），读者如有疑问可以在此提出，一定会得到满意的答复。在<http://www.toppr.net/>中还提供了如下所示的免费资源供读者下载：

- 本书所有的源代码；
- 赠送500个实例代码；
- 1000页PDF电子书；
- 40小时的Android教学视频。

作者

# 目 录

第1章 Android开发基础	1
1.1 移动智能设备系统发展现状	1
1.2 Android 系统基础	2
1.2.1 Android 系统的发展现状	2
1.2.2 常见的 Android 设备	2
1.3 Android 系统架构	3
1.3.1 最底层的操作系统层 (OS) ——C/C++实现	4
1.3.2 Android 的硬件抽象层——C/C++实现	4
1.3.3 各种库 (Libraries) 和 Android 运行环境 (RunTime) ——中间层	5
1.3.4 应用程序框架 (Application Framework)	6
1.3.5 应用程序 (Application) ——Java 实现	6
1.4 Android 和 Linux 的关系	6
1.4.1 Android 继承于 Linux	6
1.4.2 Android 和 Linux 内核的区别	7
1.5 Android 开发学习路线图	8
1.5.1 Android 开发的两大方向	8
1.5.2 Android 应用开发需要具备的基础知识	9
第2章 搭建Android开发环境	11
2.1 Android Studio 介绍	11
2.2 准备工作	12
2.2.1 系统要求	12
2.2.2 获取并安装 JDK	12
2.3 官方方式获取并安装 Android Studio	15
2.3.1 官方方式获取工具包	15
2.3.2 安装工具包	17
2.4 非官方方式获取并安装工具包	18
2.4.1 快速下载站点介绍	18
2.4.2 单独获取并安装 Android Studio	19
2.4.3 单独获取并安装 Android SDK	20
2.5 启动 Android Studio	22
2.5.1 启动前的设置工作	22
2.5.2 正式启动	24
2.6 通过官网学习搭建环境	25
第3章 Android Studio集成开发环境介绍	27
3.1 Welcome to Android Studio 面板	27
3.2 系统设置面板	28
3.3 系统默认设置面板	29

3.3.1	Appearance & Behavior (外观与行为) 面板	30
3.3.2	Keymap (快捷键) 面板	32
3.3.3	Editor (编辑器) 面板	33
3.4	主界面面板	34
3.4.1	菜单栏	35
3.4.2	工具栏	37
3.4.3	左侧面板	38
3.4.4	中间编辑区域	39
3.4.5	底部调试区域	40
3.4.6	右侧模式面板	42
<b>第4章</b>	<b>Android Studio常见操作</b>	<b>43</b>
4.1	新建一个新工程	43
4.2	Android SDK 操作	47
4.2.1	Android SDK 管理器操作	48
4.2.2	设定 Android SDK Location	49
4.2.3	安装/删除/更新 Android SDK	50
4.2.4	集中管理 Android SDK	54
4.3	AVD 模拟器操作	56
4.3.1	创建新的 AVD 模拟器	57
4.3.2	启动 AVD 模拟器	61
4.3.3	修改 AVD 模拟器	62
4.3.4	删除 AVD 模拟器	63
4.4	导入/导出操作	64
4.4.1	导入一个既有 Android Studio 项目	64
4.4.2	导入一个既有 Eclipse 项目	66
4.5	导入/导出设置	69
4.5.1	Import Settings (导入设置)	69
4.5.2	Export Settings (导出设置)	69
4.6	第一个 Android 应用程序	70
4.6.1	使用 Android Studio 新建 Android 工程	70
4.6.2	编写代码和代码分析	71
4.6.3	调试程序	71
4.6.4	模拟器运行项目	72
4.6.5	真机运行项目	73
<b>第5章</b>	<b>分析Android应用程序文件的组成</b>	<b>76</b>
5.1	两种目录结构概览	76
5.1.1	Project 模式	76
5.1.2	Android 模式	76
5.2	“app”目录	77
5.2.1	“java”子目录	77
5.2.2	“res”子目录	78
5.2.3	设置文件 AndroidManifest.xml	79
5.3	“Gradle Scripts”目录	80
5.3.1	文件 build.gradle	81
5.3.2	文件 gradle.properties	81
5.3.3	文件 settings.gradle	82

第6章 Gradle技术基础	83
6.1 两种目录结构概览	83
6.1.1 安装 Gradle	83
6.1.2 Gradle 的主要功能	83
6.2 Android 工程中的 Gradle	84
6.2.1 文件 app/build.gradle	84
6.2.2 “gradle”目录	85
6.2.3 根目录中的 build.gradle	86
6.2.4 其他文件	86
6.3 和 Android Studio 相关的几个命令	86
6.4 Android Studio 中的常见 Gradle 操作	87
6.4.1 查看并编译源码	87
6.4.2 创建二进制发布版本	88
6.4.3 Gradle 多渠道打包	90
6.4.4 自定义 BuildConfig	92
第7章 UI界面布局	93
7.1 View 视图组件	93
7.1.1 View 的常用属性和方法	93
7.1.2 Viewgroup 容器	94
7.1.3 ViewManager 类	94
7.2 Android UI 布局的方式	94
7.2.1 使用 XML 布局	95
7.2.2 在 Java 代码中控制布局	95
7.3 Android 布局管理器详解	96
7.3.1 Android 布局管理器概述	96
7.3.2 线性布局 LinearLayout	98
7.3.3 相对布局 RelativeLayout	99
7.3.4 帧布局 FrameLayout	101
7.3.5 表格布局 TableLayout	101
7.3.6 绝对布局 AbsoluteLayout	102
7.3.7 网格布局 GridLayout	103
7.3.8 实战演练——演示各种基本布局控件的用法	104
第8章 Material Design设计语言	110
8.1 Material Design 概述	110
8.2 Material Design 设计原则	111
8.3 环境因素：属性和阴影	111
8.4 动画设计	112
8.4.1 感知动作	112
8.4.2 响应式交互	112
8.5 实战演练——使用 Material Design	113
8.5.1 项目概览	113
8.5.2 程序文件	114
8.5.3 布局文件	120
8.5.4 实现测试程序	125
第9章 核心组件介绍	131
9.1 Widget 组件	131



9.1.1	创建一个 Widget 组件	131
9.1.2	使用按钮 Button	132
9.1.3	使用文本框 TextView	133
9.1.4	使用编辑框 EditText	135
9.1.5	使用多项选择控件 CheckBox	136
9.1.6	使用单项选择控件 RadioGroup	138
9.1.7	使用下拉列表控件 Spinner	139
9.1.8	使用自动完成文本控件 AutoCompleteTextView	142
9.1.9	使用日期选择器控件 DatePicker	143
9.1.10	使用时间选择器 TimePicker 控件	144
9.1.11	联合应用 DatePicker 和 TimePicker	145
9.1.12	使用滚动视图控件 ScrollView	148
9.1.13	使用进度条控件 ProgressBar	148
9.1.14	使用拖动条控件 SeekBar	149
9.1.15	使用评分组件 RatingBar	150
9.1.16	使用图片视图控件 ImageView	151
9.1.17	使用切换图片控件 ImageSwitcher 和 Gallery	152
9.1.18	使用网格视图控件 GridView	155
9.1.19	使用标签控件 Tab	156
9.2	使用 MENU 友好界面	158
9.2.1	MENU 基础	158
9.2.2	实战演练——使用 MENU 控件	158
9.3	使用列表控件 ListView	160
9.4	使用对话框控件	162
9.4.1	对话框基础	162
9.4.2	实战演练——在屏幕中使用对话框显示问候语	163
第10章	Android事件处理	169
10.1	基于监听的事件处理	169
10.1.1	监听处理模型中的3种对象	169
10.1.2	Android系统中的监听事件	171
10.1.3	实现事件监听器的方法	172
10.2	基于回调的事件处理	177
10.2.1	Android事件侦听器的回调方法	177
10.2.2	基于回调的事件传播	179
10.2.3	重写 onTouchEvent 方法响应触摸屏事件	181
10.3	响应的系统设置的事件	182
10.3.1	Configuration 类详解	183
10.3.2	重写 onConfigurationChanged 响应系统设置更改	185
10.4	Handler 消息传递机制	186
第11章	图形图像和动画处理	191
11.1	Android 绘图基础	191
11.1.1	使用 Canvas 画布	191
11.1.2	使用 Paint 类	193
11.1.3	位图操作类 Bitmap	195
11.2	使用其他的绘图类	198
11.2.1	使用设置文本颜色类 Color	198
11.2.2	使用矩形类 Rect 和 RectF	199
11.2.3	使用变换处理类 Matrix	203

11.2.4 使用 BitmapFactory 类	205
11.3 使用 Drawable 实现动画效果	208
11.3.1 Drawable 基础	208
11.3.2 使用 Drawable 实现动画效果	208
11.4 Tween Animation 动画详解	209
11.4.1 Tween 动画基础	210
11.4.2 Tween 动画类详解	213
11.4.3 Tween 应用实战	215
11.5 实现 Frame Animation 动画效果	216
11.5.1 Frame 动画基础	216
11.5.2 使用 Frame 动画	217
11.6 Property Animation 动画	218
11.6.1 Property Animation (属性) 动画基础	219
11.6.2 使用 Property Animation	220
11.7 实现动画效果的其他方法	223
11.7.1 播放 GIF 动画	223
11.7.2 实现 EditText 动画特效	225
<b>第12章 开发音频/视频应用程序</b>	<b>227</b>
12.1 音频应用接口类概述	227
12.2 AudioManager 类	228
12.2.1 AudioManager 基础	228
12.2.2 AudioManager 基本应用——设置短信提示铃声	230
12.3 录音处理	233
12.3.1 使用 MediaRecorder 接口录制音频	233
12.3.2 使用 AudioRecord 接口录制音频	234
12.4 播放音频	236
12.4.1 使用 AudioTrack 播放音频	237
12.4.2 使用 MediaPlayer 播放音频	237
12.4.3 使用 SoundPool 播放音频	239
12.4.4 使用 Ringtone 播放铃声	240
12.4.5 使用 JetPlayer 播放音频	240
12.4.6 使用 AudioEffect 处理音效	241
12.5 语音识别技术	243
12.5.1 Text-To-Speech 技术	243
12.5.2 谷歌的 Voice Recognition 技术	246
12.6 实现振动功能	248
12.7 设置闹钟	249
12.7.1 AlarmManage 基础	249
12.7.2 开发一个闹钟程序	250
12.8 使用 MediaPlayer 播放视频	254
<b>第13章 GPS地图定位</b>	<b>260</b>
13.1 位置服务	260
13.1.1 类 location 详解	260
13.1.2 实战演练——在 Android 设备中实现 GPS 定位	261
13.2 随时更新位置信息	262
13.2.1 库 Maps 中的类	262
13.2.2 使用 LocationManager 监听位置	263

13.2.3 实战演练——监听当前设备的坐标和海拔	264
13.3 在 Android 设备中使用地图	270
13.3.1 申请 Google Map API 密钥	270
13.3.2 使用 Map API 密钥的注意事项	273
13.3.3 使用 Map API 密钥	274
13.3.4 实战演练——在 Android 设备中使用谷歌地图实现定位	275
13.4 接近警报	279
13.4.1 类 Geocoder 基础	279
13.4.2 Geocoder 的公共构造器和公共方法	281
<b>第14章 Android传感器应用开发详解</b>	<b>283</b>
14.1 Android 传感器系统概述	283
14.2 Android 传感器应用开发基础	283
14.2.1 查看包含的传感器	283
14.2.2 模拟器测试工具——SensorSimulator	285
14.2.3 实战演练——检测当前设备支持的传感器	288
14.3 使用光线传感器	290
14.3.1 光线传感器介绍	290
14.3.2 使用光线传感器的方法	291
14.4 使用磁场传感器	291
14.4.1 什么是磁场传感器	292
14.4.2 Android 系统中的磁场传感器	292
14.5 使用加速度传感器	292
14.5.1 加速度传感器的分类	293
14.5.2 Android 系统中的加速度传感器	293
14.6 使用方向传感器	294
14.6.1 方向传感器基础	294
14.6.2 Android 中的方向传感器	295
14.7 使用陀螺仪传感器	295
14.7.1 陀螺仪传感器基础	296
14.7.2 Android 中的陀螺仪传感器	296
14.8 使用旋转向量传感器	297
14.9 使用距离传感器详解	298
14.9.1 距离传感器介绍	298
14.9.2 Android 系统中的距离传感器	299
14.10 使用气压传感器	300
14.10.1 气压传感器基础	301
14.10.2 气压传感器在智能手机中的应用	301
14.11 使用温度传感器	301
14.11.1 温度传感器介绍	302
14.11.2 Android 系统中温度传感器	302
14.12 使用湿度传感器	304
<b>第15章 编写安全的应用程序</b>	<b>306</b>
15.1 Android 安全机制概述	306
15.1.1 Android 的安全机制模型	307
15.1.2 Android 具有的权限	307
15.1.3 Android 的组件模型 (Component Model)	308
15.1.4 Android 安全访问设置	308

15.2 声明不同的权限	309
15.2.1 AndroidManifest.xml 文件基础	309
15.2.2 声明获取不同的权限	309
15.2.3 自定义一个权限	313
15.3 发布 Android 程序生成 APK	314
15.3.1 什么是 APK 文件	314
15.3.2 申请会员	315
15.3.3 生成签名文件	318
15.3.4 使用签名文件	323
15.3.5 发布到市场	325
<b>第16章 Google Now和Android Wear详解</b>	<b>326</b>
16.1 Google Now 概述	326
16.1.1 搜索引擎的升级——Google Now	326
16.1.2 Google Now 的用法	327
16.2 什么是 Android Wear	328
16.3 开发 Android Wear 程序	329
16.3.1 创建通知	329
16.3.2 创建声音	331
16.3.3 给通知添加页面	333
16.3.4 通知堆	334
16.3.5 通知语法介绍	335
16.4 实战演练——开发一个 Android Wear 程序	336
<b>第17章 Android应用优化详解</b>	<b>341</b>
17.1 用户体验是产品成功的关键	341
17.1.1 什么是用户体验	341
17.1.2 影响用户体验的因素	342
17.1.3 用户体验设计目标	342
17.2 Android 优化概述	343
17.3 UI 布局优化	343
17.3.1 <merge />标签在 UI 界面中的优化作用	343
17.3.2 遵循 Android Layout 优化的两段通用代码	346
17.3.3 优化 Bitmap 图片	347
17.3.4 FrameLayout 布局优化	348
17.3.5 使用 Android 为我们提供的优化工具	352
17.4 Android Lint 静态分析	359
17.4.1 使用 Android Lint	360
17.4.2 解决错误	361
17.4.3 自定义 Android Lint 的检查提示	362
17.4.4 检查工程依赖的库	364
17.5 使用 Memory Monitor 内存分析工具	365
17.6 Code CleanUp (代码清理)	367
17.7 使用第三方工具	368
<b>第18章 Android TV开发详解</b>	<b>369</b>
18.1 Android TV 概述	369
18.2 系统模块结构	370
18.3 系统主界面	370
18.3.1 系统主界面布局	371

18.3.2 系统主界面 Activity	372
18.4 本地服务	377
18.4.1 本地服务主界面	377
18.4.2 应用程序管理界面	380

Android是一款操作系统的名称，是科技界巨头谷歌（Google）公司推出的一款运行于手机和平板电脑等设备的智能操作系统。因为Android系统的底层内核是以Linux开源系统架构的，所以它是Linux家族的产品之一。虽然Android外形比较简单，但是其功能十分强大。自从2011年开始到现在为止，Android系统一直占据全球智能手机市场占有率第一的宝座。在本章的内容中，将简单介绍Android系统的诞生背景和发展历程，为读者步入本书后面知识的学习打下基础。

## 1.1 移动智能设备系统发展现状

在Android系统诞生之前，智能手机这个新鲜事物大大丰富了人们的生活，得到了广大手机用户的青睐。各大手机厂商在市场和消费者用户需求的驱动之下，纷纷研发出了各种智能手机操作系统，并且大肆招兵买马来抢夺市场份额，Android系统就是在这个风起云涌的历史背景下诞生的。在了解Android这款神奇的系统之前，将首先了解当前移动智能设备系统的发展现状。

### 智能手机和移动智能设备介绍

智能手机是指具有像个人计算机那样强大的功能，拥有独立的操作系统，用户可以自行安装应用软件、游戏等第三方服务商提供的程序，并且可以通过移动通信网络接入到无线网络中。在Android系统诞生之前已经有很多优秀的智能手机产品，例如Symbian系列和微软的Windows Mobile系列等。

对于初学者来说，可能还不知道怎样来区分智能手机。某大型专业统计站点曾经为智能手机的问题做过一项市场调查，经过大众讨论并投票之后，总结出了智能手机所必须具备的功能标准，下面是当时投票后得票率最高的前5个选项：

- (1) 操作系统必须支持新应用的安装；
- (2) 高速度处理芯片；
- (3) 支持播放式的手机电视；
- (4) 大存储芯片和存储扩展能力；
- (5) 支持GPS导航。

根据大众投票结果，手机联盟制定了一个标准，并以这个标准为基础，总结出了如下智能手机的主要特点：

- (1) 具备普通手机的全部功能，例如可以进行正常的通话和发短信等手机应用；
- (2) 是一个开放性的操作系统，在系统平台上可以安装更多的应用程序，从而实现功能的无限扩充；
- (3) 具备上网功能；
- (4) 具备PDA的功能，实现个人信息管理、日程记事、任务安排、多媒体应用和浏览网页；
- (5) 可以根据个人需要扩展机器的功能；
- (6) 扩展性能强，并且可以支持第三方软件。

随着科技的进步和发展,智能手机被归纳到移动智能设备当中。在移动智能设备中,还包含了平板电脑、游戏机和笔记本电脑。

## 1.2 Android系统基础

Android一词最早出现于法国作家Auguste Villiers de l'Isle-Adam在1886年发表的科幻小说《未来夏娃》中,他将外表像人的机器起名为Android。本书的主角就是Android系统,在本节将简要介绍Android系统的诞生和发展历程。

### 1.2.1 Android系统的发展现状

从2008年HTC和Google联手推出第一台Android手机G1开始,在2011年第一季度,Android在全球的市场份额首次超过塞班系统,跃居全球第一。下面的几条数据能够充分说明Android系统的霸主地位。

(1) 2011年11月数据,Android占据全球智能手机操作系统市场52.5%的份额,中国市场占有率为58%。2015年12月消息,数据研究公司IDC公布了最新的报告,报告称至2019年谷歌仍将继续保持领先。在这份报告中指出,预计2019年Android系统将占据全球82.6%的移动系统市场份额,届时iOS的预计份额将为14.1%,也就是说未来4年苹果的市场份额将出现小幅下滑。

(2) 如果从某一个时间段进行统计,Android系统也是雄踞市场占有率第一的位置。据著名互联网流量监测机构Net Applications发布的最新数据显示,从2013年9月到2014年7月,在这将近一年的时间里,Android市场占有率却一直处于稳步攀升状态,从最初的29.42%狂飙至44.62%,而iOS的使用量却在一路下滑,从去年9月份的53.68%降至44.19%。

(3) 如果从市场硬件产品出货量方面进行比较,Android系统则具有压倒性的优势,其市场份额高达85%,而iOS仅占11.9%

由上述统计数据可见,Android系统的市场占有率位居第一,并且毫无压力。Android机型数量庞大,简单易用,相当自由的系统能让厂商和客户轻松地定制各样的ROM,定制各种桌面部件和主题风格。简单而华丽的界面得到广大客户的认可,对手机进行刷机也是不少Android用户所津津乐道的事情。

可惜Android版本数量较多,市面上同时存在着1.6到当前最新的6.x等各种版本的Android系统手机,应用软件对各版本系统的兼容性对程序开发人员是一种不少的挑战。同时由于开发门槛低,导致应用数量虽然很多,但是应用质量参差不齐,甚至出现不少恶意软件,导致一些用户受到损失。同时Android没有对各厂商在硬件上进行限制,导致一些用户在低端机型上体验不佳。另一方面,因为Android的应用主要使用Java语言开发,其运行效率和硬件消耗一直是其他手机用户所诟病的地方。

### 1.2.2 常见的Android设备

因为Android系统的免费和开源,也因为系统本身强大的功能性,使得Android系统不仅被用于手机设备上,而且也被广泛用于其他智能设备中。在接下来的内容中,将简要介绍除了手机产品之外,常见的搭载Android系统的智能设备。

#### 1. Android智能电视

Android智能电视,顾名思义是搭载了Android操作系统的电视,使得电视智能化,能让电视机实现网页浏览、视频电影观看、聊天、办公、游戏等,与平板电脑和智能手机一样的功能。其凭借Android系统让电视实现智能化的提升,数十万款Android市场的应用、游戏等内容随意安装。

## 2. Android机顶盒

Android机顶盒是指像智能手机一样,具有全开放式平台,搭载了Android操作系统,可以由用户自行安装和卸载软件、游戏等第三方服务商提供的程序,通过此类程序来不断对电视的功能进行扩充,并可以通过网线、无线网络来实现上网冲浪的新一代机顶盒总称。

通过使用Android机顶盒,可以让电视具有上网、看网络视频、玩游戏、看电子书、听音乐等功能,使电视成为一个低成本的平板电脑。Android机顶盒不仅仅是一个高清播放器,更具有一种全新的人机交互模式,既区别于电脑、又有别于触摸屏。Android机顶盒配备红外感应条,遥控器一般采用空中飞鼠,这样就可以方便地实现触摸屏上的各种单点操作,可以方便地在电视上玩愤怒的小鸟、植物大战僵尸等经典游戏。例如乐视公司的LeTV机顶盒便是基于Android打造的,如图1-1所示。

## 3. 游戏机

Android游戏机就像Android智能手表一样,在2013年出现了爆炸式增长。在CES展会上,NVIDIA的Project Shield掌上游戏主机以绝对震撼的姿态亮相,之后又有Ouya和Gamestick相继推出。不久前, Mad Catz也发布了一款Andriod游戏机。

## 4. 智能手表

智能手表是将手表内置智能化系统、搭载智能手机系统而连接于网络而实现多功能,能同步手机中的电话、短信、邮件、照片、音乐等。

## 5. 智能家居

智能家居是以住宅为平台,利用综合布线技术、网络通信技术、智能家居-系统设计方案安全防范技术、自动控制技术、音视频技术将家居生活有关的设施集成,构建高效的住宅设施与家庭日程事务的管理系统,提升家居安全性、便利性、舒适性、艺术性,并实现环保节能的居住环境。

智能家居是在互联网影响之下的物联化体现。智能家居通过物联网技术将家中的各种设备(如音视频设备、照明系统、窗帘控制、空调控制、安防系统、数字影院系统、网络家电以及三表抄送等)连接到一起,提供家电控制、照明控制、窗帘控制、电话远程控制、室内外遥控、防盗报警、环境监测、暖通控制、红外转发以及可编程定时控制等多种功能和手段。与普通家居相比,智能家居不仅具有传统的居住功能,还兼备建筑、网络通信、信息家电、设备自动化,集系统、结构、服务、管理为一体的高效、舒适、安全、便利、环保的居住环境,提供全方位的信息交互功能。帮助家庭与外部保持信息交流畅通,优化人们的生活方式,帮助人们有效安排时间,增强家居生活的安全性,甚至为各种能源费用节约资金。

上述智能设备只是冰山一角,随着物联网和云服务的普及和发展,将有更多的智能设备诞生。

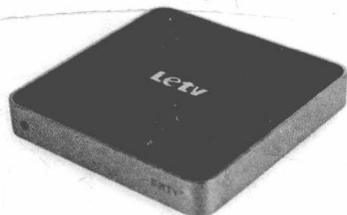


图1-1 基于Android的LeTV机顶盒

# 1.3 Android系统架构

Android系统是一个移动设备的开发平台,其软件层次结构包括操作系统(OS)、中间件(Middle Ware)和应用程序(Application)。根据Android的软件框图,其软件层次结构自下而上依次分为以下4层。

- (1) 操作系统层(OS)。
- (2) 各种库(Libraries)和Android运行环境(RunTime)。
- (3) 应用程序框架(Application Framework)。
- (4) 应用程序(Application)。

上述各个层的具体结构如图1-2所示。

在本节的内容中,将详细讲解Android系统各个层次的基本知识。



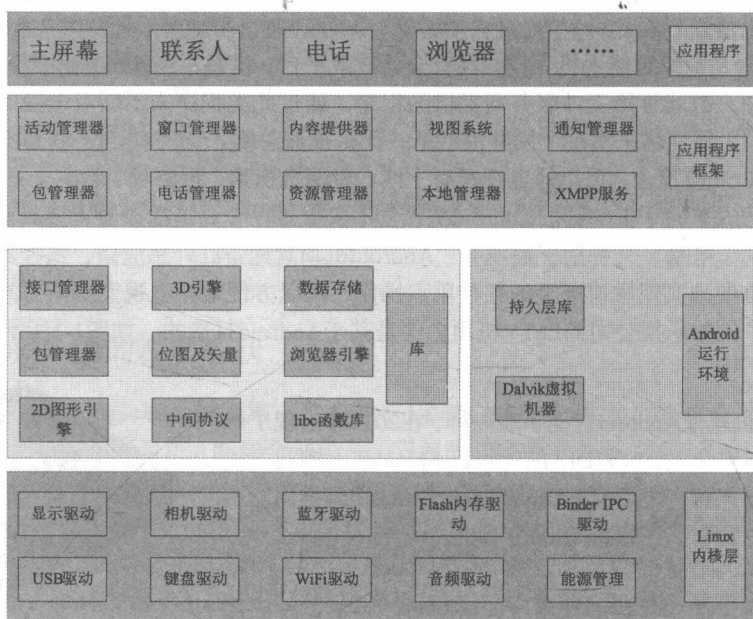


图1-2 Android操作系统的组件结构图

### 1.3.1 最底层的操作系统层（OS）——C/C++实现

Android系统的底层内核基于Linux操作系统,当前最新版本的Android的核心为标准Linux 3.10内核。Android底层的操作系统层（OS）使用C和C++语言编写实现,其实Android系统就是Linux系统,只是Android系统充分利用了已有的机制,尽量使用标准化的内容,如驱动程序,并且做出必要的扩展。Android灵活充分使用了内核到用户空间的接口,这主要表现在字符设备节点、Sys文件系统、Proc文件系统和不增加系统调用。

在Android系统中,包含的内核组件如下所示:

- Binder驱动程序（用户IPC机制）;
- Logger驱动程序（用户系统日志）;
- timed\_output驱动框架;
- timed\_gpio驱动程序;
- lowmemorykill组件;
- ram\_console组件;
- Ashmem驱动程序;
- Alarm驱动程序;
- pmem驱动程序;
- ADB Garget驱动程序;
- Android Paranoid网络。

### 1.3.2 Android的硬件抽象层——C/C++实现

其实Android生态系统的架构十分清晰,自下而上经典的模型分别为:Linux驱动、Android硬件抽象层、Android本地框架、Android的Java框架、Android的Java应用程序。因为Android系统需要运行于在不同的硬件平台上,所以需要具有很好的可移植性。其中Android系统的硬件抽象层负责建立Android系