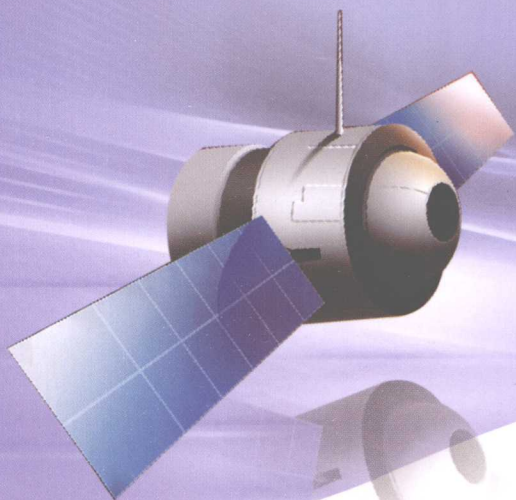


嵌入式设计及 Linux驱动开发指南

——基于ARM9处理器（第3版）

孙天泽 袁文菊 编著



- ◎ 完整、清晰地描述ARM Linux开发流程
- ◎ 深层剖析内核代码以展现Linux驱动程序框架
- ◎ 本书前两版被多所专业培训机构和高等院校选为教材



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

■ 嵌入式技术与应用丛书 ■

嵌入式设计及 Linux 驱动开发指南 ——基于 ARM 9 处理器（第 3 版）

孙天泽 袁文菊 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

ARM 作为一种嵌入式系统处理器,以高性能、低功耗、低成本等优点占领了大部分市场。目前最常见的是 ARM7 和 ARM9 两个系列。随着人们对系统功能提出了更高的要求,ARM7 在某些应用中已经不能胜任,性能更加强劲的 ARM9 处理器逐渐占据了高端产品市场。

本书以 ARM9 处理器和 Linux 操作系统为平台,结合深圳优龙公司的开发板,详细介绍了嵌入式系统开发的流程,并针对 Linux 2.6 版本的内核,介绍了 Linux 设备驱动程序开发方法。

本书是基于 ARM+Linux 嵌入式系统开发的一本实用指导书籍,着重介绍了嵌入式系统开发的一些概念及开发过程,使读者能快速了解嵌入式开发的步骤。本书内容实用易懂,适合高校相关专业的师生阅读,也可作为嵌入式系统开发与应用技术人员和 Linux 设备驱动开发人员的参考书,以及嵌入式培训机构的培训教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

嵌入式设计及 Linux 驱动开发指南:基于 ARM9 处理器 / 孙天泽,袁文菊编著.—3 版.—北京:电子工业出版社,2009.10

(嵌入式技术与应用丛书)

ISBN 978-7-121-09622-8

I. 嵌… II. ①孙…②袁… III. 微处理器, ARM—系统设计 IV.TP332

中国版本图书馆 CIP 数据核字(2009)第 176151 号

责任编辑:高买花 特约编辑:陈宁辉

印 刷: 北京市李史山胶印厂
装 订:

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1 092 1/16 印张:23.25 字数:595 千字

印 次:2009 年 10 月第 1 次印刷

印 数:4000 册 定价:46.00 元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

第 3 版前言

本书以 ARM9 处理器为平台，介绍了嵌入式系统开发的各个主要环节。本书没有过多地讲述原理，而是从代码入手，以分析的角度来学习嵌入式开发的各种技术。每一章内容都独立成为一个主题，较全面地向读者展现了嵌入式系统开发的全部过程。

本书第 1 版的发行时间是 2004 年，创作时间距今已有 6 年。当时可参考的资料非常有限，身边和我一起讨论嵌入式 Linux 的同事也不多。虽然自己已经做了一些项目，但必须承认还没有能力把知识点讲得很透彻。在向出版社交了稿件后，总觉得有限多不如意的地方。但是出乎意料，本书第 1 版在短短 3 个月内就把首印的 5000 册售完，并在一年多重印了 4 次。同时，本书也被一些高校和培训机构用做指定教材，这给了我巨大的鼓舞，同时很多读者的邮件和学员的直接反馈也给我提供了更好的思路。

第 3 版的改动

相对于第 2 版，本版对内容进行了少许调整。在整体结构上进行了优化，使得内容更贴近嵌入式 Linux 项目的开发流程。内容变化最大的两章是 Bootloader 和 USB 驱动开发。在 Bootloader 一章中，不仅进一步阐述了 ARM 启动流程，进一步分析了相关启动文件，而且加入了移植 U-Boot 的完整过程。在 USB 章节中，几乎是使用 2.6 内核进行了重新编写（前面两版都使用了 2.4 的内核）。在内核的章节中，也删除了过时的内容（2.4 内核的移植过程）。

在修订和完善部分章节的同时，也删除了几个章节。其中“调试”和“图形用户接口”两章是从第 1 版就延续下来的，内容需要完全更新。但是考虑到调试是软件开发中很重要的一个环节，嵌入式开发涉及的调试技术范围很宽（尤其是包括内核级的调试技术），在短短一章中很难介绍得详细，认为应该有一本专著介绍调试的内容，所以本书没有加入有关调试的内容。“图形用户接口”和“Java 虚拟机的移植”两章一样，也由于内容陈旧或不够完善而被删除。

组织结构

本书的目标是在每一章讲授嵌入式开发中的一个知识点，以及理解这个知识点的相关概念，因此本书各章节没有很强的相关性，读者可以从中选择感兴趣的章节。下面是本书各章内容的简要说明。

第 1 章 嵌入式系统基础。随着国内外嵌入式产品的进一步开发和推广，嵌入式技术和人们的生活越来越紧密相关，已经广泛渗透到科学研究、工程设计、军事技术、各类产业和商业文化艺术、娱乐业以及人们的日常生活等方方面面中。本章讲述嵌入式系统的基础知识，包括嵌入式系统发展的现状和未来的发展趋势，介绍了目前最流行的几种嵌入式操作系统，以及如何进行嵌入式系统的选型。

第 2 章 基于 ARM9 处理器的硬件开发平台。本章介绍了嵌入式系统及 ARM9 处理

器的一些知识。各种处理器的开发流程基本上是一致的，而首要任务就是认识处理器的信息。本章可以为后续内容打下基础。

第 3 章 创建嵌入式系统开发环境。开发人员不仅仅要熟悉，更要了解嵌入式系统开发环境的创建细节。本章详细介绍交叉编译器的制作过程，通过这个烦琐的过程，读者应该了解有关 `make` 和 `configure` 等 GNU 软件的使用方法。此外，介绍了另外几个开发环境，如 ELDK 和 AIX 等。

第 4 章 Bootloader。Bootloader 是嵌入式开发的一个重点，本章详细分析了 vivi 和 U-Boot 两种 Bootloader 代码。通过本章的学习，读者应该掌握如何编写 Bootloader。对于本章的学习建议是先熟悉 ARM 汇编语言和 ARM 体系结构。最后详细介绍了移植 U-Boot 的整个流程。

第 5 章 Linux 系统在 ARM 平台的移植。随着 2.6 内核加入了对 S3C2410 处理器的支持，移植工作简化很多。尽管本章只谈到了 Linux 操作系统的移植方法，但通过本章的学习，读者能够掌握系统移植的关键步骤和思路，以后在移植 uC/OS 及 uCLinux 等系统时，也能找到着手点。接受了一些读者的建议，在本章后半部对 Linux 内核配置选项进行了较为详细的介绍，以供初学内核编译的读者参考。

第 6 章 Linux 设备驱动程序开发。从本章开始将进入 Linux 驱动开发的内容，以一个非常简单的字符设备为例，介绍 Linux 2.6 内核下驱动程序的开发方法。内容虽然不多，但非常重要，因为大部分嵌入式设备是字符设备。在掌握了本章的内容后，读者能对 Linux 字符设备驱动程序框架和开发流程有清晰的印象。

第 7 章 网络设备驱动程序开发。本章主要讲述网络设备驱动，包括网络设备驱动体系结构、主要的数据结构（`sk_buff` 和 `net_device`）以及接口函数等。通过本章的学习，读者能够掌握编写网卡驱动程序的能力。

第 8 章 USB 驱动程序开发。本章主要讲述 USB 设备驱动，包括 USB 设备驱动体系结构、主要的数据结构以及接口函数等，并结合开发板的 USB 芯片，说明了 USB 驱动程序的实现方法。

第 9 章 嵌入式文件系统。文件系统是嵌入式开发中的另一个重点。本章对嵌入式文件系统特点以及部署进行了说明，介绍了制作根文件系统的流程。

第 10 章 系统设计开发。本章介绍嵌入式系统中的硬件接口设计，使熟悉单片机电路设计的工程师可以迅速了解基于 ARM 的硬件电路设计。

关于开发环境

在之前的第 2 版中，提供了基于光盘启动的嵌入式 Linux 开发环境，并将书中的一些源码随光盘配上，开发环境也为一定的读者带来了便利。不过通过近几年的教学工作发现，读者在学习的过程中，彻底抛开 Windows 还有些困难，可能“Windows+虚拟机”的学习环境更适合初学者使用。因此本版取消了第 2 版中配套光盘的资源。如有可能，我会在以后的版本中提供基于虚拟机的开发环境。在使用本书的过程中，读者只要安装 Linux 即可学习大部分开发内容。本书仍然使用 SuSE Linux 发行版，当然其他版本（如 Ubuntu 或 Red Hat Linux）也是可以的，不会影响阅读。

读者对象

本书是一本介绍嵌入式系统开发的书籍，同时也介绍了嵌入式 Linux 驱动程序开发的内容。本书适合以下人员阅读：

- 想学习或者刚刚进入嵌入式系统领域的开发人员；
- 在非 X86 平台上工作，想学习 ARM 处理器的开发人员；
- 熟悉 Windows 的 WDM 驱动开发，但想转到 Linux 系统下的驱动程序开发人员；
- 对嵌入式系统或者 Linux 系统非常有兴趣的人员。

尽管本书面向初级嵌入式系统开发人员，但读者需要熟悉 C 语言和汇编语言（包括 ARM 汇编语言），至少要能读懂书中提到的代码，还要熟悉一些 Linux 的基本命令。如果读者对 Linux 内核代码比较熟悉，那么读本书的代码会更加容易。如果之前没有接触过 Linux，那么最好先阅读一些 Linux 入门的书。

致谢

有太多的人对本书的出版作出了巨大贡献，在此对他们表示衷心的感谢！

首先要感谢的是我最敬爱的父母，他们给予我的支持和鼓励是我最宝贵的财富。

其次要感谢北京华清远见科技信息有限公司的专家组以及学员，他们认真阅读了本书并提出了大量中肯的建议，也帮助修订了书稿中的很多错误。

全书由孙天泽统稿并组织完成，没有下面一些朋友的帮助，本书也不会完成，在此要对他（她）们表示衷心的感谢：黄昕、蒙洋、唐坤勇、李佳、田贺祥、陈静、游成伟。

由于篇幅所限、时间仓促，加之作者的编写水平有限，书中的不妥之处在所难免，恳请读者批评指正。关于本书的任何批评和建议及嵌入式 Linux 的技术问题交流，可以发表到 www.farsight.com.cn 技术论坛，我将邀请华清远见的各位专家协助进行解答，书稿相关问题也可以和本书的策划编辑高老师（gmholife@hotmail.com）联系。

编 著 者
2009 年 9 月

目 录

第 1 章 嵌入式系统基础	1
1.1 嵌入式系统简介	1
1.1.1 嵌入式系统定义	1
1.1.2 嵌入式系统与 PC	2
1.1.3 嵌入式系统的特点	4
1.2 嵌入式系统的发展	4
1.2.1 嵌入式系统现状	4
1.2.2 嵌入式系统发展趋势	5
1.3 嵌入式操作系统与实时操作系统	6
1.3.1 Linux	7
1.3.2 uC/OS	8
1.3.3 Windows CE	9
1.3.4 VxWorks	10
1.3.5 Palm OS	11
1.3.6 QNX	11
1.4 嵌入式系统选型	12
第 2 章 基于 ARM9 处理器的硬件开发平台	13
2.1 ARM 处理器简介	13
2.1.1 ARM 公司简介	13
2.1.2 ARM 微处理器核	14
2.2 ARM9 微处理器简介	19
2.2.1 与 ARM7 处理器的比较	19
2.2.2 三星 S3C2410X 处理器详解	20
2.3 FS2410 开发平台	55
第 3 章 创建嵌入式系统开发环境	58
3.1 嵌入式 Linux 的开发环境	58
3.2 Cygwin	59
3.3 虚拟机	62
3.4 交叉编译的预备知识	64
3.4.1 Make 命令和 Makefile 文件	65
3.4.2 binutils 工具包	68
3.4.3 gcc 编译器	92
3.4.4 Glibc 库	94
3.4.5 GDB	96
3.5 交叉编译	101
3.5.1 创建编译环境	102

3.5.2	编译 binutils	104
3.5.3	编译 bootstrap_gcc	107
3.5.4	编译 Glibc	109
3.5.5	编译完整的 gcc	112
3.5.6	编译 GDB	113
3.5.7	成果	113
3.5.8	其他交叉编译方法	114
3.6	通过二进制软件包创建交叉编译环境	117
3.7	主机交叉开发环境配置	117
第 4 章	Bootloader	121
4.1	嵌入式系统的引导代码	121
4.1.1	初识 Bootloader	121
4.1.2	Bootloader 的启动流程	123
4.2	Bootloader 之 vivi	124
4.2.1	vivi 简介	124
4.2.2	vivi 的配置与编译	125
4.2.3	vivi 代码导读	126
4.3	Bootloader 之 U-Boot	141
4.3.1	U-Boot 代码结构分析	142
4.3.2	编译 U-Boot 代码	142
4.3.3	U-Boot 代码导读	143
4.3.4	U-Boot 命令	156
4.3.5	U-Boot 移植	157
4.4	FS2410 的 Bootloader	170
第 5 章	Linux 系统在 ARM 平台的移植	172
5.1	移植的概念	172
5.2	Linux 内核结构	172
5.3	Linux 内核向 ARM 平台的移植	174
5.3.1	定义平台和编译器	174
5.3.2	arch/arm/mach-s3c2410/devs.c	175
5.3.3	arch/arm/mach-s3c2410/mach-fs2410.c	178
5.3.4	串口输出	178
5.4	编译 Linux 内核	178
5.4.1	代码成熟等级选项	179
5.4.2	通用的一些选项	180
5.4.3	和模块相关的选项	181
5.4.4	和块相关的选项	182
5.4.5	和系统类型相关的选项	182
5.4.6	和总线相关的选项	183
5.4.7	和内核特性相关的选项	184

5.4.8	和系统启动相关的选项	184
5.4.9	和浮点运算相关的选项	184
5.4.10	用户空间使用的二进制文件格式的选项	185
5.4.11	和电源管理相关的选项	185
5.4.12	和网络协议相关的选项	186
5.4.13	和设备驱动程序相关的选项	188
5.4.14	和文件系统相关的选项	192
5.4.15	和程序性能分析相关的选项	194
5.4.16	和内核调试相关的选项	195
5.4.17	和安全相关的选项	196
5.4.18	和加密算法相关的选项	197
5.4.19	库选项	197
5.4.20	保存内核配置	197
第 6 章	Linux 设备驱动程序开发	199
6.1	设备驱动概述	199
6.1.1	设备驱动和文件系统的关系	200
6.1.2	设备类型分类	201
6.1.3	内核空间和用户空间	203
6.2	设备驱动基础	204
6.2.1	设备驱动中关键数据结构	204
6.2.2	字符设备驱动开发	208
第 7 章	网络设备驱动程序开发	235
7.1	网络设备驱动程序简介	235
7.1.1	device 数据结构	236
7.1.2	sk_buff 数据结构	242
7.1.3	内核的驱动程序接口	244
7.2	以太网控制器 CS8900A	245
7.2.1	特性	245
7.2.2	工作原理	246
7.2.3	电路连接	247
7.2.4	引脚	247
7.2.5	操作模式	249
7.3	网络设备驱动程序实例	249
7.3.1	初始化函数	249
7.3.2	打开函数	255
7.3.3	关闭函数	258
7.3.4	发送函数	258
7.3.5	接收函数	259
7.3.6	中断处理函数	262

第 8 章 USB 驱动程序开发	265
8.1 USB 驱动程序简介	265
8.1.1 USB 背景知识	265
8.1.2 Linux 内核对 USB 规范的支持	266
8.1.3 OHCI 简介	267
8.2 Linux 下 USB 系统文件结点	276
8.3 USB 主机驱动结构	277
8.3.1 USB 数据传输时序	279
8.3.2 USB 设备连接/断开时序	280
8.4 主要数据结构及接口函数	281
8.4.1 数据传输管道	281
8.4.2 统一的 USB 数据传输块	282
8.4.3 USB 数据描述	284
8.4.4 USB 与 HCD 驱动程序接口	285
8.4.5 设备类驱动与 USB 接口	292
8.5 USB HUB 驱动程序	302
8.5.1 HUB 驱动初始化	303
8.5.2 HUB Probe 相关函数	304
第 9 章 嵌入式文件系统	307
9.1 文件系统简介	307
9.2 Linux 根文件系统	308
9.3 Linux 系统的引导过程	310
9.3.1 启动内核	310
9.3.2 init	312
9.3.3 mingetty	314
9.4 嵌入式文件系统	314
9.4.1 嵌入式文件系统的特点和种类	314
9.4.2 常见的嵌入式文件系统	316
9.4.3 MTD	326
9.4.4 RAMDISK	329
9.4.5 网络文件系统	331
9.5 使用 Busybox 制作根文件系统	334
第 10 章 系统设计开发	339
10.1 概述	339
10.2 硬件功能的实现	340
10.2.1 功能定义	341
10.2.2 原理图设计	342
10.2.3 PCB 设计	355
10.2.4 硬件调试	357
参考文献	358

第 1 章 嵌入式系统基础



本章要点

本章将学习嵌入式系统的基本概念。读完本章，读者应该了解以下内容：

- 什么是嵌入式系统；
- 什么是嵌入式操作系统；
- 嵌入式系统的发展趋势；
- 目前有哪些流行的嵌入式操作系统，如何选择。

1.1 嵌入式系统简介

1.1.1 嵌入式系统定义

嵌入式系统（Embedded System）无疑是当今最热门的概念之一，但究竟什么是嵌入式系统？在多数网站和书籍资料中，对嵌入式系统的定义大多是这样的：嵌入式系统是以应用为中心，以计算机技术为基础，并且软硬件可裁剪，适用于应用系统对功能、可靠性、成本、体积、功耗有严格要求的专用计算机系统。嵌入式系统一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统以及用户应用程序 4 个部分组成，用于实现对其设备的控制、监视或管理等功能。

电气工程师协会的定义似乎更权威一些：嵌入式系统是用来控制或者监视机器、装置、工厂等大规模系统的设备。

北京航空航天大学何立民教授是这样定义嵌入式系统的：“嵌入到对象体系中的专用计算机系统”。“嵌入性”、“专用性”与“计算机系统”是嵌入式系统的三个基本要素。对象系统则是指嵌入式系统所嵌入的宿主系统。

MP3、PDA 和手机属于手持的嵌入式产品；DVD 机和机顶盒是嵌入式产品；车载 GPS 系统和探测火星的机器人也可以称做嵌入式系统产品。

有人这样认为：8 位单片机不是嵌入式系统，只有使用了高性能的 32 位处理器的系统才能称为嵌入式系统。这个理解是错误的。在单片机时代，计算机专业淡出了嵌入式系统领域，但随着后 PC 时代的到来，网络、通信技术得以发展，把处理器嵌入到各种个性化产品中成为大势所趋。早在 20 世纪 70 年代单片机出现时，汽车和家用电器等电子产品就内部嵌入式了电子装置，当时的这些装置已经具备了嵌入式的一些特点，不过因为 8 位芯片能力不是很强，只执行了一些简单的程序，因而不能称做“系统”，只能叫做嵌入式设备。但嵌入式系统最初的应用是基于单片机的。其实在嵌入式领域中，处理器的种类非常丰富。从位数上分，包括 4 位处理器（一般多用于控制）、8 位微控制器（最

常见的就是 89C51 系列)、16 位(多应用于汽车电子),以及日益升温的 32 位处理器(多用于嵌入式计算)。

1.1.2 嵌入式系统与 PC

一般来说,嵌入式系统由处理器、存储器、输入/输出设备和软件 4 部分组成,如图 1-1 所示。

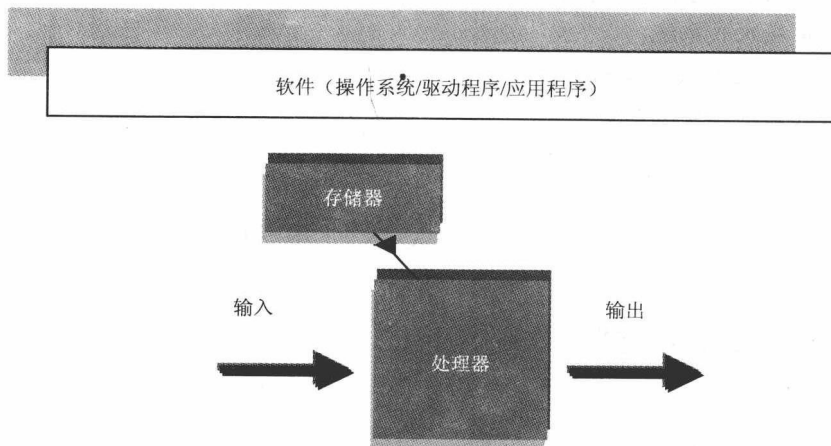


图 1-1 嵌入式系统组成

图 1-1 是一个典型的嵌入式系统的概念模型,不过看起来有点抽象,可通过具体的嵌入式产品来理解这几个部分。图 1-2 和图 1-3 是两款已经上市的嵌入式产品(智能手机)。



图 1-2 智能手机

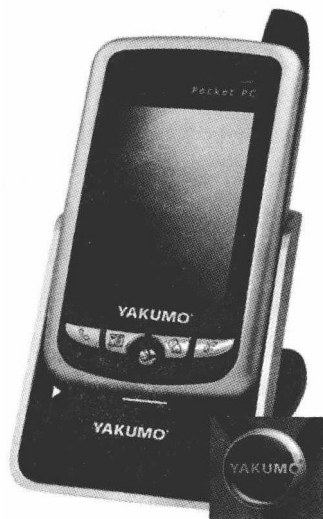


图 1-3 Yakumo 新款 PDA 照相机手机 (omikron)

某个嵌入式产品（智能手机）的内部结构如图 1-4 所示。

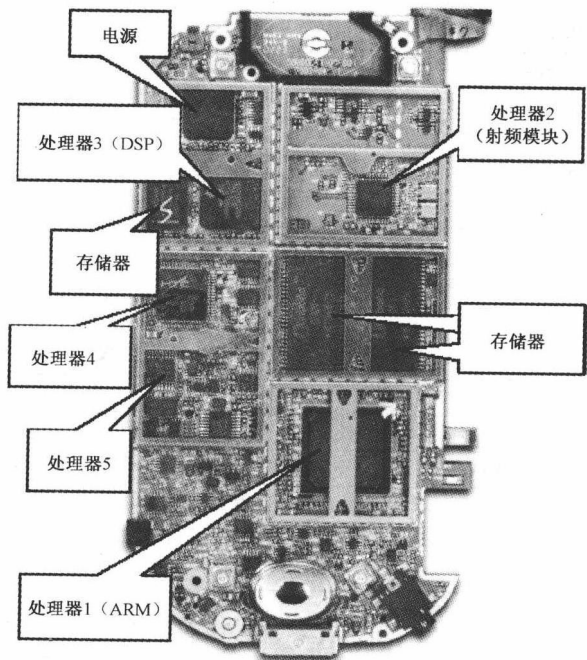


图 1-4 嵌入式产品（智能手机）内部结构

拆开手机就可以看到处理器和存储器部分。在这个设备中包括好几个处理器，分别用来完成不同的功能。存储器用来存储短信、电话号码和相片等信息。图片上美丽的画面就是由软件编程实现的。在图 1-1 中，软件这部分由四层组成，即操作系统层、驱动程序层和应用程序层。有关软件分层的内容将在后面详细说明。

与最初使用模拟信号的“大哥大”相比，现在的手机不仅仅用来通信，技术的发展赋予它更多的任务，如收取电子邮件和照相，甚至阅读 Word、Excel、Powerpoint 和 PDF 等格式的文件等。手机功能的日益强大，与嵌入式系统的飞速发展是分不开的。

既然有了显示器、键盘、处理器甚至还有内存，那么嵌入式系统和家用的 PC 有什么不同？家用的电脑是不是嵌入式系统？答案是否定的。嵌入式系统的一个特点就是“专用”，即使同样是手机，哪怕是同一个公司的不同型号的手机，它的电路和芯片都不会完全相同。而 PC 则不然，爱国者显示器可以用在清华同方主机上，逻辑鼠标可以插在 IBM 笔记本上。表 1-1 和表 1-2 对比了嵌入式系统和 PC。

1. 硬件平台

表 1-1 硬件平台的比较

设备名称	嵌入式系统	PC
CPU	嵌入式处理器（ARM, MIPS 等）	CPU（Intel 的 Pentium、AMD 的 Athlon 等）
内存	SDRAM 芯片	SDRAM, DDR 内存条

续表

设备名称	嵌入式系统	PC
存储设备	Flash 芯片	硬盘
输入设备	按键、触摸屏	鼠标、键盘
输出设备	LCD	显示器
声音设备	音频芯片	声卡
接口	MAX232 等芯片	主板集成
其他设备	USB 芯片、网卡芯片	主板集成或外接卡

2. 软件及开发平台

表 1-2 软件及开发平台的比较

	嵌入式系统	PC
引导代码	Bootloader 引导，针对不同电路板进行移植	主板的 BIOS 引导，无须改动
操作系统	Windows CE、VxWorks、Linux 等，需要移植	Windows、Linux 等，不需要移植
驱动程序	每个设备驱动都必须针对电路板进行重新开发或移植，一般不能直接下载使用	操作系统含有大多数驱动程序，或从网上下载直接使用
协议栈	需要移植	操作系统提供，或第三方提供
开发环境	借助服务器进行交叉编译	在本机就可开发调试
仿真器	需要	不需要

从以上两表可以看出，嵌入式系统和 PC 有很大的不同，这也正是嵌入式系统的灵活之处，即“软硬件可裁剪”。

1.1.3 嵌入式系统的特点

嵌入式系统同 PC 系统相比有以下特点：

- （1）嵌入式系统功耗低、体积小、专用性强。嵌入式系统与 PC 的最大不同就是嵌入式 CPU 大多工作在为特定用户群设计的系统中，能够把 PC 中许多由板卡完成的任务集成在芯片内部，从而有利于嵌入式系统设计趋于小型化。
- （2）为了提高执行速度和系统可靠性，嵌入式系统中的软件一般都固化在存储器芯片或单片机本身中，而不是存储于磁盘等载体中。
- （3）嵌入式系统的硬件和软件都必须高效率地设计，系统要精简。操作系统一般和应用软件集成在一起。
- （4）对软件代码质量要求很高，应该尽最大可能避免“死机”的发生。
- （5）嵌入式系统开发需要专门的开发工具和开发环境。

1.2 嵌入式系统的发展

1.2.1 嵌入式系统现状

后 PC 时代，嵌入式系统将拥有最大的市场。目前在世界范围内嵌入式系统带来的工

业年产值已超过了 1 万亿美元。据赛迪呼叫中心合作开展的调查显示, 目前从事嵌入式开发的软件企业占到了 30%, 计划从事开发的企业占到了 14%。尽管还没有从事该领域开发的公司占到了大多数, 但我国国内的嵌入式软件市场已处于整体启动阶段。国内嵌入式软件市场未来的发展重点在于对应用范围的拓展, 而手持设备、信息家电和工业控制则是近期市场的三大热点。展望未来, 明天的嵌入式系统将会比今天的更便宜、更小巧、更可靠、更高效而且更智能化。

近年来微电子技术迅猛发展, 处理器增长速度也随之加快。嵌入式系统领域发生了翻天覆地的变化, 特别是网络的普及, 嵌入式与互联网成为最热门的技术。现在, 可以使嵌入式系统具备网络功能, 并将它们与 Internet 或企业局域网连接起来。这种特性增强了嵌入式系统多方面的实用性。图 1-5 是嵌入式系统的一些应用。

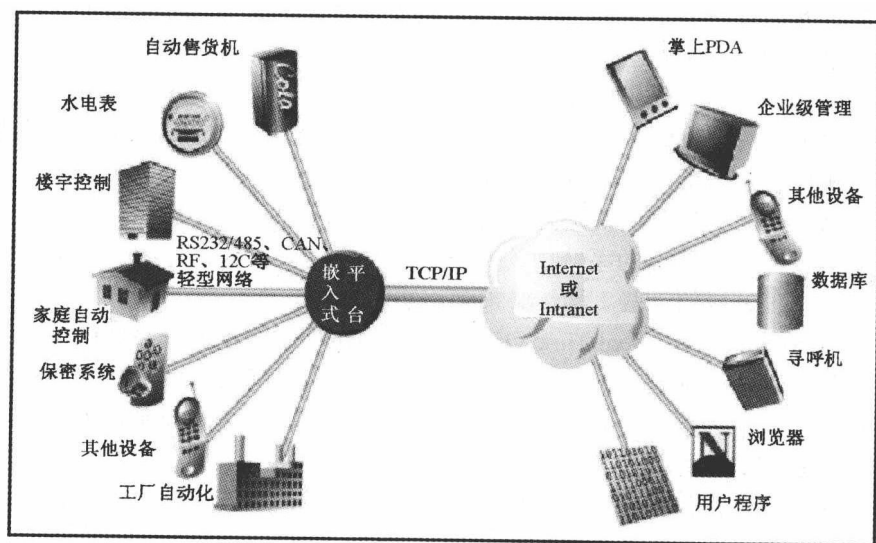


图 1-5 一些嵌入式产品

1.2.2 嵌入式系统发展趋势

现在嵌入式系统正处于高速发展阶段, 未来的几年里, 这种发展和竞争将越来越激烈。Venture Development 公司在 2003 年初预测, 嵌入式操作系统、捆绑工具及相关服务的市场到 2007 年会超过 10 亿美元, 其中亚太地区的增长最快。该公司还预测, 非捆绑嵌入式软件开发工具的市场将达到 5.252 亿美元, 年复合增长率为 9.5%。

将嵌入式系统的应用按照市场领域划分, 可以分为:

- 消费类电子产品;
- 控制系统和工业自动化;
- 机器人领域;
- 数据/无线通信。

目前一些先进的 PDA 在显示屏幕上已实现汉字写入和短消息语音发布, 应用范围也将日益广阔。对于企业专用解决方案, 如物流管理、条码扫描和移动信息采集等, 这种

小型手持嵌入式系统将发挥巨大的作用。自动控制领域，不仅可以用于 ATM 机、自动售货机、工业控制等专用设备，和移动通信设备、GPS、娱乐相结合，嵌入式系统同样可以发挥巨大的作用。

尽管嵌入式产品层出不穷，发展势头迅猛，但仍有需要我们研究探索，比如嵌入式移动数据库。就目前的车载嵌入式产品来说，还没有实现非常完善的自动导航功能，究其原因还是数据处理没有达到要求。

1.3 嵌入式操作系统与实时操作系统

据调查，目前国际上有两百多种嵌入式操作系统，这些嵌入式操作系统可分为实时操作系统和非实时操作系统两种。对于时间要求非常高的系统，可称为实时系统。我们个人电脑中常用的 Windows 系统不算实时操作系统，因为实时系统的一个重要特点就是对时间要求非常严格。如果实时系统没有在某个预定的时间内响应某个事件，系统就会出错。举例来说，假设我们在 PC 上打开一个应用程序（比如 Visual Studio）需要 3s，但有一天你的机器中了病毒，打开同样的这个程序需要 10s，你会怎么样？除了多等待几秒钟外，没有什么大不了的。这也就是为什么说 Windows 不是一个实时操作系统的原因。而在一个对时限要求特别严格的系统中，几毫秒、甚至几微秒的误差都是不可接受的。特别是在航天、军工领域的一些工程，必须使用实时操作系统。

实时多任务操作系统（Real Time Operating System）是根据操作系统的工作特性而言的。实时是指物理进程的真实时间。实时操作系统是指具有实时性，能支持实时控制系统工作的操作系统。首要任务是调度一切可利用的资源完成实时控制任务，其次才着眼于提高计算机系统的使用效率，重要特点是要满足对时间的限制和要求。实时多任务操作系统与分时多任务操作系统有明显区别。对于分时操作系统，软件的执行在时间上的要求并不严格，时间上的错误一般不会造成灾难性的后果。而对于实时操作系统，主要任务是对事件进行实时的处理，虽然事件可能在无法预知的时刻到达，但是必须在事件发生时能够在严格的时限（系统响应时间）内做出响应，即使是在尖峰负荷下也应如此，系统响应时间的超时就意味着致命的失败。另外，实时操作系统的重要特点是具有系统的可确定性，即系统能对运行的最好和最坏等情况做出精确的估计。



提示

硬实时系统有一个刚性的、不可改变的时间限制，它不允许任何超出时限的错误。超时错误会带来损害甚至导致系统失败，或者导致系统不能实现预期目标。

软实时系统的时限是柔性灵活的，它可以容忍偶然的超时错误。失败造成的后果并不严重，仅仅是轻微降低了系统的吞吐量。

“实时”和“嵌入”这两个术语联系非常紧密，以至于很多人会提出这样的问题：嵌入式系统是不是实时系统？或者问实时系统是不是嵌入式系统？应该说，它们是同义的，但不是相同的，我们可以认为嵌入式系统都是软实时系统。所有的嵌入式系统都是实时系统，但并不是所有的实时系统都是嵌入式系统。当然，读者必须理解软实时和硬

实时的区别。Linux 操作系统早期版本，比如 2.4 内核以前，社区内的开发人员并没有在实时性上投入太多的精力。直至 2.6 内核以后，Linux 内核的实时性有了很大的改进，引入了先进的进程调度算法。有关实时操作系统的内容读者可以访问下面一些网络资源。

<http://www.linuxworks.com>

<http://www.timesys.com>

<http://www.estc.com>

<http://www.realtime-info.be>

<http://www.eg3.com>

<http://www.rti.com>

从 20 世纪 80 年代开始，出现了各种各样的商用嵌入式操作系统，这些操作系统大部分都是为专有系统而开发，从而形成了现在多种商用嵌入式操作系统百家争鸣的局面，如 VxWorks、pSOS、Neculeus、Linux 和 Windows CE 等，接下来我们简单介绍目前较为流行的嵌入式操作系统。

1.3.1 Linux

Linux 真的是一个奇迹！从 1991 年 8 月 Linus Torvalds 宣布其诞生到现在已经近 20 年。在众多的操作系统中，Linux 是一个发展最快、应用最广泛的操作系统。Linux 本身的种种特性使其成为嵌入式开发中的首选。赛迪顾问发布的中国 Linux 市场报告预测，2007—2011 年未来五年内，中国 Linux 软件市场仍将处于上升阶段，但增长速度将逐步放缓。到 2011 年，中国 Linux 软件市场规模将达到 5.78 亿元，年均复合增长率为 28.8%。

根据 IDC 的报告，Linux 已经成为全球第二大操作系统。预计在服务器市场上，Linux 在未来几年内将以每年 25% 的速度增长，中国的 Linux 市场更是保持 40% 左右的增长速度。

为什么 Linux 发展如此之快？主要原因是产品的成本。在激烈的市场竞争中，只有先进的技术远远不够，最重要的是如何以最少的成本实现同样的产品。免费的 Linux 无疑为厂商节约了一大笔开支，特别是对于经济实力不强的公司来说。

嵌入式 Linux 协会（ELC）在 2002 年底发布了 ELC 产品规范版本 1.0（ELCPS）。它给我们提供了以下好处：

- 嵌入式 Linux 发布可以很清楚地显示自身的功能特性。
- 第三方类库和中间程序在很少或没有额外接口的条件下就可以运行不同的嵌入式 Linux 配置。
- 软件开发人员可以将程序段写入一个现成的 API 中，并相信这一代码段可为其他的嵌入式 Linux 环境提供接口。

另一种常用的嵌入式 Linux 是 uClinux，它去掉了 Linux 中的内存管理单元，主要用于没有内存管理单元的处理器（比如三星公司的 S3C44B0X 处理器），该系统直接访问内存，所有程序中访问的地址都是物理地址。本书中使用的处理器带有内存管理单元，因此使用完整的 Linux 系统。嵌入式 Linux 具有以下特点：