

工程设计与分析系列

# 嵌入式 (第2版) *Linux*系统与工程实践

戴璐平 何渊仁 吴志男 编著



## 视频教学

本书程序代码资源及视频下载：  
[www.hxedu.com.cn](http://www.hxedu.com.cn)

- ★ Linux —— 嵌入式系统开发的首选软件
- ★ 信息家电、工业控制、医疗器械、机器人领域的必备技术
- ★ 基础知识—实训实例—工程实例
- ★ 实例操作视频教学，轻松学习



中国工信出版集团



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

工程设计与分析系列

嵌入式

# Linux 系统与工程实践 (第2版)

戴璐平 何渊仁 吴志男 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

本书以 Linux 嵌入式系统的基本开发技术为主线,以基于 ARM 架构的嵌入式处理器为嵌入式硬件平台,全面介绍嵌入式系统开发过程、ARM 体系结构、Linux 基础、Linux 进程、Linux 开发环境的建立、Linux 操作系统移植、Bootloader 的使用、Linux 根文件系统的构建、设备驱动程序的开发、嵌入式 GUI 开发等嵌入式知识,最后介绍近年来较为热门的 GPS 导航系统的设计。书中实例全部配有视频讲解,实例代码和视频等素材,请读者到华信教育资源网的本书页面下载 ([www.hxedu.com.cn](http://www.hxedu.com.cn))。

本书可作为机电控制、信息家电、工业控制、手持仪器、医疗器械、机器人技术等方面嵌入式系统开发与应用的参考书,也可作为高等院校有关嵌入式系统教学的教材。

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 Linux 系统与工程实践/戴璐平,何渊仁,吴志男编著. —2 版. —北京:电子工业出版社,2017.3  
(工程设计与分析系列)  
ISBN 978-7-121-31053-9

I. ①嵌… II. ①戴… ②何… ③吴… III. ①Linux 操作系统—程序设计 IV. ①TP316.85

中国版本图书馆 CIP 数据核字 (2017) 第 046539 号

策划编辑:许存权

责任编辑:许存权 特约编辑:谢忠玉 等

印 刷:北京京师印务有限公司

装 订:北京京师印务有限公司

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:34.5 字数:886 千字

版 次:2012 年 1 月第 1 版

2017 年 3 月第 2 版

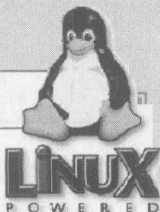
印 次:2017 年 3 月第 1 次印刷

定 价:69.00 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, 88258888。

质量投诉请发邮件至 [zlt@phei.com.cn](mailto:zlt@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

本书咨询联系方式:(010) 88254484, [xucq@phei.com.cn](mailto:xucq@phei.com.cn)。



## 前 言

嵌入式系统已经广泛地渗透到航空航天、汽车电子、医疗网络通信、工业控制等各个领域,正在以不同的形式悄悄地改变着人们的生产、生活方式,已经成为计算机领域的一个亮点。嵌入式系统和 Linux 的有机结合,成为后 PC 时代计算机最普遍的应用形式。嵌入式 Linux 不仅继承了 Linux 源代码开放、内核稳定高效、软件丰富、强大的网络支持功能、优秀的开发工具等优势,还具备支持广泛的处理器结构和硬件平台占有空间小、成本低等特点。

嵌入式 Linux 需要相应的嵌入式开发板和软件,还需要有经验的人员进行指导开发,目前国内大部分高校都很难达到这种要求,这也造成了目前国内嵌入式 Linux 开发人才极其缺乏的局面。

从技术角度来讲,嵌入式系统是软件和硬件的有机结合体,一名合格的嵌入式系统设计人员往往要求同时具备软件和硬件两方面的知识。因此,这也是编写本书的目的所在。

本书分为 4 篇 12 章,涉及嵌入式系统开发基础、嵌入式 Linux 开发入门、嵌入式系统移植与构建、嵌入式系统开发四大部分内容,依次介绍嵌入式系统基本概念、嵌入式系统开发过程与工具、嵌入式处理器体系结构、Linux 基础概述、Linux 内核、Linux 开发环境建立、Linux 操作系统移植、Bootloader 的使用、Linux 根文件系统的构建、设备驱动程序开发、嵌入式 GUI 开发、综合工程实例等内容。

第 1 章 嵌入式系统介绍。本章首先介绍嵌入式系统的组成部分、处理器、常见的嵌入式操作系统和新型的嵌入式操作系统,然后介绍嵌入式系统的应用,最后讲述嵌入式系统的发展趋势。

第 2 章 嵌入式软件开发过程与工具。本章首先讲述嵌入式软件的开发流程和软件测试技术,重点介绍基于 JTAG 的 ARM 系统调试,通过综合实例掌握 ADS 集成开发环境的使用。

第 3 章 嵌入式处理器体系结构。本章主要介绍 ARM 体系结构的组成部分、技术特征、ARM 微处理器的分类和应用选型,接着,重点以 S3C2410 处理器为例进行讲述,并详细介绍 ARM 编程模型、ARM 指令的寻址方式和 ARM 指令集的使用,最后,描述了 ARM 微处理器的异常情况。

第 4 章 Linux 开发常用操作。本章首先讲述 Linux 的概况,然后分别介绍 Linux 命令和 vi 编辑器的使用,接着讲述 Shell 程序设计语言的使用。最后,通过讲述编写清除 /var/log 下的 log 文件和编写寻找死链接文件两个综合实例的操作,掌握 Shell 编程的具体使用。

第 5 章 Linux 内核介绍。本章首先讲述进程结构的控制操作和属性,然后分别介绍管道、信号、信号量、共享内存和消息队列的使用,最后通过多个实例的操作,掌握进程的具体使用。



第 6 章 Linux 开发环境的构建。本章首先讲述 Cygwin 和 VMware Workstation 两种开发环境的建立, 介绍建立交叉编译环境的主要过程, 然后分别介绍 gcc 编辑器和 gdb 调试器的使用方法, 最后详细讲解 Makefile 变量的使用以及隐含规则的应用。通过多个实例的操作, 掌握 Linux 开发环境的建立。

第 7 章 Bootloader 的移植。本章首先介绍 Bootloader 的工作模式, 讲述 Bootloader 的启动方式和流程。然后详细介绍 vivi 代码的两个阶段, 并重点介绍 vivi 的配置与编译、U-boot 常用命令和源代码目录结构, 讲述 U-boot 的启动模式和启动流程, 并重点介绍 U-boot 在 S3C2410 上的移植。最后简单介绍其他常见的 Bootloader, 通过多个实例的操作, 掌握 Bootloader 的使用。

第 8 章 Linux 内核裁剪和移植。本章主要介绍 Linux 操作系统移植知识, 重点讲述 Linux 内核结构和操作系统移植, 通过多个综合实例的操作, 掌握 Linux 操作系统的移植技术。

第 9 章 Linux 根文件系统的构建。本章首先讲述文件系统, 然后利用 BusyBox 构建根文件系统, 使读者对根文件系统有全面的了解。

第 10 章 设备驱动程序开发。本章首先讲述 Linux 设备驱动程序的分类, 包括字符设备、块设备、网络设备, 以及驱动程序在 Linux 中的层次结构和其特点。然后对设备驱动程序与文件系统的关系、Linux 设备驱动程序的接口、Linux 驱动程序的加载方法及其步骤进行分析。接着, 讲述设备驱动程序的使用、网络设备的基础知识和网络设备驱动程序的体系结构、模块分析、实现模式。最后, 通过多个实例的操作, 掌握设备驱动程序的具体使用。

第 11 章 嵌入式常用 GUI 开发。本章主要介绍各种嵌入式 GUI 的相关知识, 包括 MiniGUI 的实现、Qt/E 的界面编程和 Qtopia 移植等。大部分知识点后面都有相关的实例, 介绍其内容的具体应用。通过本章的学习, 掌握如何建立 Qt/Embedded 的开发环境及编写 Qt/Embedded 或 Qtopia 程序的开发流程。

第 12 章 嵌入式系统工程实例。本章主要介绍文件系统的构建和烧写、数码相框、基于 Linux 的 Mplayer 解码播放器和基于 Linux 的 GPS 导航系统的开发实例。在开发这些工程实例的过程中, 熟悉 Linux 系统在嵌入式方面的应用, 熟悉其他章节的知识在开发中的具体应用。

本书主要由戴璐平、何渊仁、吴志男编写, 另外, 参加本书编写和实例测试的还有: 朱小远、谢龙汉、林伟、魏艳光、林木议、王悦阳、林伟洁、林树财、郑晓、吴苗、刘文超、刘新东。本书是在第 1 版的基础上, 结合众多读者的反馈意见进行改版, 对图书内容和实例程序代码进行了优化, 书中实例全部配有视频讲解, 使读者能够轻松掌握书中知识, 并尽快应用于实际工作中。本书实例代码和视频等素材, 请读者到华信教育资源网的本书页面下载 ([www.hxedu.com.cn](http://www.hxedu.com.cn))。

由于作者教学任务重、时间紧, 书中仍会有不妥之处, 请读者批评指正, 读者可通过电子邮件 [xucq@phei.com.cn](mailto:xucq@phei.com.cn) 与我们交流。

编著者



# 目 录

## 第一篇 嵌入式系统开发基础

|                    |      |                           |      |
|--------------------|------|---------------------------|------|
| 第 1 章 嵌入式系统介绍      | (1)  | 2.1.2 嵌入式软件的分类            | (22) |
| 1.1 嵌入式系统的概念       | (1)  | 2.1.3 嵌入式软件的开发流程          | (22) |
| 1.1.1 嵌入式系统的定义     | (1)  | 2.1.4 嵌入式软件开发工具的发展趋势      | (24) |
| 1.1.2 嵌入式系统的特点     | (2)  | 2.2 嵌入式软件的调试技术            | (25) |
| 1.2 嵌入式系统的组成       | (2)  | 2.2.1 调试技术介绍              | (25) |
| 1.2.1 嵌入式处理器       | (3)  | 2.2.2 基于 JTAG 的 ARM 系统调试  | (26) |
| 1.2.2 外围设备         | (3)  | 2.3 嵌入式软件测试技术             | (27) |
| 1.2.3 嵌入式操作系统      | (3)  | 2.3.1 宿主机—目标机开发模式         | (27) |
| 1.2.4 应用软件         | (3)  | 2.3.2 目标监控器               | (28) |
| 1.3 嵌入式处理器         | (4)  | 2.4 嵌入式系统集成开发环境           | (30) |
| 1.3.1 嵌入式处理器的分类    | (4)  | 2.4.1 ADS 的介绍             | (30) |
| 1.3.2 嵌入式微处理器      | (5)  | 2.4.2 ADS 建立工程的使用介绍       | (32) |
| 1.3.3 嵌入式微控制器      | (5)  | 2.4.3 AXD 调试器的使用介绍        | (37) |
| 1.3.4 嵌入式 DSP 处理器  | (6)  | 实例 2-1 ARM 开发环境 ADS 的使用实例 | (39) |
| 1.3.5 嵌入式片上系统      | (6)  | 2.5 本章小结                  | (43) |
| 1.3.6 选择嵌入式处理器     | (7)  | 第 3 章 嵌入式处理器体系结构          | (44) |
| 1.4 嵌入式操作系统        | (7)  | 3.1 ARM 体系结构概述            | (44) |
| 1.4.1 操作系统的概念和分类   | (8)  | 3.1.1 ARM 体系结构简介          | (45) |
| 1.4.2 实时操作系统       | (8)  | 3.1.2 ARM 体系结构的技术特征       | (47) |
| 1.4.3 常用的嵌入式操作系统   | (10) | 3.1.3 CISC 的体系结构          | (48) |
| 1.5 新型的嵌入式操作系统     | (13) | 3.1.4 RISC 的体系结构          | (48) |
| 1.5.1 Android      | (13) | 3.1.5 RISC 系统和 CISC 系统的比较 | (50) |
| 1.5.2 MontaVista   | (15) | 3.2 ARM 微处理器的分类           | (51) |
| 1.6 嵌入式系统的应用       | (15) | 3.2.1 ARM7 微处理器           | (51) |
| 1.7 嵌入式系统的发展趋势     | (18) | 3.2.2 ARM9 微处理器           | (52) |
| 1.7.1 嵌入式系统面临的挑战   | (18) |                           |      |
| 1.7.2 嵌入式系统的发展前景   | (19) |                           |      |
| 1.8 本章小结           | (20) |                           |      |
| 第 2 章 嵌入式软件开发过程与工具 | (21) |                           |      |
| 2.1 嵌入式软件开发介绍      | (21) |                           |      |
| 2.1.1 嵌入式软件开发的特殊性  | (21) |                           |      |

|                       |                   |       |        |                               |       |
|-----------------------|-------------------|-------|--------|-------------------------------|-------|
| 3.2.3                 | ARM9E 微处理器        | (52)  | 3.6.2  | 寄存器寻址                         | (68)  |
| 3.2.4                 | ARM10E 微处理器       | (53)  | 3.6.3  | 寄存器间接寻址                       | (68)  |
| 3.2.5                 | ARM11 微处理器        | (54)  | 3.6.4  | 相对寻址                          | (69)  |
| 3.2.6                 | SecurCore 微处理器    | (54)  | 3.6.5  | 堆栈寻址                          | (69)  |
| 3.2.7                 | StrongARM 微处理器    | (54)  | 3.6.6  | 块复制寻址                         | (70)  |
| 3.2.8                 | XScale 微处理器       | (55)  | 3.6.7  | 变址寻址                          | (70)  |
| 3.3                   | ARM 微处理器的应用       | (55)  | 3.6.8  | 多寄存器寻址                        | (71)  |
| 3.3.1                 | ARM 微处理器的应用<br>选型 | (55)  | 3.7    | ARM 指令集                       | (71)  |
| 3.3.2                 | S3C2410 处理器       | (56)  | 3.7.1  | ARM 指令的格式                     | (71)  |
| 3.4                   | 存储器               | (57)  | 3.7.2  | ARM 指令分类                      | (72)  |
| 3.4.1                 | 存储器简介             | (57)  | 3.7.3  | Thumb 指令介绍                    | (78)  |
| 3.4.2                 | SDRAM 操作          | (59)  | 3.7.4  | Thumb 指令分类                    | (79)  |
| 3.4.3                 | Flash             | (60)  | 3.7.5  | ARM 指令集和 Thumb 指令<br>集的区别     | (82)  |
| 3.5                   | ARM 编程模型          | (61)  | 3.8    | ARM 微处理器的异常                   | (83)  |
| 3.5.1                 | 数据类型              | (61)  | 3.8.1  | ARM 体系结构所支持的异<br>常类型          | (84)  |
| 3.5.2                 | 存储器格式             | (62)  | 3.8.2  | 异常矢量表                         | (85)  |
| 3.5.3                 | 处理器工作状态           | (62)  | 3.8.3  | 异常优先级                         | (85)  |
| 3.5.4                 | 处理器运行模式           | (63)  | 3.8.4  | 应用程序中的异常处理                    | (86)  |
| 3.5.5                 | 寄存器组织             | (63)  | 3.8.5  | 各类异常的具体描述                     | (87)  |
| 3.5.6                 | 内部寄存器             | (66)  | 3.9    | 本章小结                          | (91)  |
| 3.6                   | ARM 指令的寻址方式       | (67)  |        |                               |       |
| 3.6.1                 | 立即寻址              | (67)  |        |                               |       |
| <b>第二篇 Linux 开发入门</b> |                   |       |        |                               |       |
| 第 4 章                 | Linux 开发常用操作      | (92)  | 4.4.3  | 运行 Shell 程序                   | (116) |
| 4.1                   | Linux 系统介绍        | (92)  | 4.4.4  | Shell 程序设计的流程<br>控制           | (117) |
| 4.1.1                 | Linux 的概况         | (93)  | 4.4.5  | Shell 输入与输出                   | (121) |
| 4.1.2                 | Linux 操作系统的构成     | (94)  | 4.4.6  | bash 介绍                       | (123) |
| 4.1.3                 | Linux 常见的发行版本     | (95)  | 4.5    | 综合实例                          | (124) |
| 4.1.4                 | Linux 内核的特点       | (97)  | 实例 4-2 | 编写清除/var/log 下的<br>log 文件综合实例 | (124) |
| 4.2                   | Linux 命令的使用       | (98)  | 实例 4-3 | 编写寻找死链接文件<br>综合实例             | (127) |
| 4.3                   | vi 编辑器的使用         | (108) | 4.6    | 本章小结                          | (131) |
| 4.3.1                 | vi 编辑器的进入         | (108) | 第 5 章  | Linux 内核介绍                    | (132) |
| 4.3.2                 | 命令模式的命令           | (109) | 5.1    | 进程概述                          | (132) |
| 4.3.3                 | 末行模式的命令           | (110) | 5.1.1  | 进程结构                          | (133) |
| 实例 4-1                | vi 编辑器使用实例        | (110) | 5.1.2  | 进程的控制操作                       | (134) |
| 4.4                   | Shell 编程          | (112) |        |                               |       |
| 4.4.1                 | Shell 基础介绍        | (112) |        |                               |       |
| 4.4.2                 | Shell 程序的变量和参数    | (114) |        |                               |       |

|                                   |       |                                   |       |
|-----------------------------------|-------|-----------------------------------|-------|
| 5.1.3 进程的属性 .....                 | (136) | 环境 .....                          | (183) |
| 5.1.4 进程的创建和调度 .....              | (137) | 6.2 交叉编译的使用 .....                 | (185) |
| 5.1.5 Linux 进程命令 .....            | (139) | 6.2.1 GNU 交叉工具链的设置 .....          | (185) |
| 5.2 系统调用 .....                    | (143) | 6.2.2 ARM GNU 常用汇编语言 .....        | (188) |
| 5.2.1 系统调用概述 .....                | (143) | 6.2.3 GNU 交叉工具链的常用工具 .....        | (190) |
| 5.2.2 系统调用的进入 .....               | (144) | 6.2.4 交叉编译环境 .....                | (193) |
| 5.2.3 与进程管理相关的系统调用 .....          | (145) | 6.3 Linux 下的 C 编程 .....           | (196) |
| 5.3 管道 .....                      | (145) | 6.3.1 Linux 程序设计特点 .....          | (196) |
| 5.3.1 管道系统调用 .....                | (145) | 6.3.2 Linux 下 C 语言编码的风格 .....     | (197) |
| 5.3.2 管道的分类 .....                 | (147) | 6.3.3 Linux 程序基础 .....            | (197) |
| 实例 5-1 管道通信实例 .....               | (148) | 6.3.4 Linux 下 C 编程的库依赖 .....      | (199) |
| 5.4 信号 .....                      | (150) | 6.4 gcc 的使用与开发 .....              | (199) |
| 5.4.1 常见的信号种类 .....               | (150) | 6.4.1 gcc 简介和使用 .....             | (199) |
| 5.4.2 系统调用函数 .....                | (151) | 6.4.2 gcc 选项 .....                | (200) |
| 5.4.3 信号的处理 .....                 | (152) | 6.4.3 gcc 的错误类型 .....             | (203) |
| 5.4.4 信号与系统调用的关系 .....            | (152) | 实例 6-1 gcc 编译器环境的应用 .....         | (204) |
| 实例 5-2 信号实例 .....                 | (153) | 6.5 gdb 调试器的介绍和使用 .....           | (205) |
| 5.5 信号量 .....                     | (154) | 6.5.1 gdb 调试器的使用 .....            | (205) |
| 5.5.1 信号量概述 .....                 | (154) | 6.5.2 在 gdb 中运行程序 .....           | (206) |
| 5.5.2 相关的数据结构 .....               | (155) | 6.5.3 暂停和恢复程序运行 .....             | (208) |
| 5.5.3 相关的函数 .....                 | (157) | 6.5.4 远程调试 .....                  | (211) |
| 实例 5-3 信号量实例 .....                | (158) | 实例 6-2 gdb 调试器环境的应用 .....         | (211) |
| 5.6 共享内存 .....                    | (163) | 6.6 GNU make 和 Makefile 的使用 ..... | (213) |
| 5.6.1 共享内存原理 .....                | (163) | 6.6.1 Makefile 的基本结构 .....        | (214) |
| 5.6.2 共享内存对象的结构 .....             | (164) | 6.6.2 Makefile 的变量 .....          | (215) |
| 5.6.3 相关的函数 .....                 | (165) | 6.6.3 Makefile 的隐含规则 .....        | (217) |
| 实例 5-4 共享内存实例 .....               | (166) | 6.6.4 Makefile 的命令使用 .....        | (220) |
| 5.7 消息队列 .....                    | (171) | 6.6.5 Makefile 的函数使用 .....        | (221) |
| 5.7.1 有关的数据结构 .....               | (171) | 6.6.6 Makefile 文件的运行 .....        | (223) |
| 5.7.2 相关的函数 .....                 | (174) | 6.6.7 Makefile 规则书写命令 .....       | (225) |
| 实例 5-5 消息队列实例 .....               | (176) | 实例 6-3 Makefile 的命令使用 .....       | (231) |
| 5.8 综合实例 .....                    | (179) | 6.7 autoconf 和 automake 的使用 ..... | (233) |
| 实例 5-6 多线程编程实例 .....              | (179) |                                   |       |
| 5.9 本章小结 .....                    | (180) |                                   |       |
| 第 6 章 Linux 开发环境的构建 .....         | (181) |                                   |       |
| 6.1 建立 Linux 开发环境概述 .....         | (181) |                                   |       |
| 6.1.1 Cygwin 开发环境 .....           | (181) |                                   |       |
| 6.1.2 VMware Workstation 开发 ..... |       |                                   |       |



|                |                                    |       |        |                         |       |
|----------------|------------------------------------|-------|--------|-------------------------|-------|
| 6.7.1          | autoconf 的使用                       | (233) | 6.8    | 综合实例                    | (238) |
| 6.7.2          | Makefile.am 的编写                    | (236) | 实例 6-4 | gcc 编译器的综合实例            | (238) |
| 6.7.3          | automake 的使用                       | (236) | 实例 6-5 | gdb 调试器的综合实例            | (241) |
| 6.7.4          | 使用 automake 和 autoconf 产生 Makefile | (237) | 实例 6-6 | Makefile 的综合实例          | (244) |
| 6.7.5          | 自动生成 Makefile 的方法                  | (237) | 6.9    | 本章小结                    | (246) |
| 第三篇 嵌入式系统移植与构建 |                                    |       |        |                         |       |
| 第 7 章          | Bootloader 的使用                     | (247) | 实例 7-3 | Bootloader 设计实例         | (283) |
| 7.1            | Bootloader 概述                      | (248) | 7.6    | 本章小结                    | (285) |
| 7.1.1          | Bootloader 的作用                     | (248) | 第 8 章  | Linux 内核裁剪与移植           | (286) |
| 7.1.2          | Bootloader 的功能                     | (249) | 8.1    | Linux 移植简介              | (286) |
| 7.1.3          | Bootloader 的种类                     | (250) | 8.1.1  | Linux 可移植性发展            | (287) |
| 7.1.4          | Bootloader 的工作模式                   | (251) | 8.1.2  | Linux 的移植性              | (287) |
| 7.1.5          | Bootloader 的启动方式                   | (251) | 8.2    | Linux 内核结构              | (288) |
| 7.1.6          | Bootloader 的启动流程                   | (253) | 8.2.1  | Linux 内核组成              | (288) |
| 7.1.7          | Bootloader 与主机的通信                  | (253) | 8.2.2  | 子系统相互间的关系               | (292) |
| 7.2            | vivi 的移植                           | (253) | 8.2.3  | 系统数据结构                  | (293) |
| 7.2.1          | vivi 的常用命令和文件结构                    | (254) | 8.2.4  | Linux 内核源代码             | (293) |
| 7.2.2          | vivi 第一阶段的分析                       | (255) | 8.3    | Linux 内核配置              | (297) |
| 7.2.3          | vivi 第二阶段的分析                       | (261) | 实例 8-1 | Linux 内核配置实例            | (297) |
| 7.2.4          | vivi 的配置与编译                        | (263) | 8.4    | Linux 操作系统移植介绍          | (300) |
| 7.3            | U-boot 的移植                         | (264) | 8.4.1  | Linux 系统移植的两大部分         | (300) |
| 7.3.1          | U-boot 常用命令和源代码目录结构                | (264) | 8.4.2  | 内核文件的修改                 | (302) |
| 7.3.2          | U-boot 支持的主要功能                     | (268) | 8.4.3  | 系统移植所必需的环境              | (306) |
| 7.3.3          | U-boot 的编译和添加命令                    | (268) | 8.5    | 综合实例                    | (309) |
| 7.3.4          | U-boot 的启动介绍                       | (272) | 实例 8-2 | 编译 Linux 内核应用实例         | (309) |
| 7.3.5          | U-boot 的移植和使用                      | (273) | 实例 8-3 | Linux 内核的烧写实例           | (311) |
| 7.3.6          | U-boot 的启动过程                       | (275) | 实例 8-4 | 使用 KGDB 构建 Linux 内核调试环境 | (312) |
| 7.3.7          | U-boot 的调试                         | (276) | 8.6    | 本章小结                    | (321) |
| 7.4            | 其他常见的 Bootloader                   | (278) | 第 9 章  | Linux 根文件系统的构建          | (322) |
| 7.5            | 综合实例                               | (280) | 9.1    | Linux 文件系统概述            | (322) |
| 实例 7-1         | vivi 编译实例                          | (280) | 9.1.1  | Linux 文件系统的特点           | (322) |
| 实例 7-2         | U-boot 在 S3C2410 上的移植实例            | (281) | 9.1.2  | 其他常见的嵌入式文件系统            | (324) |

|                                 |       |                                      |       |
|---------------------------------|-------|--------------------------------------|-------|
| 9.1.3 Linux 根文件目录结构             | (326) | 实例 9-2 构建根文件系统                       | (339) |
| 9.1.4 Linux 文件属性介绍              | (326) | 9.4 配置 yaffs 文件                      | (341) |
| 9.2 使用 BusyBox 生成工具集            | (327) | 9.4.1 yaffs 文件系统设置                   | (342) |
| 9.2.1 BusyBox 概述                | (327) | 9.4.2 yaffs 文件系统测试                   | (344) |
| 9.2.2 BusyBox 进程和用户程序<br>启动过程   | (328) | 9.5 综合实例                             | (345) |
| 9.2.3 编译/安装 BusyBox             | (329) | 实例 9-3 制作/使用 yaffs 文件系<br>统映像文件      | (345) |
| 实例 9-1 用 BusyBox 建立简单的<br>根文件系统 | (333) | 实例 9-4 制作/使用 jffs2 文件系<br>统映像文件      | (347) |
| 9.3 构建根文件系统                     | (335) | 9.6 本章小结                             | (349) |
| 9.3.1 根文件系统制作流程                 | (335) |                                      |       |
| 第四篇 嵌入式系统开发                     |       |                                      |       |
| 第 10 章 设备驱动程序开发                 | (350) | 结构                                   | (371) |
| 10.1 设备驱动程序概述                   | (351) | 10.5.2 网络设备驱动程序模块<br>分析              | (372) |
| 10.1.1 驱动程序的简介                  | (351) | 10.5.3 网络设备驱动程序的实现<br>模式             | (378) |
| 10.1.2 设备分类                     | (351) | 10.5.4 网络设备驱动程序的数据<br>结构             | (379) |
| 10.1.3 设备号                      | (352) | 10.6 综合实例                            | (383) |
| 10.1.4 设备节点                     | (353) | 实例 10-1 键盘驱动开发实例                     | (383) |
| 10.1.5 驱动层次结构                   | (353) | 实例 10-2 I2C 总线驱动的编写<br>实例            | (386) |
| 10.1.6 设备驱动程序的特点                | (354) | 实例 10-3 TFT-LCD 显示驱动<br>实例           | (390) |
| 10.2 设备驱动程序与文件系统                | (355) | 10.7 本章小结                            | (395) |
| 10.2.1 设备驱动程序与文件系<br>统的关系       | (355) | 第 11 章 嵌入式常用 GUI 开发                  | (396) |
| 10.2.2 设备驱动程序与操作系<br>统的关系       | (355) | 11.1 嵌入式系统中的 GUI 简介                  | (397) |
| 10.2.3 Linux 设备驱动程序的<br>接口      | (356) | 11.1.1 嵌入式 GUI 系统的<br>介绍             | (397) |
| 10.2.4 设备驱动程序开发的<br>基本函数        | (361) | 11.1.2 基于嵌入式 Linux 的 GUI<br>系统底层实现基础 | (399) |
| 10.2.5 Linux 驱动程序的加载            | (362) | 11.1.3 嵌入式 GUI 系统的分析<br>与比较          | (399) |
| 10.3 设备驱动程序的使用                  | (366) | 11.2 嵌入式系统下 MiniGUI 的<br>实现          | (401) |
| 10.3.1 驱动程序模块的加载                | (366) | 11.2.1 图形用户界面 MiniGUI<br>简介          | (401) |
| 10.3.2 创建设备文件                   | (367) | 11.2.2 MiniGUI 的发布版本                 | (403) |
| 10.3.3 使用设备                     | (367) |                                      |       |
| 10.4 网络设备基础知识                   | (367) |                                      |       |
| 10.4.1 网络协议                     | (367) |                                      |       |
| 10.4.2 网络设备接口基础                 | (369) |                                      |       |
| 10.5 网络设备驱动程序的架构                | (371) |                                      |       |
| 10.5.1 网络设备驱动程序体系               |       |                                      |       |

|                                                   |                                              |
|---------------------------------------------------|----------------------------------------------|
| 11.2.3 MiniGUI 在 S3C2410 处<br>理器上的移植过程····· (406) | 12.1 文件系统的生成与烧写····· (446)                   |
| 11.3 Qt/Embedded 嵌入式图形开发<br>基础····· (409)         | 12.1.1 yaffs 文件系统的制作与<br>生成····· (447)       |
| 11.3.1 Qt/Embedded 开发环境的<br>安装····· (409)         | 12.1.2 jffs2 文件系统的制作与<br>生成····· (451)       |
| 11.3.2 Qt/Embedded 底层支持及<br>实现代码分析····· (413)     | 12.2 基于 Linux 的数码相框····· (454)               |
| 11.3.3 Qt/Embedded 信号和插槽<br>机制····· (414)         | 12.2.1 系统需求分析····· (454)                     |
| 11.3.4 Qt/Embedded 窗口部件··· (417)                  | 12.2.2 系统总体设计····· (455)                     |
| 11.3.5 Qt/Embedded 图形界面<br>编程····· (420)          | 12.2.3 软件设计实现····· (456)                     |
| 11.3.6 Qt/Embedded 对话框<br>设计····· (421)           | 12.2.4 软硬件集成····· (463)                      |
| 11.3.7 数据库····· (422)                             | 12.3 基于 Linux 的 MPlayer 解码播<br>放器····· (464) |
| 实例 11-1 Qt/Embedded 图形开<br>发应用实例····· (425)       | 12.3.1 可行性分析报告····· (464)                    |
| 11.4 Qtopia 移植····· (426)                         | 12.3.2 系统总体设计····· (465)                     |
| 11.4.1 Qtopia 简介····· (426)                       | 12.3.3 软件总体设计····· (466)                     |
| 11.4.2 交叉编译、安装 Qtopia·· (426)                     | 12.3.4 软件详细设计····· (470)                     |
| 实例 11-2 Qtopia 移植应用<br>实例····· (428)              | 12.3.5 软硬件集成····· (480)                      |
| 11.5 Qt/Embedded 应用开发····· (429)                  | 12.4 基于 Linux 的 GPS 导航系统的<br>开发····· (481)   |
| 11.5.1 嵌入式硬件开发平台的<br>选择····· (429)                | 12.4.1 嵌入式开发流程····· (482)                    |
| 11.5.2 Qt/Embedded 常用工具的<br>介绍····· (431)         | 12.4.2 GPS 导航定位系统的系统<br>定义····· (484)        |
| 11.5.3 交叉编译 Qt/Embedded<br>的库····· (432)          | 12.4.3 GPS 导航系统的可行性分<br>析报告····· (489)       |
| 11.5.4 Qt/E 程序的编译与<br>执行····· (433)               | 12.4.4 GPS 导航系统需求分析·· (490)                  |
| 实例 11-3 Qt/Embedded 实战<br>演练····· (434)           | 12.4.5 GPS 导航系统总体设计<br>实现····· (493)         |
| 11.6 综合实例····· (438)                              | 12.4.6 GPS 导航系统硬件设计<br>实现····· (494)         |
| 实例 11-4 Hello, Qt/Embedded<br>应用程序····· (438)     | 12.4.7 GPS 导航系统软件概括<br>设计····· (498)         |
| 实例 11-5 基本绘图应用程序的<br>编写····· (441)                | 12.4.8 GPS 导航系统软件详细<br>设计····· (498)         |
| 11.7 本章小结····· (445)                              | 12.4.9 GPS 导航系统数据库的<br>配置设计····· (525)       |
| 第 12 章 嵌入式系统工程实例····· (446)                       | 12.4.10 GPS 导航系统软件实现·· (537)                 |
|                                                   | 12.4.11 GPS 导航系统软硬件<br>集成····· (538)         |
|                                                   | 12.4.12 GPS 导航系统功能性能<br>测试····· (540)        |
|                                                   | 12.5 本章小结····· (541)                         |



# 第一篇 嵌入式系统开发基础

## 第 1 章 嵌入式系统介绍

嵌入式技术是计算机技术、半导体技术和微电子技术等多种先进技术的融合。在 PC 时代,嵌入式技术成为最有生命力的技术之一,它被应用在军事国防、医疗卫生、科学教育等各方面。



### 本章内容

- 嵌入式系统的基本定义和特点
- 嵌入式系统的组成部分和处理器
- 常见的嵌入式操作系统
- 新型的嵌入式操作系统
- 嵌入式系统的应用领域
- 嵌入式系统的发展趋势

### 1.1 嵌入式系统的概念

嵌入式系统技术是目前最热门的技术之一。随着 21 世纪手机、PDA、MP4 等大量数码产品的广泛使用,嵌入式系统从早期工业控制系统的应用渐渐渗透到人们工作和生活中,从制造工业、过程控制、仪器仪表、汽车船舶到通信及各种消费类数码产品的各个应用领域。它将无所不在,为人类生产带来革命性的发展。

#### 1.1.1 嵌入式系统的定义

关于嵌入式系统的定义,一般认为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。它一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统及用户的应用程序 4 部分组成, 用于实现对其他设备的控制、监视或管理等功能。按照这种定义, 典型的嵌入式系统有各种使用 x86 的小型嵌入式工控主板, 它们在各种自动化设备、数字机械产品中有非常广阔的应用空间; 另外一大类是使用 Intel、Samsung、Motorola 等专用芯片构成的小系统, 它不仅在新兴的消费电子、通信设备和仪表等方面获得了巨大的发展应用空间, 甚至有取代传统的工控机的趋势。

### 1.1.2 嵌入式系统的特点

嵌入式系统可以称为后 PC 时代和后网络时代的新秀, 与传统的通用计算机、数字产品相比, 利用嵌入式技术的产品特点如下。

(1) 由于嵌入式系统采用的是微处理器, 独立的操作系统, 实现相对单一的功能, 所以往往不需要大量的外围器件, 因而在体积上、功耗上有其自身的优势。一般来说, 一个使用 Windows CE 的 PDA, 仅靠机内电源就可以使用几天, 而任何一台笔记本电脑仅能够维持几小时左右。

(2) 嵌入式系统是将计算机技术、电子技术和材料技术与各个行业的具体应用结合后的产物, 是一门综合技术学科。由于空间和各种资源相对不足, 嵌入式系统的硬件和软件都必须高效率地设计, 力争在同样的硅片面积上实现更高的性能, 这样才能在具体应用中对外处理器的选择上更具有竞争力。

(3) 嵌入式系统是一个软硬件高度结合的产物。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。片上系统、板上系统的实现, 使得以 PDA、手机等为代表的这类产品拥有更加熟悉的操作界面和操作方式, 比传统的电子记事本等功能更加完善、实用。

(4) 由于嵌入式系统一般应用于小型电子装置, 系统资源相对有限, 所以, 内核比传统的操作系统要小得多。例如, ENEA 公司的 OSE 分布式系统, 内核只有 5KB, 而 Windows 的内核则要大多得多。

(5) 嵌入式系统一般没有系统软件和应用软件的明显区分, 不要求其功能设计及实现上过于复杂, 这样一方面利于控制系统成本, 另一方面也利于实现系统安全。

(6) 嵌入式系统不可垄断性。从计算机的发展来看, 现在的计算机市场基本上已被 Wintel 联盟垄断, 而嵌入式系统工业的基础是以应用为中心的“芯片”设计和面向应用的软件产品开发, 是针对不同的系统、不同的产品来进行的相应地开发, 故这一市场不可能被一家或几家大公司垄断。

## 1.2 嵌入式系统的组成

嵌入式系统通常由嵌入式处理器、外围设备、嵌入式操作系统和应用软件等几大部分组成, 如图 1-1 所示。

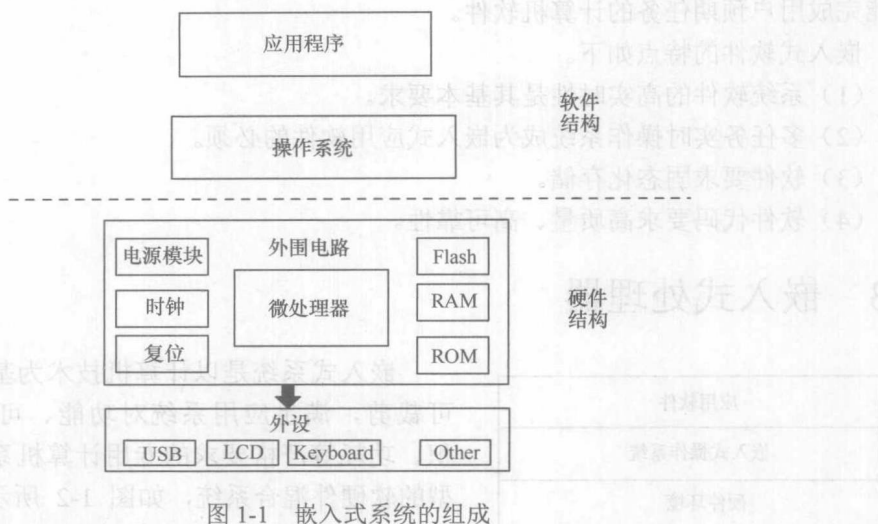


图 1-1 嵌入式系统的组成

### 1.2.1 嵌入式处理器

嵌入式处理器属于嵌入式系统的核心部件。嵌入式处理器与通用处理器的最大不同点在于嵌入式处理器大多工作在为特定用户群设计的系统中。它有利于嵌入式系统设计趋于小型化，并具有高效率、高可靠性等特征。

### 1.2.2 外围设备

外围设备是指在一个嵌入式系统中，除了嵌入式处理器以外的完成存储、通信、调试、显示等辅助功能的其他部件。

根据外围设备的功能可分为以下 3 类。

- (1) 人机交互：LCD、键盘和触摸屏等人机交互设备。
- (2) 存储器：动态存储器、非易失型存储器和静态易失型存储器。其中，动态存储器以可擦写次数多、存储速度快、容量大及价格低等优点在嵌入式领域得到了广泛的应用。
- (3) 接口：应用最为广泛的包括并口、I<sup>2</sup>C（InterIC）总线接口、USB 通用串行总线接口、RS-232 串口、IrDA 红外接口、SPI 串行外围设备接口和 Ethernet 网口等。

### 1.2.3 嵌入式操作系统

嵌入式操作系统是用来管理中断处理、任务间通信、存储器分配和定时器响应的软件模块集合。嵌入式操作系统常常有实时要求，所以，嵌入式操作系统往往又是实时操作系统。

### 1.2.4 应用软件

嵌入式系统的应用软件是针对特定的实际专业领域的，基于相应的嵌入式硬件平台，

并能完成用户预期任务的计算机软件。

嵌入式软件的特点如下。

- (1) 系统软件的高实时性是其基本要求。
- (2) 多任务实时操作系统成为嵌入式应用软件的必须。
- (3) 软件要求固态化存储。
- (4) 软件代码要求高质量、高可靠性。

## 1.3 嵌入式处理器

|         |
|---------|
| 应用软件    |
| 嵌入式操作系统 |
| 硬件环境    |

图 1-2 系统层次图

嵌入式系统是以计算机技术为基础, 软件、硬件可裁剪, 满足应用系统对功能、可靠性、成本、体积、功耗等严格要求的专用计算机系统。它是一种典型的软硬件混合系统, 如图 1-2 所示。由下向上可分为 3 个组成部分: 硬件环境、嵌入式操作系统和应用软件。

### 1.3.1 嵌入式处理器的分类

硬件环境是嵌入式实时操作系统和应用程序运行的硬件平台。由于嵌入式系统是嵌入于宿主设备的计算机系统, 完成宿主设备的功能要求, 所以, 不同的应用通常会有不同的硬件环境。嵌入式系统的核心部件是嵌入式处理器, 目前, 全世界的处理器品种数量已经超过 1000 种, 流行体系结构有 30 多个系列。根据其发展现状, 大致可以分为以下 4 种类型。

#### (1) 嵌入式微处理器 (EMPU)

其基础是通用计算机中的 CPU。在应用中将微处理器装配在专门设计的电路板上, 只保留和嵌入式应用有关的功能, 可以大幅度减小系统体积和功耗。嵌入式 CPU 在功能上和标准 CPU 基本一致, 具有体积小、质量小、成本低、可靠性高的优点, 但在电路板上必须包括 ROM、RAM、总线接口和相关外设。嵌入式 CPU 及其存储器、总线、外设等被安装在一块电路板上, 称为单板计算机。嵌入式 CPU 目前主要有 MIPS, ARM, PowerPC, SC-400, 386EX 等。

#### (2) 嵌入式微控制器 (MCU)

嵌入式微控制器是以微处理器内核为核心, 内部集成 EPROM、RAM、总线及其逻辑、定时计数器、WatchDog、I/O、串行口、脉宽调制输出 PWM、A/D、D/A、FlashRAM、EEPROM 等各种必要功能和外设。MCU 是目前嵌入式系统工业的主流, 其最大的特点是单片化、体积小、成本低, 最具代表性的有 51/52, 96/196, C166/167, MC68HC, M16C, XA, AVR 等系列。

#### (3) 嵌入式 DSP 处理器 (DSP)

它对系统结构和指令进行了特殊设计, 适合于执行 DSP 算法, 编译效率高, 指令执行速度很快。主要应用在数字滤波、谱分析、生物信息识别终端、实时语音解压、图像处理、网络通信、虚拟现实等高速数据处理领域。嵌入式 DSP 处理器主要是 TexasInstrument

的 TMS320 系列和 Motorola 的 DSP56000 系列。

#### (4) 嵌入式片上系统 (SoC)

随着 EDI 的推广、VLSI 设计的进步及半导体工艺的迅速发展, 将整个嵌入式系统或其大部分集成到一块或几块芯片中成为现实, 即片上系统 (SystemonChip)。它以通用 CPU 内核为标准库, 用 VHDL 等语言描述。嵌入式片上系统分为通用和专用两类, 通用系列包括 Infineon 的 TriCore, Motorola 的 M-Core 等。专用的片上系统一般用于某个或者某类系统中, 不为一般用户所知, 目前有 Siemens 的 TriCore, Motorola 的 M-Core, 英国的 ARM 核及产品化的 C8051F 等。

### 1.3.2 嵌入式微处理器

在应用中, 将微处理器装配在专门设计的电路板上, 只保留和嵌入式应用有关的母板功能, 这样可以大幅度减小系统体积和功耗。嵌入式微处理器虽然在功能上和标准微处理器基本是一样的, 但在工作温度、抗电磁干扰、可靠性等方面一般都有提高。

与工业控制计算机相比, 嵌入式微处理器具有体积小、质量小、成本低、可靠性高的优点, 但是在电路板上必须配置 ROM、RAM、总线接口和各种外设等器件, 降低了系统的可靠性, 技术保密性也较差。嵌入式微处理器及其存储器、总线、外设等安装在一块电路板上, 称为单板计算机。

嵌入式微处理器一般具备以下 4 个特点。

① 可扩展的处理器结构, 能最迅速地开展出满足应用的最高性能的嵌入式微处理器。

② 具有功能很强的存储区保护功能。这是由于嵌入式系统的软件结构已模块化, 而为了避免在软件模块之间出现错误的交叉作用, 需要设计强大的存储区保护功能, 同时也有利于软件诊断。

③ 对实时多任务有很强的支持能力, 能完成多任务并且有较短的中断响应时间, 从而使内部的代码和实时内核的执行时间减少到最低限度。

④ 嵌入式微处理器功耗必须很低, 尤其是用于便携式的无线及移动的计算和通信设备中靠电池供电的嵌入式系统更是如此。

### 1.3.3 嵌入式微控制器

微控制器是将微型计算机的主要部分集成在一个芯片上的单芯片微型计算机。微控制器诞生于 20 世纪 70 年代中期, 经过 20 多年的发展, 其成本越来越低, 而性能越来越强大, 这使其应用无处不在, 遍及各个领域。例如, 电机控制、消费类电子、游戏设备、楼宇安全与门禁控制、工业控制与自动化和白色家电等。

微控制器可从不同方面进行分类: 根据内嵌程序存储器的类别可分为 OTP、掩膜、EPROM/EEPROM 和 Flash 闪存; 根据数据总线宽度可分为 8 位、16 位和 32 位机; 根据存储器结构可分为 Harvard 结构和 VonNeumann 结构; 根据指令结构又可分为 CISC 和 RISC 微控制器。

针对 4 位 MCU, 大部分供货商采用单生产, 目前, 4 位 MCU 大部分应用在儿童玩



具、充电器、温湿度计、计算器、车用防盗装置、呼叫器、无线电话、遥控器等领域；8 位 MCU 大部分应用在变频式冷气机、传真机、来电辨识器、电表、电动玩具机、电话录音机、CRTDisplay、键盘及 USB 等领域；16 位 MCU 大部分应用在数码相机、移动电话及摄录放映机等领域；32 位 MCU 大部分应用在 PDA、HPC、STB、Modem、GPS、Router、工作站、ISDN 电话、激光打印机与彩色传真机领域；64 位 MCU 大部分应用在高阶工作站、多媒体互动系统、高级电视游乐器及高级终端机等领域。

### 1.3.4 嵌入式 DSP 处理器

嵌入式 DSP 处理器是一种独特的微处理器，是以数字信号来处理大量信息的器件。它不仅具有可编程性，而且其实时运行速度可达每秒数以千万条复杂指令程序，远远超过通用微处理器，是数字化电子世界中日益重要的计算机芯片。

DSP 芯片也称数字信号处理器，是一种特别适合于进行数字信号处理运算的微处理器具，其主机应用是实时快速地实现各种数字信号处理算法。根据数字信号处理的要求，DSP 芯片一般具有如下主要特点。

- ① 可以并行执行多个操作。
- ② 程序和数据空间分开，可以同时访问指令和数据；快速的中断处理和硬件 I/O 支持。
- ③ 具有在单周期内操作的多个硬件地址产生器。
- ④ 片内具有快速 RAM，通常可通过独立的数据总线在两块中同时访问。
- ⑤ 具有低开销或无开销循环及跳转的硬件支持。
- ⑥ 支持流水线操作，使取指、译码和执行等操作可以重叠执行。

DSP 是模拟电子时代向数字电子时代前进的理论基础，而 DSP 是随着数字信号处理而专门设计的可编程处理器，是现代电子技术、计算机技术和信号处理技术相结合的产物。随着信息处理技术的飞速发展，DSP 在电子信息、通信、软件无线电、自动控制、仪器仪表、信息家电等高科技领域得到了越来越广泛的应用。DSP 不仅快速实现了各种数字信号处理算法，而且拓宽了数字信号处理的应用范围。DSP 的功能将越来越强大，应用范围也将越来越广泛。

### 1.3.5 嵌入式片上系统

嵌入式片上系统 (SoC) 是一个有专用目标的集成电路，其中包含完整系统并有嵌入软件的全部内容。同时它又是一种技术，用于实现从确定系统功能开始，到软/硬件划分，并完成设计的整个过程。从狭义角度讲，它是信息系统核心的芯片集成，是将系统关键部件集成在一块芯片上；从广义角度讲，国内外学术界一般倾向将 SoC 定义为将微处理器、模拟 IP 核、数字 IP 核和存储器集成在单一芯片上，它通常是客户定制的，或是面向特定用途的标准产品。

系统级芯片的构成可以是系统级芯片控制逻辑模块、微处理器/微控制器 CPU 内核模块、数字信号处理器 DSP 模块、嵌入的存储器模块和外部进行通信的接口模块、含有 ADC/DAC 的模拟前端模块、电源提供和功耗管理模块，对于一个无线 SoC 还有射