

源码解读，对Android底层源码进行系统剖析。

硬件实践，基于开源硬件pcDuino和Xilinx最新的可编程SoC——Zynq。

专家力推，谷歌、华为、CSDN社区专家联袂推荐。

移动开发



Underlying Software Development and Practice of Android

Android 底层 开发实战

周庆国 郑灵翔 康筱彬 刘同山◎编著



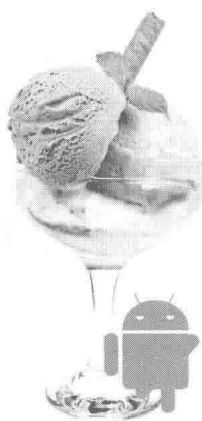
机械工业出版社
China Machine Press

移动开发

Underlying Software Development and Practice of Android

Android底层 开发实战

周庆国 郑灵翔 康筱彬 刘同山◎编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Android 底层开发实战 / 周庆国等编著. —北京: 机械工业出版社, 2015.9
(电子与嵌入式系统设计丛书)

ISBN 978-7-111-51611-8

I. A… II. 周… 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 (2015) 第 225084 号

本书主要面向对 Android 设备驱动开发有浓厚兴趣的人员, 无论是具有一定经验的开发人员还是初学者, 都能根据本书所提供的案例进行学习。

书中介绍了嵌入式系统的定义、Linux 系统和 Android 开发环境搭建等基础知识; 分析了 Android 系统底层源码和内核结构, 介绍了 init、Zygote、Binder、Ashmem、Low Memory Killer 和 Logger 等模块; 剖析了系统开发工具 Dalvik、JNI、Boot Loader 的原理和工作方式; 对驱动程序设计中 NDK 的编程方法以及 HAL 层的调用进行了详细阐述, 并通过 Camera 与 WiFi 驱动的实现进行进一步的说明。书中理论部分的介绍旨在呈现一个清晰的开发索引, 同时, 也有利于读者后续进行深度开发。

本书一大特色是具有针对不同架构的实例, 涵盖了当前较流行的 ZedBoard、pcDuino、BeagleBoard 三类开发板。通过对三类开发板实施系统移植、驱动设计、内核跟踪预测 3 个项目, 使得读者对书中所涉及的所有知识都能有更加深入的了解和应用。

Android 底层开发实战

出版发行: 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: 张国强

责任校对: 董纪丽

印刷: 北京市荣盛彩色印刷有限公司

版次: 2015 年 10 月第 1 版第 1 次印刷

开本: 186mm × 240mm 1/16

印张: 16

书号: ISBN 978-7-111-51611-8

定价: 59.00 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) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

推 荐 序

自从2008年谷歌公司发布了Android系统之后，全球便开启了智能机的研究热潮。与此同时，谷歌联合OHA（手机开放联盟）共同推进Android智能机的发展。Android系统的开源及开放等特性，不仅改变了手机移动智能行业，也深深影响了其他科技产业，如嵌入式物联网等，从而掀起了新的一波移动浪潮。

如今市面上介绍Android的书籍比较多，但从底层入手由浅入深地介绍Android开发的比较少。而本书正是从底层的角度出发，引导大家重新审视Android的控件，深入挖掘Android中底层的内容。当然，本书也不是什么大全之作，并没有深入剖析每个知识点，不过应对一般的开发应该足够了。

本书适合有一定编程基础（至少了解C/C++或Java）并对Android开发比较感兴趣的读者。本书首先从Android嵌入式系统及Linux开发常用基础入手，详细介绍Android的系统开发环境及源码结构，并对内核及相关环境库进行讲解，接着讲述相对比较高级的Android驱动开发设计，最后通过一系列开发实例将之前的内容融会贯通，从而让读者循序渐进、由浅入深地学习Android系统的底层开发知识。

学完本书后，读者可以在实际开发中自然而然地体会并运用所学知识，知道如何发现和解决问题以及为什么这样做。希望大家能够从本书中有所收获，从而对工作、学习或者研究起到帮助作用。

谷歌大学合作部安卓培训课程主讲人 郭国勇

2015年8月

自序

随着4G网络在全国范围内大规模部署，以及各种无线热点如雨后春笋般在城市遍地开花，甚至在乡村开枝散叶，无线宽带变得越来越普遍，这使得移动设备的应用场景变得更加丰富多彩，人与人之间能够更加方便地随时随地通过多种方式进行沟通，文字、图片已经不再新鲜，视频通信、视频点播也不在话下。智能手机在性能、功能上不断提升，而价格却越来越亲民。如今，在国内大部分人都有一部智能手机。手机犹如钱包、钥匙一样，已经成为每个人随身携带的必备物。智能手机已经不仅仅是一部移动电话，更是一个移动平台，既能当作语音、视频的通信平台，又能当作游戏、娱乐的平台，甚至还可以在手机上处理业务。

如今，传统的互联网已经转向移动互联网，以前还是PC之间的互联，如今变成了移动设备之间的互联。而“万物联网”的概念也已经炒得非常火热，不管是英特尔、三星、高通这样的集成芯片巨头，还是像微软这样的软件巨头，都在推出自己相应的软硬件产品，以趁这个发展势头在物联网时代占得自己的一席之地。将来，所有的“物”都会更加智能，能够更好地与人互动，与其他“物”通信。那时的网络会成为一个人与人、人与物、物与物之间有着万亿连接的庞大网络。

而在物联网时代，移动设备，尤其是智能手机，势必会成为一个核心的接入点，去连接和感知周围环境中的一切。比如通过连接家里的空调、电视机、空气净化器等家用电器，手机会告知甚至自动为用户调节家里的温度和湿度，或控制家里的电视、灯具；通过连接身上携带的传感器并跟踪用户的身体参数，如心跳、呼吸、运动量等，手机可以告知健康状况并给出专业的建议。此外，智能手机也会成为车联网的一大重要接入点，可以远程开启车门，远程反馈汽车的整体状态。总而言之，在万物联网的时代，智能手机会扮演举足轻重的角色。

2007年11月5日，Google推出开放的Android操作系统，同时宣布建立一个全球性的联盟组织，该组织由34家手机制造商、软件开发商、电信运营商以及芯片制造商共同组成。这一联盟将支持Google发布的手机操作系统以及应用软件，将共同开发Android系统的开放源代码。其他公司和个人也可以免费获取源代码并进行SDK的开发。正是因为Android的开放性以及优异的性能，使它得到了众多厂商的支持。前有三星、摩托罗拉、HTC、索尼，后有异军

突起的小米、联想、华为等，纷纷推出了自己定制的Android平台手机。国际研究及顾问机构Gartner公布的最新统计数据表明2014年全球智能手机销售量总计12亿部。另据市场调研公司IDC提供的数据，2014年全年，Android全球智能机所占市场份额为81.5%；其次才是苹果公司的iOS，占14.8%。众星拱月式的拥护，使Android在短短5年内超越iOS以及曾经辉煌一时的塞班系统，顺利雄踞智能手机市场占有率榜首。

不管从功能、性能上，还是从市场占有率及开发者数量上，都说明Android会成为物联网时代一个重要的操作系统。

对于Android底层开发者来说，需要面对复杂多变的应用场景、纷繁的外围设备、不同标准的互联通信，这些无疑都是巨大的挑战。这就要求底层开发者除了掌握软件编程，还要熟悉底层的硬件设备，需要在实践中锻炼自己的技能。目前在市场上也出现了兼容Android操作系统的开源硬件平台，比如BeagleBone、pcDuino等，这对于很多爱好Android的创客来说，是一大利好。就像开源软件Linux和开源硬件Arduino，正是依靠庞大的社区，才使其能够在全世界得到飞快的传播。庞大的社区，不管是对于开发者本身的学习和开发过程，还是对于Android的发展，都会带来积极的促进作用。

在编写本书的过程中，我们与创客社团和Google开发者社区（GDG）的成员沟通时发现，无论是初学者还是开发者，都偏重于软件而缺乏硬件基础，所以非常希望更深入地了解底层的硬件。而且，兼容Android的外围设备越来越多，也迫使底层开发人员去掌握基本的硬件知识和底层驱动开发。

因此，在Google大学合作部的大力支持下，几位从事嵌入式开发的教师合作编写了本书。本书在编写过程中也融入了几位老师的研究课题以及教学经验，希望能对从事和学习Android底层开发的人员有所帮助。

编者

2015年8月

前 言

自第一款搭载Android系统的智能手机HTC G1发布至今已近6年，凭借日益完善与强大的功能、完全开放的内核源码以及Google公司在网络应用领域的无缝支持，Android系统从初出茅庐的行业新锐，已经成长壮大为手持设备智能化产业的推动者和市场的领导者，这一点已经毋庸置疑。

不可否认，众多独立软件开发者与商业软件公司的加入使Android系统不仅仅作为终端产品，还作为开发平台深入到各行业的技术开发中，包括学校、公司，乃至开发人员的工作、生活的各方面，这是Android系统旺盛生命力的表现，同时也是Android系统持续发展壮大的保障。近期，支持Android系统的开发板如Zynq、pcDuino等在功能不断完善的前提下成本也在大幅下降，助力了Android的推广与拓展；可穿戴设备以及嵌入式芯片的应用又将为Android系统及其软硬件开发提供新的发展机遇。接触Android领域、使用Android产品、学习Android系统，现在已经成为计算机软件、电子、自动化控制及相关专业的学生和已经参加工作的软硬件工程师的首选。

但是目前市场上同类的Android相关书籍中，多是介绍基于Android SDK的单纯应用程序（APP）的开发，且种类繁杂，对于Android的底层源码与系统内核的分析、驱动程序设计与存储优化、平台移植与内核测试等较为深入的内容则甚少涉及。为了对这些底层领域相关知识稍作弥补，作者编写了本书。

本书而用浅显易懂的语言向广大Android爱好者和开发人员讲解Android系统下嵌入式开发板的设计。

全书共8章，前3章为预备知识，简要介绍嵌入式系统的定义与软硬件开发以及Android开发环境的搭建。这一部分为基础知识，有一定Linux基础和Android开发经验的读者可以选读。

第4~6章为系统结构，主要介绍Android系统的源码结构、内核与相关工具以及环境库。内容包括：Android源码结构、init初始化脚本、Zygote、Android系统编译；Android内核启动、Binder框架、Ashmem内存管理、系统日志Logger实现；Android开发工具、Dalvik虚拟机、JNI、Boot Loader。

第7章为驱动设计，主要介绍Android系统中常用外接设备的驱动架构以及实现。内容包括NDK编程、Android中HAL模型架构与实现流程。作为教学实例，最后还分析了Android系统中的Camera与WiFi两个功能部件的驱动设计。

第8章为实例分析，介绍具体开发板硬件结构的编程原理以及系统级的实例。在简要介绍系统底层开发流程的基础上，首先分析主流的Zynq和pcDuino平台上开发环境的搭建、Linux内核以及Android系统的编译、下载；然后结合前文学习过的知识点，从零开始设计LED显示系统的Linux内核驱动、Android HAL支持和服务层设计、顶层App的实现；作为进阶部分，最后介绍了对Android内核进行跟踪调试与性能测试的工具软件、工作流程以及结果分析。

致谢

在此要特别感谢华章公司的策划编辑张国强，是他对Android和嵌入式系统开发的关注促成了本书的出版。笔者在撰写书稿时，他也对本书提出了宝贵的写作建议，并进行了仔细的审阅。

本书的编写同样离不开许多朋友的支持，在此特别感谢兰州大学信息科学与工程学院的陈华明的大力协助，这本书的出版离不开他的贡献。

还要感谢兰州大学的高博、郭守超、王小强和朱芳芳，感谢他们在本书编写过程中所给予的帮助与建议。

目 录

推荐序

自 序

前 言

第1章 Android嵌入式系统导论 1

1.1 Android嵌入式系统概述 1

1.1.1 嵌入式系统定义 1

1.1.2 基于Android的嵌入式系统 构成 5

1.1.3 移动电话系统 9

1.1.4 基于ARM的移动电话硬件 结构 10

1.2 嵌入式系统实例 17

1.2.1 pcDuino部分硬件功能介绍 17

1.2.2 基于Android的嵌入式系统 30

第2章 Linux系统详解 34

2.1 系统简介 34

2.2 基础命令 35

2.2.1 cd和ls命令 35

2.2.2 touch和mkdir命令 37

2.2.3 rm和rmdir命令 38

2.2.4 cp和mv命令 38

2.2.5 find和awk命令 39

2.2.6 vim编辑器的使用 43

2.3 Bash Shell 44

2.3.1 Bash Shell简介 44

2.3.2 Bash Shell脚本简介 44

2.4 Linux源码与Android源码介绍 45

2.4.1 Linux源码简介 45

2.4.2 Android源码简介 45

第3章 Android系统开发环境 搭建 47

3.1 编译前奏——Android上的开发 工作 47

3.1.1 Android的移植开发 47

3.1.2 系统开发 48

3.1.3 应用开发 49

3.2 Android的系统架构 49

3.2.1 软件结构 49

3.2.2 源代码的结构 51

3.3 搭建开发环境 54

3.3.1 搭建编译环境 54

3.3.2 使用repo 64

3.3.3 Android的编译 66

第4章 Android系统底层源码结构 分析 69

4.1 源码结构分析 69

4.1.1 底层库结构介绍	71	5.4.3 Binder的机制和原理	125
4.1.2 C基础函数库bionic	73	5.5 Ashmem内存管理方式	128
4.1.3 C语言底层库libcutils	74	5.5.1 概述	128
4.1.4 C++工具库libutils	74	5.5.2 Ashmem初始化	128
4.1.5 底层文件系统库system	75	5.5.3 内存的创建和释放	131
4.1.6 增加本地库的方法	76	5.5.4 内存的映射	135
4.2 Android编译系统介绍	78	5.5.5 内存的锁定和解锁	137
4.2.1 build系统	78	5.6 低内存管理	139
4.2.2 SDK	79	5.7 Logger	145
4.3 init初始化脚本语言介绍	82	5.7.1 Logger概述	145
4.3.1 概述	82	5.7.2 Logger实现原理	146
4.3.2 init进程源码分析	82		
4.3.3 脚本文件的创建与分析	85	第6章 Android系统相关工具及	
4.3.4 创建设备节点文件	89	运行环境	156
4.3.5 子进程的创建与终止	92	6.1 Android开发工具分类及介绍	156
4.3.6 属性服务	93	6.1.1 应用程序开发工具	156
4.4 Zygote	95	6.1.2 框架开发工具	157
4.4.1 Zygote概述	95	6.1.3 交叉编译工具	159
4.4.2 AppRuntime分析	95	6.1.4 内核开发工具	159
4.4.3 system_server分析	100	6.2 Dalvik虚拟机	160
第5章 Android系统内核分析	109	6.2.1 概述	160
5.1 Linux内核基础	109	6.2.2 dex文件	163
5.1.1 概述	109	6.2.3 Dalvik内存管理	165
5.1.2 Linux内核的主要子系统	110	6.2.4 Dalvik编译器	169
5.1.3 Linux启动过程分析	111	6.3 JNI	171
5.2 Android内核概况	112	6.3.1 概述	171
5.3 Android启动过程分析	113	6.3.2 JNI的架构	171
5.4 Binder框架分析	115	6.3.3 JNI的实现方式	172
5.4.1 概述	116	6.4 Boot Loader	173
5.4.2 Binder的系统架构	117	6.4.1 概述	173
		6.4.2 Boot Loader 的操作模式	174

6.4.3 启动过程	175	8.2.4 SD卡的准备以及Android 系统的启动	205
6.5 busybox的使用	176	8.3 移植讲解——基于pcDuino的 Android移植	205
第7章 Android驱动程序设计	177	8.3.1 pcDuino介绍	206
7.1 Android驱动概述	177	8.3.2 环境搭建	208
7.2 Android NDK编程	178	8.3.3 编译内核	209
7.3 Android系统中的HAL层	183	8.3.4 编译Android	210
7.3.1 HAL_legacy和HAL对比	184	8.3.5 烧录镜像	211
7.3.2 HAL module 架构分析	184	8.4 Android LED驱动设计	212
7.3.3 HAL实现流程	186	8.4.1 硬件原理	212
7.4 Android系统Camera与WiFi实现	187	8.4.2 Linux驱动设计	212
7.4.1 Android中的Camera实现	187	8.4.3 Android HAL层驱动	220
7.4.2 Android系统WiFi实现	190	8.4.4 硬件服务层	224
第8章 Android底层开发实例 讲解	193	8.4.5 App应用编写	229
8.1 底层开发相关技术概览	193	8.5 进阶讲解——针对Android系统 的内核跟踪与测试	231
8.2 实例讲解——基于Zynq的 Android移植	193	8.5.1 使用平台简介	231
8.2.1 主机开发环境的搭建	194	8.5.2 测试环境的建立	232
8.2.2 Linux内核的编译	194	8.5.3 测试工具	235
8.2.3 Android文件系统的编译	203	8.5.4 Android内核调试与性能 测试	239

第 1 章

Android 嵌入式系统导论

本章主要围绕 ARM 嵌入式系统进行总体讲解，并对相应的硬件系统和软件系统进行讲解。

1.1 Android 嵌入式系统概述

1.1.1 嵌入式系统定义

Android 是一款以 Linux 为基础的开源移动设备操作系统，一直由 Google 公司领导和开发。Google 对于 Android 系统所持有的开放态度，令 Android 系统一经发布就风靡全球。Google 于 2007 年 11 月 5 日正式发布 Android 系统，该平台由操作系统、中间件、用户界面和应用软件组成，是首个为移动终端打造的真正开放的和完整的移动设备软件。2012 年 11 月数据显示，Android 占据全球智能手机操作系统市场的 76%，在中国市场的占有率为 90%。据 2013 年数据显示，全世界采用这款系统的设备数量已经达到 10 亿台。

这里所说的 Android 系统，是基于 Linux 内核发展起来的嵌入式操作系统。Android 系统基于 Linux 内核所做的改进将在后续章节进行详细介绍。

本章将详细介绍嵌入式系统，主要涉及 Android 系统所运行的平台、处理器芯片以及相关的技术。

1. 什么是嵌入式系统

嵌入式系统 (Embedded System) 是一种完全嵌入受控器件内部、为特定应用而设计的专用计算机系统。嵌入式系统的本质就是计算机系统，因而它也是由软件以及硬件构成的。与普通计算机不同，嵌入式系统通常仅拥有非常有限的硬件资源，这种配置使它们的成本大幅下降，但也对软件的优化提出更高的要求。嵌入式系统一般运行固定的程序或固定的操作系统，再加上可变的应用程序。有些为工业系统，仅用于某个特定的控制目的；有些因为有应用程序的加入而更显灵活，一般用于手机、平板电脑。如图 1-1 所示。

第一个被大家认可的现代嵌入式系统是麻省理工学院仪器研究室的查尔斯·斯塔克·德雷珀开发的阿波罗导航计算机。在两次月球飞行中，太空驾驶舱和月球登陆舱都使用了这种

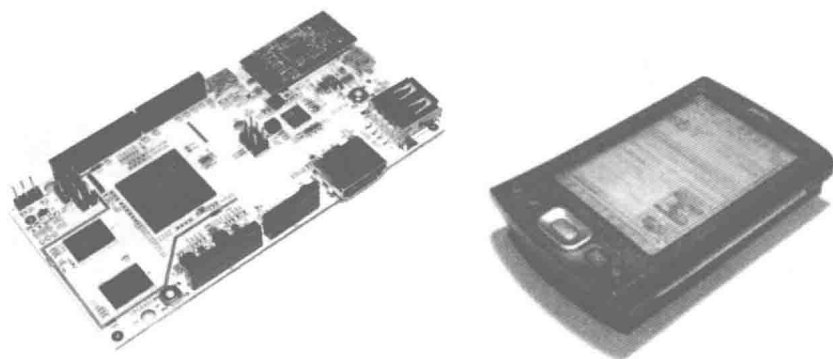


图 1-1 典型的嵌入式系统

惯性导航系统。在计划刚开始的时候，阿波罗导航计算机被认为是阿波罗计划中风险最大的部分。为了减小尺寸和重量而使用的当时最新的单片集成电路，这更加大了阿波罗计划的风险。第一款大批量生产的嵌入式系统是美国军方于 1961 年发布的民兵 I 导弹上的 D-17 自动导航控制计算机。它是由独立的晶体管逻辑电路构成的，并带有一个作为主内存的硬盘。当民兵 II 导弹在 1966 年开始生产的时候，D-17 由第一次使用大量集成电路的更新计算机所替代。仅这个项目就将与非门集成电路模块的价格从每个 1000 美元降低到了每个 3 美元，使集成电路的商用成为可能。民兵导弹的嵌入式计算机有一个重要的设计特性：它能够在项目后期对制导算法重新编程以获得更高的制导精度，并且能够使用计算机测试导弹，从而减轻测试所用的电缆和接头的重量。这些 20 世纪 60 年代的早期应用使嵌入式系统得到了长足发展，它的价格开始下降，同时处理能力和功能获得了巨大的提高。

英特尔 4004 是第一款微处理器，它在计算器和其他小型系统中找到了用武之地。但是，它仍然需要外部存储设备和外部支持芯片。1978 年，美国国家工程制造商协会（NEMA）发布了可编程单片机的“标准”，包括几乎所有以计算机为基础的控制器，如单片机、数控设备，以及基于事件的控制器。随着单片机和微处理器价格的下降，使一些消费性产品用单片机的数字电路取代昂贵模拟组件成为可能。到了 20 世纪 80 年代中期，许多以前是外部系统的组件被集成到了处理器芯片中，这种结构的微处理器得到了更广泛的应用。到了 20 世纪 80 年代末期，微处理器已经出现在几乎所有的电子设备中。

现代的嵌入式系统一般分为简单嵌入式系统和复杂嵌入式系统。简单嵌入式系统一般被认为是由单片集成控制器作为硬件核心的嵌入式系统，其核心只有一片芯片，却集成了处理器、闪存、内存、数字和模拟外设这些设备，这样的系统开发难度低。然而，由于种种限制，其性能一般，仅适合于自动化、运动控制、电源控制等简单的控制类应用。与之相反，复杂嵌入式系统一般由独立的处理器和闪存构成，处理器本身不集成大量的外设，仅执行处理任务，类似于传统计算机的 CPU。这样的系统灵活多变，性能优异，但是成本高昂，普遍用于人机接口、智能设备、手机等性能要求高的场合，如图 1-2 所示。

以下是一些嵌入式系统的典型应用：

- ATM 取款机
- 航空制导系统
- 交换机、路由器、ADSL 终端
- 可穿戴传感器
- 计算机硬盘
- 工业控制器
- 计算器
- 机顶盒
- MP3、MP4
- GPS 导航仪
- 手机、平板电脑
- 智能测试测量仪器仪表

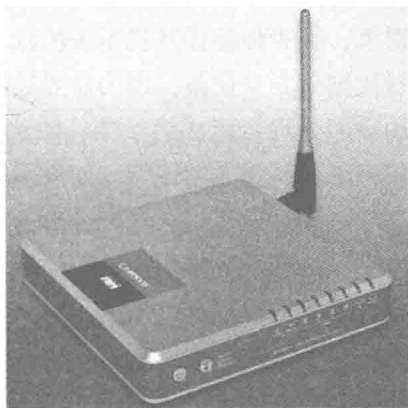


图 1-2 嵌入式系统应用

2. 为什么选择嵌入式系统

很多人觉得没有接触过嵌入式系统，嵌入式系统离他们很遥远。其实，嵌入式系统无处不在。从白色家电到大型网络系统，嵌入式系统时时刻刻为我们服务着。与传统计算机比较，嵌入式系统尽管有开发难度大、通用性差和人机接口普遍落后等劣势，却有着传统计算机所没有的关键优势。

嵌入式系统可以做到极低的成本。一般来说，用于简单工业控制和白色家电的单片芯片集成了复杂的模拟外设、数字外设，而且不用外界任何存储设备，如图 1-3 所示。这样的一片芯片往往售价不超过 10 元人民币，更加令人惊奇的是，这种芯片可以在很宽的电源电压范围内工作，又有良好的可靠性，从而进一步降低了对外部环境的要求，使微电脑控制技术得以广泛普及。

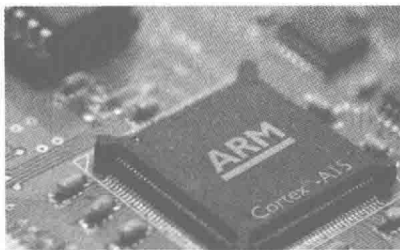


图 1-3 嵌入式系统处理器

嵌入式系统极其可靠。一般说来，系统中串联工作的部件越多，系统的可靠性越差；系统中并联工作的部件越多，系统的可靠性越好。这里的串联指的是相互依赖的工作方式，并联指的是互为冗余的工作方式。嵌入式系统往往有更精简的结构，从而使其有更少的出错机会。从硬件上讲，各模块之间的依赖关系更加清晰，模块数量精简，从而使串联部件减少。从软件上讲，由于使用了定制的操作系统和应用程序，甚至没有操作系统，从而使得软件组件大幅减少，也减少了串联部件。而且，由于结构精简，从而可以留出更多成本预算来做冗余。这样，在减少串联部件的同时增加并联部件，使得嵌入式系统可以提供传统计算机系统所难以比拟的高可靠性。

嵌入式系统极其高效。虽然绝大多数的嵌入式系统拥有较差的计算资源，它们通常仍能完成任务。与通用计算机不同，嵌入式系统的硬件、软件都可以根据实际需求而加以定制，

这使得系统得以精简。除了提高系统的可靠性，精简的系统还能减少硬件资源，尤其是 CPU 资源和内存资源的浪费。一台主频只有 16MHz 的计算机是什么样子？然而，多数用于工业控制的单片机的主频不超过 16MHz，却能井井有条地控制大型机械设备。对于某些超高性能需求，因为传统计算机系统的低效、臃肿，嵌入式系统是唯一的选择。比如说大型电信路由器或网络安全设备，因为需要处理大量的数据请求，还不能有太多的网络延迟或丢包，高效的专用处理器就成为了唯一的选择。它们被制成功能单一的芯片，只能做一种简单的任务，却有着极其强大的性能。如果采用通用计算机来支撑网络社会，可能会因为成本过高而无法实现。

嵌入式系统体积小、功耗低。手机、平板电脑都是嵌入式系统，却也有着不逊于传统计算机系统的性能。一般来说，嵌入式系统的功耗不会大于 20 瓦。多数的单片机功耗在几十毫瓦左右，而多数复杂嵌入式系统的功耗不过几百毫瓦。即使是最先进的手机系统，其峰值功耗不过二三瓦，而其平均功耗不过二三百毫瓦或更低。反观传统计算机，即使是最省电的笔记本电脑也要消耗数十瓦的功率。在减小功耗的同时，嵌入式系统的散热问题也随之消失，更简单的电源管理和几乎不使用散热装置，使得嵌入式系统的体积更小，如图 1-4 所示。

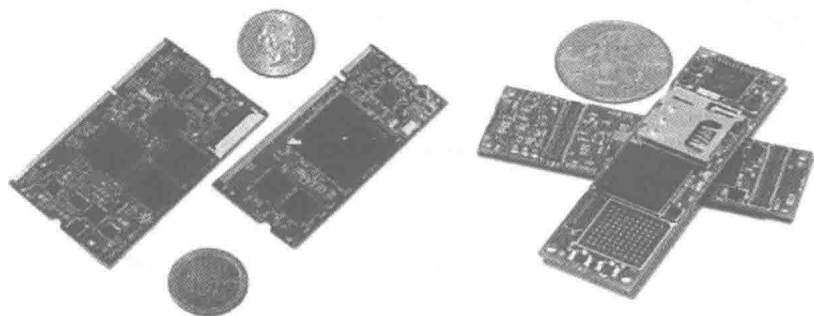


图 1-4 典型的嵌入式系统体积

3. 如何选择

如前文所述，嵌入式系统可分成简单嵌入式系统和复杂嵌入式系统。简单嵌入式系统一般为单片机，比如 MCS51 系列、PIC 系列、AVR 系列和新兴的 MSP430 系列。这些单片机成本低廉，外设丰富，而且在上电后可以立即运行，适合于对性能要求不高的控制类应用，如智能仪表、电机控制、可穿戴传感器和数字电源等。

复杂嵌入式系统的构成则要复杂得多，其性能、成本也高得多。一般来说，该类系统包括基于 DSP 的嵌入式系统、基于 ARM 的嵌入式系统、基于 MIPS 的嵌入式系统和基于 x86 的嵌入式系统。基于 DSP 的嵌入式系统一般用于处理大量的数据，典型应用为语音处理、雷达信号处理等。基于 MIPS 的嵌入式系统一般应用于通用计算，因为 MIPS 在开发之初就被用于通用处理器。基于 x86 的嵌入式系统实际上就是把传统的计算机压缩、精简，一般因其强大的性能与兼容性被用于对成本、功耗要求不高的场合，比如工业计算机等。

复杂嵌入式系统中最常见的要属基于 ARM 的嵌入式系统（以下简称 ARM 系统）。ARM

系统通常拥有足够的硬件资源和相对较低的功耗,这使得 ARM 处理器非常适用于复杂的工业环境和移动终端,这当中绝大部分就基于 Android 操作系统。

1.1.2 基于 Android 的嵌入式系统构成

1. 嵌入式系统架构概述

嵌入式系统由软件和硬件构成。硬件部分包括微处理器、内存、闪存/硬盘、外部设备等;软件部分则包括引导器、操作系统、文件系统、用户程序等。对于简单嵌入式系统来说,硬件部分通常只是一块单片机芯片,集成了所需的全部硬件资源,而软件部分则只是一个单一的二进制文件,包含全部的代码、常数、存储分配结构等。本书的研究对象是基于 ARM 处理器和 Android 操作系统的复杂嵌入式系统。

2. Android 系统的发展与版本演变

Android 一词的本义指“机器人”,该平台由操作系统、中间件、用户界面和应用软件组成。Android 操作系统最初由 Andy Rubin 开发完成,是一种基于 Linux 内核的自由及开放源代码的操作系统,主要应用于移动设备,如智能手机和平板电脑。2005 年 8 月由 Google 收购并注资。2007 年 11 月,Google 与 34 家硬件制造商、软件开发商及电信运营商组建开放手机联盟,共同研发改进 Android 系统,并且以 Apache 开源许可证的授权方式,发布了 Android 的源代码。第一部 Android 智能手机 HTC G1 发布于 2008 年 10 月。随后 Android 操作系统逐渐扩展到平板电脑及其他领域,如电视、数码相机、游戏机等。2011 年第一季度,Android 在全球的市场份额首次超过塞班系统,跃居第一。

2008 年,在 Google I/O 大会上,Google 提出了 Android HAL 架构图,在同年 8 月 18 日,Android 获得了美国联邦通信委员会的批准。在 2008 年 9 月,Google 正式发布了 Android 1.0 系统,这也是 Android 系统最早的版本。

2009 年 4 月,Google 发布了 Android 1.5 版本,并开始以甜品的名字对各个版本进行命名,Android 1.5 命名为 Cupcake。该版本与 1.0 版相比有了很大的改进。

2009 年 9 月,Google 发布了 Android 1.6 的正式版,并且推出了搭载 Android 1.6 正式版的 HTC G3 手机。凭借着出色的外观设计以及全新的 Android 1.6 操作系统,HTC G3 成为当时全球最受欢迎的手机。Android 1.6 也有一个有趣的甜品名称,即 Donut。

2010 年 2 月,Linux 内核开发者 Greg Kroah-Hartman 将 Android 的驱动程序从 Linux 内核 staging tree 上除去。从此,Android 与 Linux 开发主流分道扬镳。同年 5 月,Google 正式发布了 Android 2.2 操作系统,并将其命名为 Froyo。

2010 年 10 月,Google 宣布 Android 系统达到了第一个里程碑,即在电子市场上获得官方数字认证的 Android 应用数量已经达到了 10 万个,Android 系统的应用增长非常迅速。在 2010 年 12 月,Google 正式发布了 Android 2.3 操作系统 Gingerbread。

2011 年 1 月,Google 宣布每日新增的 Android 设备用户数达到 30 万部,到 2011 年 7 月,