

国内首本汇集底层驱动、中间层JNI制作和上层UI接口设计于一体的著作。通过丰富的实验程序、完整的项目实例，帮助你快速提高嵌入式硬件与Android的开发能力。



单片机与嵌入式



Android Embedded System Development (based on Cortex-A8)

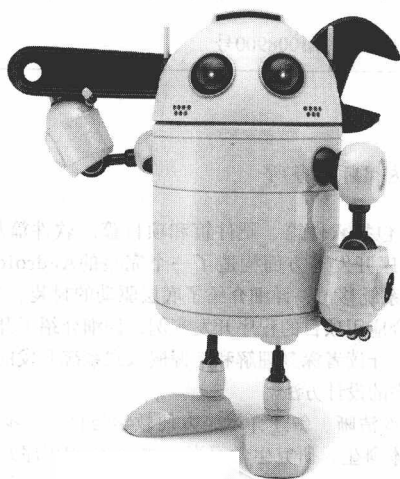
Android 嵌入式系统程序开发 (基于Cortex-A8)

胡文 宁世勇 李明俊 金雪松 编著



机械工业出版社
China Machine Press

单片机与嵌入式



Android Embedded System Development (based on Cortex-A8)

Android 嵌入式系统程序开发 (基于Cortex-A8)

胡文 宁世勇 李明俊 金雪松 编著



机械工业出版社
China Machine Press

图书在版编目(CIP)数据

Android嵌入式系统程序开发(基于Cortex-A8)/胡文, 宁世勇, 李明俊, 金雪松编著. —北京: 机械工业出版社, 2013.2

(单片机与嵌入式丛书)

ISBN 978-7-111-41169-7

I. A… II. ①胡… ②宁… ③李… ④金… III. 移动终端—应用程序—程序设计 IV. TN929.53

中国版本图书馆CIP数据核字(2013)第008900号

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问 北京市展达律师事务所

本书主要分为三部分, 包括软件篇、硬件篇和项目篇。软件篇从Android 体系结构和开发环境搭建, 以及Android 应用程序开发等方面构造了一个完整的Android 开发流程; 硬件篇重点讲解了S5PV210硬件结构和Android系统移植, 详细介绍了底层驱动的封装、中间层jni 的制作, 以及上层UI的设计; 项目篇提供完整的综合应用项目的程序开发实例, 详细介绍了开发过程和原始程序代码。并且安排了丰富的实验内容与实践, 让读者深刻理解和掌握嵌入式系统开发的整个过程, 了解底层驱动程序驱动硬件的原理和上层应用程序的设计方法。

本书内容丰富实用、层次清晰、叙述详尽, 方便教学与自学。本书可作为高等院校计算机类、电子类和控制类专业高年级本科生、研究生学习嵌入式Android程序开发的教材, 也可作为全国大学生电子设计竞赛培训教材, 以及工程技术人员进行嵌入式系统开发与应用的参考书。

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑: 王 彬 秦 健

三河市杨庄长鸣印刷装订厂印刷

2013年4月第1版第1次印刷

186mm×240mm·28印张

标准书号: ISBN 978-7-111-41169-7

定 价: 69.00元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991; 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294; 88379649; 68995259

读者信箱: hzjsj@hzbook.com

前言

本书的特点是以嵌入式操作系统Android和Cortex-A8微处理器S5PV210为基础,从Android体系结构和搭建开发环境,以及Android应用程序开发等方面构造了一个完整的Android开发流程。本书重点介绍了S5PV210硬件结构和Android系统移植,详细讲解了底层驱动的封装、中间层jni的制作,以及上层UI的设计,采用Java开发接口驱动程序。本书还提供了完整的综合应用项目的程序开发实例,介绍了功能模块设计和数据库设计,详细讲解了开发过程和原始程序代码;使读者深刻理解和掌握嵌入式系统开发的整个过程,了解底层驱动程序驱动硬件的原理和上层应用程序的设计方法,真正做到了底层驱动的开发与上层应用程序的开发相结合。

本书内容丰富实用、叙述详尽清晰,方便教学与自学。结合DMA-210XP平台的实验程序,有利于读者掌握Android系统的应用程序设计方法,培养读者综合分析、开发创新和工程设计的能力。通过本书的学习,读者可以快速提高Android的编程能力和实际开发水平。

全书分三部分,共9章。

第一部分:软件篇

- 第1章介绍了Android的体系结构,以及Android系统的整体架构的各层组成,并介绍了如何搭建Windows、Linux和NDK下的开发环境。
- 第2章介绍了第一个HelloEveryone的Android应用程序,以及Android应用程序组成,并介绍如何使用Android Manifest文件定义应用程序。
- 第3章介绍了Android应用程序开发过程中涉及的控件,介绍了Activity转换和Intent消息传递、Menu设计修改、对话框实例和Android本地数据库SQLite应用。

第二部分:硬件篇

- 第4章详细介绍了Android内核结构和设备驱动、Android内核基本配置、Android内核编译和DMA-210XP平台Android文件系统烧写过程。
- 第5章介绍了S5PV210的硬件结构,包括S5PV210微处理器、GPIO接口、PWM定时器、DMA控制器、UART串行接口、SPI接口、IIC总线接口和ADC及触摸屏接口。介绍了接口应用实例及驱动程序。
- 第6章介绍了在Android开发环境下,采用Java编写接口驱动程序。包括LED接口及驱动程序、背光调节控制程序、键盘接口及驱动程序、UART串行口及通信程序、ZigBee接口及驱动程序、Wi-Fi接口及通信程序、3G接口及驱动程序、MediaPlayer播放器程序等。

第三部分:项目篇

- 第7章介绍了嵌入式组态软件结构,以及界面设计、功能选择区设计、辅助功能区设计

和程序生成区设计Android应用，采用Java开发嵌入式组态软件。

- 第8章以MyMap服务系统为实例，介绍Android Google Map、Android定位服务、案例重构，采用Java开发GPS与Google Map定位应用程序。
- 第9章以3D传感器动力球游戏为例，介绍小球快跑游戏背景及功能、游戏的架构，详细讲解了游戏主菜单、游戏界面模块，游戏中各个图层、游戏后台逻辑和传感器计算模块等的程序开发。

本书由胡文教授拟定编写大纲和目录。胡文编写了第6章和第7章，宁世勇编写了第2章和第3章，李明俊编写了第1章、第8章和第9章，金雪松编写了第4章和第5章。百度公司的胡玥，哈尔滨商业大学的李杨、赵艳丽、张凯、陈楠等人为本书的编写做了大量的工作，在此一并表示衷心感谢。本书在编写过程中，参考了大量的国内外著作和资料，得到了许多专家和学者的大力支持，并听取了多方面的宝贵意见和建议，在此也对他们表示衷心感谢。

由于时间仓促和作者水平所限，本书难免有疏漏和不足之处，敬请各位读者批评指正，以期再版时修订。

编者

2013年1月

目 录

前言

软件篇

第1章 Android体系结构及开发环境 / 2

1.1 Android体系结构介绍 / 2

1.1.1 应用程序 / 3

1.1.2 应用程序框架 / 3

1.1.3 函数库 / 4

1.1.4 Linux内核 / 4

1.2 搭建Windows下的开发环境 / 5

1.2.1 安装JDK / 5

1.2.2 安装Eclipse / 6

1.2.3 安装ADT / 6

1.2.4 安装SDK / 7

1.2.5 创建Android虚拟设备 / 9

1.3 搭建Linux下的开发环境 / 11

1.3.1 安装JDK / 11

1.3.2 安装Eclipse / 12

1.3.3 安装ADT / 13

1.3.4 安装SDK / 15

1.3.5 创建Android虚拟设备 / 17

1.4 搭建NDK开发环境 / 20

1.4.1 NDK开发环境与安装目录 / 20

1.4.2 系统和软件需求 / 20

1.4.3 Windows 平台NDK环境
搭建 / 21

1.4.4 Linux平台NDK环境搭建 / 25

第2章 第一个应用程序 / 28

2.1 第一个HelloEveryone / 28

2.1.1 创建第一个Android
项目HelloEveryone / 28

2.1.2 Android应用工程组成 / 29

2.2 调试Android应用程序 / 30

2.2.1 修改Android项目 / 30

2.2.2 设置断点 / 34

2.2.3 启动调试 / 34

2.2.4 单步跟踪 / 34

2.2.5 真机调试 / 35

2.3 Android应用程序组成 / 35

2.3.1 Activity介绍 / 36

2.3.2 Intent介绍 / 40

2.3.3 Broadcast Receiver介绍 / 43

2.3.4 Service介绍 / 43

2.3.5 Content Provider介绍 / 44

2.4 使用AndroidManifest文件定义 应用程序 / 44

2.4.1 管理应用程序身份 / 46

2.4.2 注册Activity和其他应用
程序组件 / 46

- 2.4.3 使用许可权限 / 48
- 2.4.4 指定应用程序所需输入设备和软件 / 49
- 2.4.5 使用库和Android SDK版本 / 50
- 2.4.6 定义应用程序的其他配置参数 / 50

第3章 Android应用程序开发 / 51

- 3.1 控件简介 / 51
 - 3.1.1 视图控件 / 51
 - 3.1.2 布局控件 / 51
 - 3.1.3 布局参数 / 51
- 3.2 常用Widget控件介绍 / 52
 - 3.2.1 用Widget控件创建Android项目 / 52
 - 3.2.2 按钮 / 53
 - 3.2.3 文字框 / 54
 - 3.2.4 编辑框 / 55
 - 3.2.5 多项选择框 / 57
 - 3.2.6 单项选择框 / 59
 - 3.2.7 下拉列表 / 60
 - 3.2.8 自动完成文本 / 61
 - 3.2.9 日期、时间选择器 / 62
 - 3.2.10 进度条 / 64
 - 3.2.11 拖动条 / 65
 - 3.2.12 图片视图 / 66
 - 3.2.13 基于网格索引的图片浏览器 / 67
 - 3.2.14 选项卡 / 70
 - 3.2.15 列表 / 72
 - 3.2.16 图片按钮 / 73
 - 3.2.17 拖动效果 / 74

- 3.3 Activity转换和Intent消息传递 / 76
 - 3.3.1 Activity转换 / 76
 - 3.3.2 Intent消息传递 / 78
- 3.4 Menu设计修改 / 80
- 3.5 对话框实例 / 82
- 3.6 Toast和Notification应用 / 85
- 3.7 数据库应用 / 87
 - 3.7.1 SQLite数据库介绍 / 87
 - 3.7.2 Android平台对SQLite数据库的支持 / 88
 - 3.7.3 SQLite数据库应用模式 / 89
 - 3.7.4 SQLite数据库开发实例 / 90
 - 3.7.5 基于SQLite数据库的日记账工具 / 97

硬件篇

第4章 Android系统移植 / 112

- 4.1 Android 结构介绍 / 112
- 4.2 Android 内核结构和设备驱动 / 113
 - 4.2.1 Android内核源代码结构 / 113
 - 4.2.2 Android常用设备驱动 / 115
- 4.3 Android 内核基本配置 / 121
 - 4.3.1 Android内核中的Kconfig文件 / 122
 - 4.3.2 Android内核配置选项 / 122
- 4.4. Android内核编译 / 125
 - 4.4.1 Android内核中的Makefile文件 / 126
 - 4.4.2 解压Android内核源代码 / 127
 - 4.4.3 编译Android 内核 / 127

- 4.5 DMA-210XP平台Android文件系统
烧写 / 128
 - 4.5.1 烧写u-boot到Nand Flash / 128
 - 4.5.2 烧写zImage内核映像
文件 / 131
 - 4.5.3 烧写ramdisk-uboot.img
映像文件 / 132
 - 4.5.4 烧写Android System.img
文件 / 132
 - 4.5.5 烧写Android userdata.img
文件 / 133
 - 4.5.6 启动Android系统 / 133
 - 4.5.7 TF卡自动更新kernel和
Android系统文件 / 134

第5章 S5PV210硬件结构 / 135

- 5.1 S5PV210微处理器 / 135
 - 5.1.1 概述 / 135
 - 5.1.2 S5PV210体系结构 / 135
 - 5.1.3 S5PV210关键特性 / 135
- 5.2 GPIO接口 / 146
 - 5.2.1 GPIO概述 / 146
 - 5.2.2 GPIO专用寄存器 / 147
 - 5.2.3 GPIO接口应用举例 / 154
- 5.3 PWM定时器 / 156
 - 5.3.1 PWM概述 / 156
 - 5.3.2 PWM操作 / 156
 - 5.3.3 PWM专用寄存器 / 158
 - 5.3.4 PWM接口应用举例 / 162
- 5.4 DMA控制器 / 164
 - 5.4.1 DMA概述 / 164
 - 5.4.2 DMA专用寄存器 / 165
- 5.5 UART串行接口 / 169

- 5.5.1 UART接口特性 / 169
- 5.5.2 UART操作 / 170
- 5.5.3 UART专用寄存器 / 173
- 5.5.4 UART接口应用举例 / 179
- 5.6 SPI接口 / 183
 - 5.6.1 SPI接口特性 / 183
 - 5.6.2 SPI操作 / 183
 - 5.6.3 SPI专用寄存器 / 185
 - 5.6.4 SPI接口应用举例 / 190
- 5.7 IIC总线接口 / 194
 - 5.7.1 IIC概述 / 194
 - 5.7.2 IIC操作 / 194
 - 5.7.3 IIC专用寄存器 / 198
 - 5.7.4 IIC接口应用举例 / 200
- 5.8 ADC及触摸屏接口 / 203
 - 5.8.1 ADC及触摸屏特性 / 204
 - 5.8.2 功能描述 / 205
 - 5.8.3 ADC及触摸屏专用
寄存器 / 206
 - 5.8.4 ADC转换应用举例 / 210

第6章 接口驱动程序开发 / 212

- 6.1 LED接口及驱动程序 / 212
 - 6.1.1 LED驱动电路 / 212
 - 6.1.2 LED驱动程序分析 / 215
 - 6.1.3 LED驱动程序Makefile
文件 / 218
 - 6.1.4 LED驱动测试 / 218
 - 6.1.5 LED界面设计 / 219
 - 6.1.6 LED JNI设计 / 225
 - 6.1.7 LED程序测试 / 228
- 6.2 背光调节控制程序 / 230
 - 6.2.1 背光控制接口 / 230

- 6.2.2 背光驱动设计 / 233
- 6.2.3 背光界面设计 / 235
- 6.2.4 背光JNI设计 / 241
- 6.2.5 背光程序测试 / 243
- 6.3 键盘接口及驱动程序 / 244
 - 6.3.1 键盘接口 / 244
 - 6.3.2 内核驱动 / 249
 - 6.3.3 键盘界面设计 / 253
 - 6.3.4 键盘程序测试 / 258
- 6.4 UART串行接口及通信程序 / 258
 - 6.4.1 UART串行接口 / 258
 - 6.4.2 Linux串行接口操作 / 262
 - 6.4.3 UART界面设计 / 263
 - 6.4.4 UART JNI设计 / 272
 - 6.4.5 UART程序测试 / 275
- 6.5 ZigBee接口及通信程序 / 276
 - 6.5.1 ZigBee介绍 / 276
 - 6.5.2 ZigBee接口设计 / 277
 - 6.5.3 ZigBee JNI设计 / 291
 - 6.5.4 ZigBee程序测试 / 296
- 6.6 Wi-Fi接口及通信程序 / 297
 - 6.6.1 Wi-Fi介绍 / 297
 - 6.6.2 Wi-Fi程序设计 / 300
 - 6.6.3 Wi-Fi程序测试 / 307
- 6.7 3G接口及通信程序 / 307
 - 6.7.1 3G介绍 / 308
 - 6.7.2 3G程序设计 / 310
 - 6.7.3 3G程序测试 / 315
- 6.8 MediaPlayer播放器程序 / 317
 - 6.8.1 MediaPlayer介绍 / 317
 - 6.8.2 MediaPlayer程序设计 / 320
 - 6.8.3 MediaPlayer程序测试 / 327

项目篇

第7章 Android嵌入式组态软件 / 330

- 7.1 Android嵌入式组态软件介绍 / 330
- 7.2 Android嵌入式组态软件结构 / 330
- 7.3 Android嵌入式组态软件界面设计 / 332
 - 7.3.1 界面设计 / 332
 - 7.3.2 界面内容添加 / 332
- 7.4 Android嵌入式组态软件功能选择区设计 / 339
 - 7.4.1 界面设计 / 340
 - 7.4.2 IO设计 / 352
 - 7.4.3 数据库设计 / 354
- 7.5 Android嵌入式组态软件辅助功能区设计 / 359
- 7.6 Android嵌入式组态软件程序生成区设计 / 360
 - 7.6.1 Android嵌入式组态软件程序生成区功能介绍 / 360
 - 7.6.2 Android嵌入式组态软件程序生成区的代码实现 / 361

第8章 GPS与Google Map定位系统 / 374

- 8.1 MyMap服务系统 / 374
- 8.2 Android Google Map / 375
 - 8.2.1 申请Google Map Android API Key / 375
 - 8.2.2 编写Google Map框架程序 / 376

- 8.2.3 控制地图 / 378
- 8.2.4 地图的显示模式 / 380
- 8.2.5 地图的图层 / 383
- 8.2.6 查询与定位 / 385
- 8.3 Android定位服务 / 391
 - 8.3.1 开启定位服务 / 391
 - 8.3.2 模拟测试 / 393
 - 8.3.3 GPS与GoogleMap
定位 / 395
- 8.4 案例重构 / 396
 - 8.4.1 地图的显示模式 / 396
 - 8.4.2 重构“查询方法” / 398

第9章 3D物理传感器游戏——小球 快跑 / 401

- 9.1 游戏背景及功能概述 / 401
 - 9.1.1 背景概述 / 401
 - 9.1.2 功能概述 / 401
- 9.2 游戏的策划及准备工作 / 403
 - 9.2.1 游戏的策划 / 403
 - 9.2.2 小球快跑游戏开发的准备
工作 / 403
- 9.3 游戏的架构 / 404
 - 9.3.1 游戏的总体架构 / 404
 - 9.3.2 游戏的类结构 / 405
- 9.4 DriftBall类的开发 / 406
 - 9.4.1 DriftBall类的代码框架 / 406
 - 9.4.2 DriftBall类的主要成员
方法的实现 / 407
- 9.5 游戏主菜单的开发 / 409

- 9.5.1 WelcomeView类的代码
框架 / 409
- 9.5.2 WelcomeView类的主要
成员方法的实现 / 410
- 9.5.3 WelcomeThread类的
开发 / 412
- 9.5.4 菜单界面的用户交互
事件处理 / 413
- 9.6 游戏界面模块的开发 / 414
 - 9.6.1 GameView的成员变量 / 414
 - 9.6.2 GameView的成员方法
概述 / 415
 - 9.6.3 GameThread类的代码
框架 / 416
 - 9.6.4 GameMenuThread类的
开发 / 417
 - 9.6.5 用户交互事件处理 / 418
- 9.7 游戏中各个图层的开发 / 419
 - 9.7.1 地图图层的开发 / 419
 - 9.7.2 其他图层的开发与
实现 / 421
- 9.8 游戏后台逻辑的开发 / 424
 - 9.8.1 小球的运动控制 / 424
 - 9.8.2 小球的碰撞检测 / 425
 - 9.8.3 Cannon和Missile类的
开发 / 428
- 9.9 传感器计算模块的开发 / 432
 - 9.9.1 BallListener类的开发 / 432
 - 9.9.2 RotateUtil类的代码框架 / 433
 - 9.9.3 RotateUtil类的开发 / 435

参考文献 / 438

软 件 篇

- ◆ 第1章 Android体系结构及开发环境
- ◆ 第2章 第一个应用程序
- ◆ 第3章 Android应用程序开发

第1章 Android体系结构及开发环境

1.1 Android体系结构介绍

Android是Google公司为移动设备开发的平台，它是一款开放的软件系统，其系统体系结构如图1-1所示，自上而下分为以下几个层次。

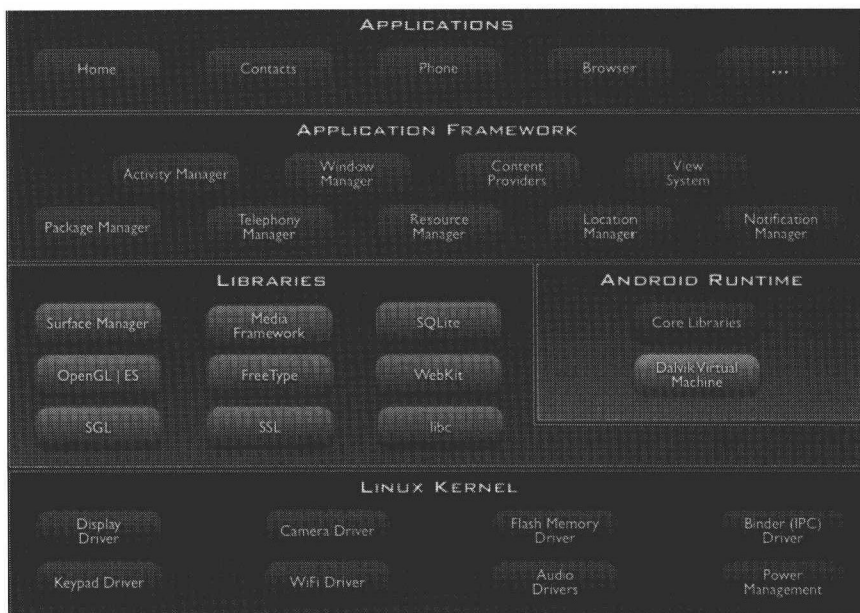


图1-1 Android系统体系结构

- 应用程序（Application）
- 应用程序框架（Application Framework）
- 函数库（Libraries）和Android运行时（Android Runtime）
- Linux内核（Linux Kernel）

Android的内核系统服务依赖于Linux2.6内核，随着Android 发布版本的升级，Android 系统使用的Linux内核也在不断升级，下面仅列举到Android 2.3版本。与Android 系统对应的内核及其版本如下：

Android	系统版本内核版本
Android 1.0	Linux 2.6.25
Android 1.5 (Cupcake)	Linux 2.6.27
Android 1.6 (Donut)	Linux 2.6.29
Android 2.1 (Eclair)	Linux 2.6.29
Android 2.2 (Froyo)	Linux 2.6.32
Android 2.3 (GingerBread)	Linux 2.6.35

在本节中将对每一层次进行简单说明。

1.1.1 应用程序

Android应用程序是基于Java语言编写的，为使用者提供操作接口。使用者直接操作应用程序，实现一定的功能。目前Android系统提供了计算器、联系人（Contacts）、电话（Phone）、浏览器（Browser）、E-mail客户端、SMS短消息程序、日历、地图等内核应用程序，开发者还可以使用Android提供的组件编写满足特定功能的应用程序。

由用户开发的Android应用程序和Android内核应用程序是同一层次的，它们都是基于Android系统的API构建的。

1.1.2 应用程序框架

开发人员可以访问内核应用程序所使用的API框架。应用程序体系结构设计简化了组件的重用，任何一个应用程序都可以发布其功能块，并且任何其他的应用程序都可以使用应用程序体系结构所发布的功能块（不过要遵循框架的安全性限制）。同样，应用程序重用机制也使使用者可以方便地替换程序组件。隐藏在每个应用后面的是一系列的系统服务，这些系统服务包括：

- 丰富且可扩展的视图（View System），可以用来构建应用程序，这些视图包括列表（List）、网格（Grid）、文本块（Text Box）、按钮（Button），甚至可嵌入Web浏览器。
- 内容提供者（Content Provider）使应用程序可以访问另一个应用程序的数据（如联系人数据库），或者共享它们自己的数据。
- 资源管理器（Resource Manager）提供非程序代码资源的访问，如本地字符串、图形和布局文件（Layout File）。
- 通知管理器（Notification Manager）使得应用程序可以在状态列中显示自定义的提示信息。
- 活动管理器（Activity Manager）用来管理应用程序生命周期并提供常用的导航回退功能。
- 窗口管理器（Window Manager）管理所有窗口程序。
- 包管理器（Package Manager）管理Android系统内的程序。
- 通信管理器（Telephony Manager）管理Android系统的通信功能。
- 定位管理器（Location Manager）提供Android系统的定位等相关服务。

1.1.3 函数库

Android包含一些C/C++函数库，这些函数库能被Android系统中不同的组件使用。它们通过Android应用程序框架为开发者提供服务。这些内核函数库包括：

- **libc**：标准C系统函数库，它是专门为基于Embedded Linux 的设备定制的。
- **Media Framework**：基于PacketVideo OpenCORE，该函数库支持多种常用的音效、视频格式回放和录制，同时支持静态影像文件。编码格式包括MPEG4、H.264、MP3、AAC、AMR、JPG以及PNG。
- **Surface Manager**：显示子系统的管理，并且为多个应用程序提供了2D和3D图层的无缝融合。
- **Webkit**：提供Web浏览引擎的支持。
- **SGL**：底层的2D图形引擎。
- **OpenGL ES**：基于OpenGL ES 1.0 APIs实现，该函数库可以使用硬件3D加速（如果可用）或者使用高度优化的3D软加速。
- **FreeType**：位图（bitmap）和向量（vector）字体显示。
- **SQLite**：一个对于所有应用程序可用、功能强大的轻量级关系型数据库引擎。
- **SSL**：安全套接层（Secure Sockets Layer）是为网络通信提供安全及数据完整性的一种安全协议。

1.1.4 Linux内核

Android 的内核系统服务如安全管理、内存管理、进程管理、网络通信和驱动模型依赖于Linux 2.6内核，Linux内核也同时作为硬件和软件堆栈之间的抽象层。关于Android对Linux内核的修改，这里重点阐述以下两个：

- **Binder（IPC）Driver**：提供高效率的进程间通信（Inter-Process Communication）。Android系统中有很多服务，上层的应用程序经常要取用这些服务。虽然一般的Linux 系统已经提供了很多IPC的方式，但是Android几乎重新制作了一套自己的IPC。Android文件中解释说，一般IPC会造成额外资源花费以及安全问题。
- **Power Management**：与台式计算机或笔记本电脑不同，手持设备的电源一向相当有限，必须想尽一切办法省电，而又不能影响顺畅的使用体验。Android在此采取了颇为积极的做法：如果不使用，就关掉。例如，某程序在播放MP3音乐，于是此程序需要CPU的计算能力，那系统就得提供。如果与此同时没有执行其他程序，那么LCD显示器就可能被关闭，以便省电。一般的Linux内核考虑的都是计算机上的做法，所以多数只有进入暂停、休眠等选择，而不会如此细致地控制各个小装置的电源供应。

以上详细介绍了Android体系结构情况，读者可以了解了Android体系结构的基本情况，为开发Android应用程序打下基础。在开发Android应用程序之前首先需要搭建开发的环境，下面将介绍Android应用程序开发环境是如何搭建的。

1.2 搭建Windows下的开发环境

在搭建Windows下的开发环境之前，必须先下载Android SDK组件、Eclipse以及JDK（Java Development Kit）组件（因为Android的应用程序是采用Java语言编写的）。

1.2.1 安装JDK

安装Eclipse的开发环境需要JRE的支持，如果没有JRE，则启动Eclipse时会报告错误。在Windows XP上安装JRE/JDK非常简单，步骤如下：

1) 从Sun公司官方网站下载最新版的JDK。

2) 双击JDK安装文件，打开安装向导，然后按照默认的设置点击“下一步”进行安装。默认情况下JDK的安装路径为：C:\Program Files\Java（这里以C盘为系统盘）。安装完成后还需设定JDK的环境变量，将JDK的bin文件的路径C:\Program Files\Java\jdk1.6.0_14\bin添加到path中。具体做法是：右击“我的电脑”，点击“属性”，打开“系统属性”对话框，在其“高级”选项卡中点击“环境变量”按钮，打开“环境变量”对话框，选择“系统变量”列表中的“Path”选项，如图1-2所示。

点击“编辑”按钮，添加JDK目录中bin文件所在路径即可，末尾要以半角的分号结尾，如图1-3所示。

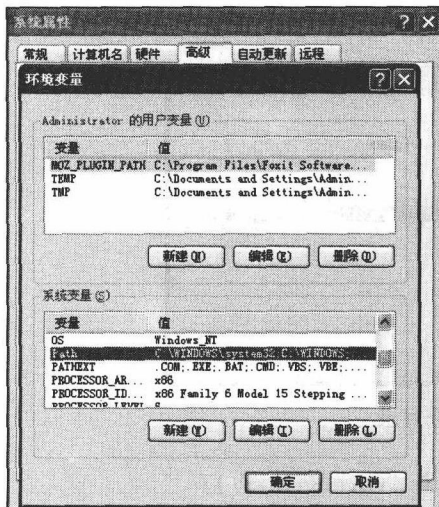


图1-2 环境变量配置1

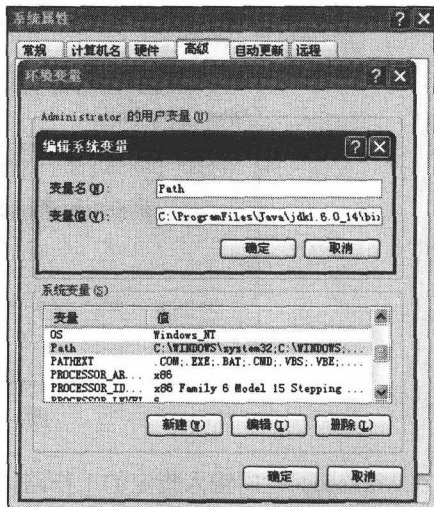


图1-3 环境变量配置2

配置完成后，点击“确定”按钮退出。

3) 下面测试安装是否成功。点击“开始”→“运行”，在出现的窗口中输入“cmd”命令，在弹出的命令窗口中输入命令：java -version，如果出现图1-4所示信息，则说明JDK安装成功。

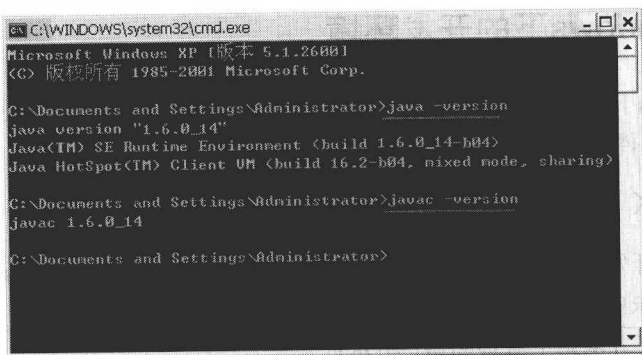


图1-4 检测配置信息

1.2.2 安装Eclipse

JDK安装完成之后,就可以安装Eclipse了。安装步骤如下:

- 1) 从Eclipse官方网站下载最新版的Eclipse。
- 2) 下载完成之后,解压Eclipse压缩包文件,然后进入解压目录,可以看到一个名为eclipse.exe的可执行文件。
- 3) 双击eclipse.exe文件,即可直接运行Eclipse。如果是第一次启动Eclipse,将会看到一个选择工作目录(Workspace)的提示,如图1-5所示。

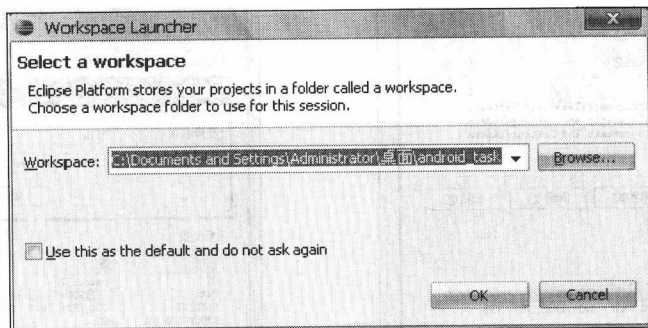


图1-5 选择工作目录

在Workspace栏内输入指定的工作路径,然后点击“OK”按钮即可。

1.2.3 安装ADT

成功安装Eclipse之后,还需要安装ADT开发工具。ADT用于为Eclipse打造一个Android专属的开发环境,包括创建Android开发实例、执行和调试Android程序。

- 1) 首先下载ADT插件,与Android 2.1对应的版本为ADT-0.9.6。ADT下载网址为: <http://androidappdocs.appspot.com/sdk/eclipse-adt.html>。

2) 下载完成后, 启动Eclipse, 在Eclipse界面内点击菜单栏中的“Help”→“Software Updates”。如图1-6所示。

3) 进入Software Updates and Add-ons窗口, 其中有两个选项卡, 分别是Installed Software和Available Software, 如图1-7所示。

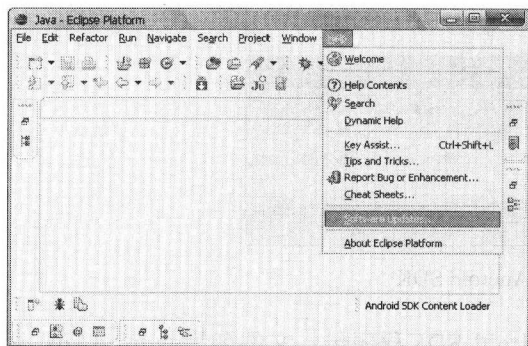


图1-6 选择“Software Updates”命令

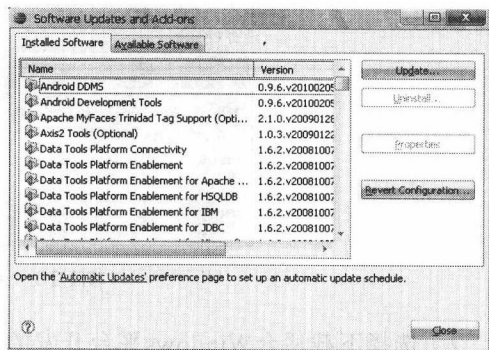


图1-7 Software Updates and Add-ons窗口

4) 由于已经得到了ADT文件, 所以选择Available Software选项卡直接安装即可, 点击“Add Site”按钮, 并在Add Site对话框中输入存放ADT的路径, 如图1-8所示。

5) 点击“OK”按钮退出对话框。然后在Available Software选项卡中找到Developer Tools, 勾选其下面的两个选项, 如图1-9所示。

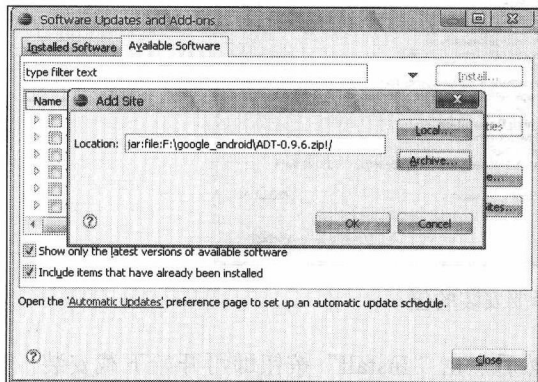


图1-8 Add Site对话框

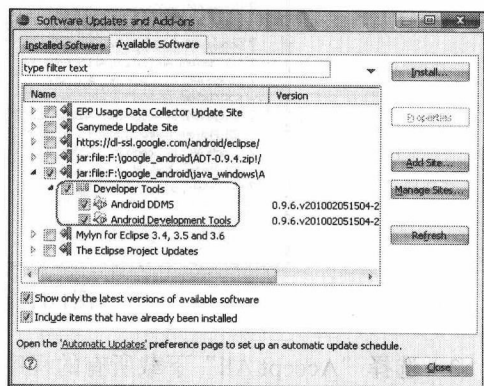


图1-9 勾选Developer Tools下面的两个选项

6) 确认操作无误后点击“Install”按钮进行安装。这里最好使用默认设置, 安装完成后重新启动Eclipse。

1.2.4 安装SDK

1) Android SDK的官方网站为<http://androidappdocs.appspot.com/sdk/>, 可以从该网站下载最