

- 中国高等职业技术教育研究会推荐
- 高职高专电子、通信类专业“十一五”规划教材

基于ARM9嵌入式Linux系统开发原理与实践

马小陆 主编



西安电子科技大学出版社
<http://www.xduph.com>

□ 中国高等职业技术教育研究会推荐

高职高专电子、通信类专业“十一五”规划教材

基于 ARM9 的嵌入式 Linux 系统 开发原理与实践

马小陆 主编

西安电子科技大学出版社

内 容 简 介

本书从嵌入式 Linux 系统入门必备知识点、嵌入式 Linux 系统的构建、驱动程序开发和应用程序设计四个方面着手。全书分为三篇：基础篇、提高篇和实践篇。基础篇包括第 1~5 章，主要介绍嵌入式 Linux 的基本知识、相关原理和实践；提高篇包括第 6~10 章，主要介绍如何构建嵌入式 Linux 系统；实践篇包括第 11~13 章，主要讲述驱动程序开发和应用程序的设计。

本书可作为高职高专电子、电气、控制、计算机、通信等专业的教学用书，也可作为从事嵌入式 Linux 系统开发的技术人员的参考书。

图书在版编目（CIP）数据

基于 ARM9 的嵌入式 Linux 系统开发原理与实践 / 马小陆主编. —西安：西安电子科技大学出版社，2011.9

高职高专电子、通信类专业“十一五”规划教材

ISBN 978-7-5606-2500-3

I. ① 基… II. ① 马… III. ① 微处理器，ARM—高等职业教育—教材 ② Linux 操作系统—系统设计—高等职业教育—教材 IV. ① TP332 ② TP316.89

中国版本图书馆 CIP 数据核字（2011）第 177795 号

策 划 薛 媛

责任编辑 薛 媛 张晓燕

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 西安文化彩印厂

版 次 2011 年 10 月第 1 版 2011 年 10 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 23

字 数 545 千字

印 数 1~3000 册

定 价 35.00 元

ISBN 978-7-5606-2500-3/TP·1245

XDUP 2792001-1

如有印装问题可调换

本社图书封面为激光防伪覆膜，谨防盗版。

前 言

嵌入式 Linux 系统在工业生产控制、智能仪表、信息家电、网络通信等领域中都有着广泛的应用。在最近几年，嵌入式 Linux 系统取得了前所未有的发展，多媒体手机、数字个人助理 PDA、数字导航仪、MP3/MP4、网络路由器等无一不是嵌入式系统的一种。可以相信随着数字多媒体时代的来临，嵌入式 Linux 系统将会有更加广阔的发展前景。也正是因为如此，已经有越来越多的开发人员开始从单片机和 DSP 系统开发渐渐地向基于 ARM 处理器的嵌入式 Linux 系统开发过渡。

综观当前国内图书市场有关嵌入式 Linux 系统学习的教材类书籍，虽不乏非常优秀的教材，但缺少一种适合引导读者一步一步深入学习和开发嵌入式 Linux 系统的教材。已出版的教材中，有些教材偏重于理论的讲述，有些教材偏重于实践的讲述。这样导致的结果是：前者读者看过之后，面对具体的硬件开发平台无从下手；而后者因缺乏系统理论知识的讲述、内容组织不合理和操作过程不详尽，读者看过之后，依然无法在硬件平台上一步一步地开发自己的嵌入式 Linux 系统。鉴于此，本书编者基于自己多年有关嵌入式 Linux 系统开发和教学经验，编写了此教材，以飨读者。

本书立足于将原理和实践内容相结合，一步一步指导读者在掌握理论知识的基础上，在 ARM9 开发平台上建立自己的嵌入式 Linux 系统，较全面地了解嵌入式 Linux 系统软件开发的全部过程。

全书分为三篇：基础篇、提高篇和实践篇。其中，基础篇包括第 1~5 章；提高篇包括第 6~10 章；实践篇包括第 11~13 章。

第 1 章介绍嵌入式系统和嵌入式操作系统的概况，讲述嵌入式 Linux 操作系统的一些特征，简单介绍嵌入式 Linux 的特殊开发环境，总体上说明嵌入式 Linux 系统开发的特点。通过本章内容的学习，读者应该对嵌入式 Linux 系统有一个整体的认识，并能理解嵌入式 Linux 开发的要点，对后面内容的学习有一个宏观的思路。

第 2 章介绍发行版 Linux 操作系统的安装方法，主要讲述 Linux 的一些基本知识，重点介绍 Shell 编程。通过本章的学习和操作，读者应该能够对 Linux 操作系统有一个常识性的了解，为后续章节的学习打下基础。

第 3 章讲述发行版 Linux 上常用的编程工具。这部分内容是以后嵌入式 Linux 软件开发的基础，读者可以对照本章内容上机练习，加深对 Linux 的理解。

第 4 章讲述交叉开发环境的相关知识，针对 UP-NETARM2410 开发平台，具体讲述开发环境的建立过程。读者可结合建立过程来理解交叉开发环境每一步的建立原因，从而可加深对交叉开发环境的理解。

第 5 章讲述 ARM 概述、ARM 微处理器结构、ARM 程序组织形式的硬件架构和 ARM 的编程模型，较详细讲述了 ARM 指令集，最后讲述 ARM 汇编程序设计。读者可以从实际编程中去理解 ARM 处理器及 ARM 指令等相关内容。

第 6 章介绍一步一步制作交叉开发工具链的详细过程，以加深读者对编译原理和交叉

开发环境的相关概念的理解。

第7章介绍 Bootloader 概念和 Bootloader 通用共性,包括 Bootloader 所支持的 CPU 和嵌入式板、安装媒介、控制方式,操作模式和与主机通信方式等。针对特定引导程序 vivi 进行了分析,并对其配置、编译和启动原理进行了深入分析。最后针对博创 2410 开发板对 vivi 进行移植和实验验证。通过本章的学习,读者能够了解引导程序的工作原理和作用,为下一步内核的移植打下基础。

第8章介绍 Linux 内核、Linux 内核代码结构和组织结构原理,读者可通过本章中的一个配置实例去实践其配置过程,加深对内核配置原理的理解。最后通过移植 Linux2.6.14.1 到博创 2410 开发板进行了测试验证。通过本章的学习,读者能够了解内核的工作原理、作用以及 Linux 内核移植的相关内容和具体过程。

第9章介绍文件系统、根文件系统和根文件系统制作的原理,最后在博创 2410 开发板上制作了 cramfs 根文件系统,通过本章的学习,读者能够了解根文件系统的工作原理、作用以及嵌入式根文件系统制作的相关内容和具体过程。

第10章介绍设备驱动程序的相关概念、设备驱动程序接口及其实现过程,通过一个字符设备驱动实例讲述如何编写一个驱动程序。最后讲述 Linux 2.6 内核的驱动程序原理。通过本章的学习,读者能够理解驱动程序的工作原理、作用及它和内核之间的关系,为在下一篇中学习驱动程序开发实例打下基础。

第11章介绍 TFT-LCD 及 LCD 液晶显示器的工作原理等基础知识,讲述 Linux2.2 内核以后版本中出现的一种驱动程序接口 FrameBuffer 的工作原理、内核中 FrameBuffer 相关重要数据结构和 S3C2410 与 LCD 相关的数据结构等,最后针对博创 2410 开发板上的 TFT-LCD,在 Linux2.6 内核下对 TFT-LCD 驱动程序进行了移植,并进行了测试验证。通过本章的学习,读者能够加深对驱动程序原理及驱动与内核之间关系的理解。

第12章介绍 USB 的工作原理等基础知识,讲述 Linux2.6 内核对 USB 驱动的支持原理;最后针对博创 2410 开发板上的 USB 接口,在 Linux2.6 内核下对 USB 驱动程序进行了移植,并进行了测试验证。通过本章的学习,读者能够加深对驱动程序原理及其与内核之间关系的理解。

第13章介绍图形用户界面 GUI、Qt 和 Qt/E 的相关知识,通过一个加法程序实例讲述了 Qt 的编程方法并对重要代码进行了分析,构建了 Qt2 和 Qt2/E 开发环境,将 Qt2/E 程序移植到博创 2410 开发板上,并在开发板的 TFT-LCD 液晶上显示了 Qt2/E 程序结果,对整个系统进行了重新构建,最后讲述 Qtopia 开发环境建立的过程。通过本章的学习,读者能够加深对系统的构建、驱动程序原理、应用程序及它们之间关系的理解,从而能够全面理解嵌入式 Linux 系统开发所涉及的相关内容。

本书立足于将原理和实践内容相结合,一步一步指导读者在掌握理论知识基础之后,在 ARM9 开发平台上建立自己的嵌入式 Linux 系统。因此,本书适合于下列人员阅读:

- 刚刚进入嵌入式 Linux 系统学习的学生或开发人员。
- 熟悉单片机或 DSP 的开发,想转到嵌入式 Linux 系统开发的技术人员。
- 接触过嵌入式 Linux 系统开发,但对其本质原理不是很清楚的学生或开发人员。
- 想深入学习嵌入式 Linux 系统的学生或开发人员。

本书可作为高职高专电子、电气、控制、计算机通信等专业学生学习嵌入式 Linux 的教

材，也可作为广大嵌入式领域科研和工程技术人员的参考书。

在成书过程中，感谢北京博创科技有限公司，本书所有程序是在其生产的 ARM9 硬件开发平台上完成的，书中的部分硬件电路原理图也由其提供；感谢开放源码软件和开放文档的作者们；感谢本书的责任编辑薛媛，没有她的努力，本书是不可能与读者见面的；还要感谢以下人员的支持：杜培明、王培珍、程竹明、贺容波、赵江东等。

由于时间仓促，加之作者的编写水平有限，书中疏漏在所难免，恳请广大读者批评指正。

编 者

2011 年 8 月

目 录

第一篇 基础篇

第 1 章 嵌入式系统概述.....	3	2.6.4 Shell 中的 test 命令.....	37
1.1 嵌入式系统.....	3	2.6.5 条件语句.....	39
1.1.1 嵌入式系统的定义.....	3	2.6.6 循环语句.....	40
1.1.2 嵌入式系统组成.....	3	2.6.7 Shell 中的函数.....	44
1.1.3 嵌入式系统的特点.....	5	本章小结.....	45
1.1.4 嵌入式系统的应用.....	6		
1.2 嵌入式操作系统.....	7	第 3 章 Linux 编程环境.....	46
1.2.1 嵌入式操作系统分类.....	8	3.1 编辑器 vi.....	46
1.2.2 当前主流的嵌入式操作系统.....	9	3.1.1 vi 简介.....	46
1.3 嵌入式 Linux 操作系统.....	9	3.1.2 vi 用法.....	47
1.3.1 Linux 简介.....	9	3.1.3 vi 使用实例分析.....	48
1.3.2 嵌入式 Linux 操作系统.....	11	3.2 编译器 gcc.....	50
1.3.3 嵌入式 Linux 操作系统的优势.....	11	3.2.1 gcc 简介.....	50
1.4 嵌入式 Linux 开发环境.....	12	3.2.2 gcc 的执行过程.....	50
1.5 嵌入式 Linux 系统开发要点.....	13	3.2.3 gcc 的基本用法和选项.....	53
本章小结.....	14	3.2.4 gcc 编译实例.....	54
		3.3 调试器 GDB.....	55
第 2 章 初识 Linux.....	15	3.3.1 GDB 简介.....	55
2.1 安装发行版 Linux 操作系统.....	15	3.3.2 GDB 常用命令.....	55
2.1.1 安装虚拟机 VMware.....	15	3.3.3 GDB 使用实例.....	58
2.1.2 在虚拟机 VMware 中安装		3.4 make 工具和 Makefile 规则.....	62
RedHat Linux 9.0 操作系统.....	16	3.4.1 make 和 Makefile 概述.....	62
2.1.3 安装 VMware Tools.....	20	3.4.2 Makefile 文件内容.....	62
2.2 Linux 的目录结构.....	22	3.4.3 Makefile 规则.....	63
2.3 Linux 的文件属性.....	23	3.4.4 make 工作原理.....	64
2.4 Linux 的文件类型.....	23	3.4.5 Makefile 中的变量.....	64
2.5 Shell.....	25	3.4.6 clean 命令的应用.....	65
2.5.1 Shell 简介.....	25	3.4.7 Makefile 文件示例.....	65
2.5.2 常用 Shell 命令.....	26	3.5 glibc.....	67
2.6 Shell 编程基础.....	32	3.5.1 Linux 编程库简介.....	67
2.6.1 建立和运行 Shell 程序.....	32	3.5.2 Linux 线程库.....	68
2.6.2 Shell 中的变量.....	32	本章小结.....	70
2.6.3 Shell 中引号的使用方法.....	36		

第 4 章 交叉开发环境	71
4.1 交叉编译	71
4.2 交叉开发环境	71
4.3 交叉开发工具组成	72
4.4 宿主机与目标机之间的通信方式	74
4.4.1 互联通信方式	74
4.4.2 可拔插通信方式	75
4.5 交叉开发环境建立	75
4.5.1 创建交叉开发工具链	75
4.5.2 宿主机开发环境配置	75
4.6 基于 UP-NETARM2410-S 开发平台	
交叉开发环境建立	83
4.6.1 安装 PC 机 RedHat Linux 虚拟机 ...	83
4.6.2 开发工具软件的安装	83
4.6.3 宿主机网络配置	84
4.6.4 宿主机串口配置	85
4.6.5 基于 UP-NETARM2410-S 开发平台	
交叉开发环境建立的总结	85
4.7 交叉编译和交叉调试实例	86
本章小结	88

第 5 章 ARM 处理器	89
5.1 ARM 处理器概述	89

5.1.1 ARM	89
5.1.2 ARM 微处理器的应用领域及特点	90
5.1.3 ARM 微处理器版本及系列	90
5.2 ARM 微处理器结构	94
5.2.1 RISC 体系结构	94
5.2.2 ARM 微处理器的寄存器结构	95
5.2.3 ARM 微处理器的指令结构	95
5.3 ARM 编程模型	95
5.3.1 处理器的工作状态	95
5.3.2 流水线技术	96
5.3.3 存储器模式	98
5.3.4 数据和指令类型	98
5.3.5 处理器工作模式	98
5.3.6 寄存器组织	98
5.3.7 异常	104
5.4 ARM 指令集	108
5.4.1 ARM 处理器寻址方式	108
5.4.2 常见 ARM 指令集	110
5.5 ARM 汇编程序设计	124
5.5.1 ARM 汇编程序一些规范	124
5.5.2 ARM 汇编程序设计举例	125
本章小结	126

第二篇 提高篇

第 6 章 创建嵌入式 Linux 交叉开发工具链	129
6.1 下载源文件	129
6.2 建立工作目录	130
6.3 输出环境变量	131
6.4 建立内核头文件	132
6.5 建立二进制工具	134
6.6 建立初始编译器	136
6.7 建立 c 库	138
6.8 建立全套编译器	140
6.9 完成工具链的设置	141
6.10 测试和验证交叉编译工具	141
本章小结	142

第 7 章 嵌入式 Linux 引导程序	143
7.1 引导程序(Bootloader)概述	143
7.2 ARM Bootloader 的作用	143
7.3 Bootloader 通用的概念	144
7.4 Bootloader 的主要任务与框架结构	146
7.5 vivi	148
7.5.1 vivi 简介	148
7.5.2 vivi 的软件结构分析	150
7.5.3 vivi 启动原理	160
7.6 vivi 移植实例(移植 vivi 到博创 2410) ...	178
7.6.1 对 vivi 进行移植修改	179
7.6.2 对 vivi 进行配置与编译	181
7.6.3 烧写 vivi	183

7.6.4 测试 vivi.....	185
本章小结	190
第 8 章 嵌入式 Linux 内核	191
8.1 Linux 内核简介	191
8.2 Linux 内核软件结构分析	191
8.2.1 Linux 内核配置系统原理.....	191
8.2.2 配置 Linux 内核.....	203
8.2.3 Linux 内核启动原理.....	205
8.3 Linux 内核移植实例(移植 Linux2.6.14.1 到博创 2410)	207
8.3.1 下载 Linux 内核并解压.....	207
8.3.2 修改 Makefile.....	207
8.3.3 设置 Flash 分区	207
8.3.4 配置内核	211
8.3.5 测试 2.6.14.1 内核	215
8.3.6 内核编译过程中产生的 部分错误及其解决方法	218
本章小结	219
第 9 章 嵌入式 Linux 根文件系统	221
9.1 文件系统.....	221
9.1.1 文件系统简介	221
9.1.2 文件和目录	221
9.1.3 文件权限	223
9.1.4 文件链接	225
9.1.5 文件系统	226
9.2 Linux 文件系统	226
9.2.1 Linux 文件系统概述.....	226
9.2.2 Linux 文件系统组成.....	227
9.2.3 Linux 文件系统管理.....	227
9.2.4 虚拟文件系统(VFS).....	227
9.2.5 安装和卸载文件系统	228
9.3 嵌入式闪存文件系统.....	230
9.3.1 嵌入式系统存储设备及其 管理机制分析	230
9.3.2 基于 Flash 的文件系统	232
9.3.3 基于 RAM 的文件系统.....	233

9.3.4 网络文件系统 NFS(Network File System)	234
9.4 构建嵌入式 Linux 根文件系统原理	234
9.4.1 根文件系统关键目录	235
9.4.2 根文件系统关键文件	236
9.4.3 BusyBox	240
9.5 构建嵌入式 Linux 根文件系统实例—— CramFS 根文件系统构建(博创 2410).....	241
9.5.1 创建根文件系统目录	242
9.5.2 创建设备文件	242
9.5.3 编译 BusyBox	242
9.5.4 系统配置文件的建立	244
9.5.5 添加应用程序	245
9.5.6 创建 lib 库文件	245
9.5.7 cramfs 文件系统映像的制作	246
9.5.8 下载 cramfs 根文件系统映像到 开发板	246
9.5.9 测试 cramfs 根文件系统	246
9.5.10 cramfs 根文件系统制作中的 常见错误及解决办法	247
本章小结	250
第 10 章 嵌入式 Linux 设备驱动	251
10.1 Linux 设备驱动基本原理	251
10.1.1 概述	251
10.1.2 设备文件	252
10.1.3 使用/proc 目录中的文件监视 驱动程序的状态	254
10.2 Linux 设备驱动程序模块	254
10.3 Linux 设备驱动程序接口	256
10.4 Linux 设备驱动程序实现过程	258
10.4.1 驱动程序的注册与注销	258
10.4.2 设备的打开与释放	259
10.4.3 设备的控制操作(ioctl).....	260
10.4.4 中断处理	262
10.5 怎样编写设备驱动程序	263
10.6 Linux2.6 内核驱动原理	270
本章小结	274

第三篇 实 践 篇

第 11 章 TFT-LCD 驱动程序开发.....277

11.1 TFT-LCD 工作原理.....277	
11.1.1 LCD 概述.....277	
11.1.2 LCD 显示原理.....278	
11.1.3 LCD 驱动原理.....279	
11.1.4 LCD 的分类.....281	
11.2 LQ080V3DG01 液晶屏.....282	
11.3 SC2410 内部 LCD 控制器.....282	
11.3.1 管脚.....282	
11.3.2 寄存器介绍.....283	
11.3.3 控制流程.....284	
11.3.4 TFT 控制器操作.....285	
11.4 Linux 驱动程序接口	
——FrameBuffer.....285	
11.4.1 FrameBuffer.....285	
11.4.2 数据结构.....286	
11.5 LCD 驱动程序主要代码结构以及	
关键代码分析.....290	
11.5.1 FrameBuffer 驱动的统一管理.....290	
11.5.2 实现消息的分派.....291	
11.5.3 开发板 S3C2410 LCD 驱动的	
流程.....292	
11.6 LCD 驱动程序开发实例.....294	
11.6.1 S3C2410 与 LCD 模块的	
连接电路.....294	
11.6.2 修改代码(一)——LCD 显示	
企鹅图像.....295	
11.6.3 修改代码(二)——LCD 显示	
小车图像.....300	
本章小结.....301	

第 12 章 USB 驱动程序开发.....302

12.1 USB 简介.....302	
12.1.1 USB.....302	
12.1.2 USB 的结构及传输方式.....303	
12.1.3 USB 设备的使用.....303	
12.1.4 USB 的优势.....304	

12.2 Linux USB 子系统.....306	
12.2.1 文件系统.....306	
12.2.2 USB 驱动程序(USBD).....307	
12.2.3 主机控制器驱动程序(hcd).....307	
12.2.4 客户端驱动程序.....307	
12.3 USB 客户端驱动程序架构和开发.....307	
12.3.1 USB 的协议规范初识.....307	
12.3.2 Linux 的 USB 驱动框架.....309	
12.4 在 Linux 内核中加入 USB 模块.....317	
12.4.1 配置文件.....317	
12.4.2 调试.....318	
12.5 USB 驱动程序开发实例.....319	
12.5.1 S3C2410 与 USB 模块的	
连接电路.....319	
12.5.2 制作新的根文件系统.....319	
12.5.3 修改代码.....320	
本章小结.....326	

第 13 章 Qt 图形应用程序设计.....327

13.1 图形用户界面系统(GUI).....327	
13.2 Qt 和 Qt/Embedded 简介.....327	
13.3 Qt/Embedded 体系结构.....328	
13.3.1 交叉开发平台.....328	
13.3.2 Qt/E 的窗口系统.....329	
13.3.3 Qt/E 图形引擎的实现.....330	
13.4 Qt 编程入门.....331	
13.4.1 Qt 编程实例.....331	
13.4.2 代码解释.....338	
13.5 Qt 和 Qt/E 开发环境的建立和移植.....341	
13.5.1 Qt2 和 Qt2/E 开发环境的建立.....341	
13.5.2 Qt2/E 程序在博创 2410 开发板	
上的移植.....347	
13.5.3 Qtopia 开发环境的建立.....354	
本章小结.....356	

参考文献.....357

第一篇

基 础 篇

对于没有接触过 Linux 和嵌入式 Linux 编程环境的读者来说，嵌入式 Linux 下有很多的概念、名词及操作环境是陌生的，它们与 Windows 环境及 Windows 下集成开发环境在本质上虽然有类似之处，但表面上存在很多差异。

本篇讲述的主要内容：

- 第 1 章 嵌入式系统概述
- 第 2 章 初识 Linux
- 第 3 章 Linux 编程环境
- 第 4 章 交叉开发环境
- 第 5 章 ARM 处理器

作者从多年的嵌入式 Linux 的教学和开发经验中发现，对嵌入式 Linux 的学习关键在于入门，入门最好的途径是有兴趣学习嵌入式 Linux 系统，而培养兴趣最有效的途径是在实际操作，因此，读者一定要多上机练习，才能更好地掌握本篇的内容。



第1章 嵌入式系统概述

1.1 嵌入式系统

1.1.1 嵌入式系统的定义

目前,在嵌入式系统应用领域中,不少读者对什么是嵌入式系统不甚了解。在讲述嵌入式系统定义之前,先从现代计算机的发展历史去了解嵌入式系统的由来,从而才能够较准确理解嵌入式系统的定义。

电子数字计算机诞生于1946年,在其后漫长的历史进程中,计算机始终存在于特殊的机房中,是实现数值计算的大型昂贵设备。直到20世纪70年代微处理器出现,计算机才出现了历史性的变化。以微处理器为核心的微型计算机因其型小、价廉、可靠性高等特点,迅速走出机房。基于高速数值计算能力的微型机,表现出的智能化水平引起了控制专业人员的兴趣,他们要求将微型机嵌入到一个对象体系中,实现对对象体系的智能化控制。例如,将微型计算机经电气加固、机械加固,并配置各种外围接口电路,安装到大型舰船中,构成自动驾驶仪或轮机状态监测系统。这样一来,计算机便失去了原来的形态与通用的计算功能。为了区别于原有的通用计算机系统,就把嵌入到对象体系中,实现对对象体系智能化控制的计算机,称做嵌入式计算机系统。

对象系统是指嵌入式系统所嵌入的宿主系统。“嵌入性”、“专用性”和“计算机系统”是嵌入式系统的三个基本要素。

① 嵌入性:是指嵌入到对象系统中,满足对象系统的环境要求,如物理环境(小型)、电气环境(可靠)、成本(价廉)等要求。

② 专用性:嵌入式系统的软、硬件可裁剪,满足对象要求的最小软、硬件配置等。

③ 计算机系统:嵌入式系统必须是能满足对象系统控制要求的计算机系统,这样的计算机必须配置有与对象系统相适应的接口电路。

不要将嵌入式系统与嵌入式设备相混淆。嵌入式设备是指内部有嵌入式系统的产品、设备。例如,内含单片机的家用电器、仪器仪表、工控单元、机器人、手机以及PDA等。

1.1.2 嵌入式系统组成

本书所指的嵌入式系统是嵌入式系统的高端应用,其组成包括硬件和软件两大部分,其结构如图1-1所示。



图 1-1 嵌入式系统组成结构

1. 硬件层

硬件层由嵌入式处理器、存储器系统、通用设备接口和 I/O 接口(如 A/D、D/A、I/O 等)组成。在一片嵌入式微处理器基础上增加电源电路、时钟电路和存储器电路(ROM 和 SDRAM 等),就构成了七个嵌入式核心控制模块。其中,操作系统和应用程序都可以固化在 ROM 中。

2. 中间层

中间层也称为硬件抽象层或板级支持包,它把系统软件与底层硬件部分隔离。板级支持包对上具有操作系统相关性,对下具有硬件相关性。设计一个完整的中间层需要完成两部分工作:

- ① 嵌入式系统的初始化,它包括片级初始化、板级初始化和系统级初始化。
- ② 设计硬件相关的设备驱动。

3. 软件层

软件层由实时多任务操作系统(RTOS)、文件系统、图形用户接口(GUI)、网络系统及通用组件模块组成。RTOS 是嵌入式应用软件的基础和开发平台;是一段嵌入式目标代码中的程序,系统复位后首先执行,相当于用户的主程序,用户的其他应用程序都建立在 RTOS 之上;是一个标准的内核,它将 CPU 时钟、中断、I/O、定时器等资源都封装起来,留给用户的是一个标准的 API 函数接口。

4. 功能层与执行装置

功能层由基于 RTOS 开发的应用程序组成,用来完成对被控对象的控制功能。功能层是面向被控制对象和用户的,为方便用户操作,往往需要提供一个友好的人机界面。执行装置是指那些可以接受嵌入式计算机系统发出控制命令,执行所规定的操作或任务的设备和装置。在不同应用领域中,嵌入式系统的执行装置一般是不同的,应该根据具体的应用场合和系统所要求实现的功能选择不同的设备和执行装置。

为了更好地理解嵌入式系统的组成,将其与 PC 机的组成进行了比较,如表 1-1 所示。



表 1-1 嵌入式系统与 PC 机比较

设备名称	嵌入式系统	PC 机
CPU	嵌入式处理器(ARM、MIPS)	CPU(Intel 的 Pentium, AMD 的 Athlon 等)
内存	SDRAM 芯片	SDRAM、DDR 内存条
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
其它设备	USB 芯片、网卡芯片	主板集成或外接卡
引导代码	Bootloader 针对不同电路板进行移植	主板的 BIOS, 无须改动
操作系统	WinCE、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	操作系统含有大部分驱动程序或从网上下载直接使用
协议栈	需要移植	操作系统包括或第三方提供
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

1.1.3 嵌入式系统的特点

从上述嵌入式系统定义、组成及与 PC 机的比较可看出嵌入式系统的应用有如下一些特点:

1. 专用性强

由于嵌入式系统通常是面向某个特定应用的, 因此嵌入式系统的硬件和软件, 尤其是软件, 都是为特定用户群设计的, 它通常都具有某种专用性的特点。

2. 实时性好

目前, 嵌入式系统广泛应用于生产过程控制、数据采集、传输通信等场合, 因为主要用来对宿主对象进行控制, 所以都对嵌入式系统有或多或少的实时性要求。例如, 对嵌入在武器装备中的嵌入式系统、在火箭中的嵌入式系统以及一些工业控制装置中的控制系统, 其实时性要求就极高。也正因为这种要求, 在硬件上嵌入式系统极少使用存取速度慢的磁盘等存储器, 在软件上更是需要精心设计, 从而可使嵌入式系统快速地响应外部事件。随着嵌入式系统应用的扩展, 有些系统对实时性要求并不是很高, 例如近年来发展速度比较快的手持式计算机、掌上电脑等。但总体来说, 实时性是对嵌入式系统的普遍要求, 是设计者和用户重点考虑的一个重要指标。

3. 可裁剪性好

从嵌入式系统专用性的特点来看, 作为嵌入式系统的供应者, 理应提供各式各样的硬件和软件以供选用。但是这样做势必会提高产品的成本。为了既不提高成本, 又满足专用性的需要, 嵌入式系统的供应者必须采取相应措施使产品在通用和专用之间进行某种平衡。



目前的做法是,把嵌入式系统硬件和操作系统设计成可裁剪的,以便使嵌入式系统开发人员根据实际应用需要来量体裁衣,去除冗余,从而使系统在满足应用要求的前提下达到最精简的配置。

4. 可靠性高

由于有些嵌入式系统所承担的计算任务涉及产品质量、人身设备安全、国家机密等重大事务,加之有些嵌入式系统的宿主对象要工作在无人值守的场合,例如危险性高的工业环境中、内嵌有嵌入式系统的仪器仪表中、人际罕至的气象检测系统中以及为侦察敌方行动的小型智能装置中等。因此与普通系统相比较,对嵌入式系统可靠性的要求极高。

5. 功耗低

由于有很多嵌入式系统的宿主对象都是一些小型应用系统,例如移动电话、PDA、MP3、飞机、舰船、数码相机等,这些设备不可能配备容量较大的电源,因此低功耗一直是嵌入式系统追求的目标。当然,为了降低系统的功耗,嵌入式系统中的软件一般不存储在磁盘等载体中,而都固化在存储器芯片或单片系统的存储器之中。

1.1.4 嵌入式系统的应用

嵌入式技术是后 PC 时代技术的春天,应用领域非常广泛,主要有以下几方面。

1. 工业控制

基于嵌入式芯片的工业自动化设备将获得长足的发展,目前已经有大量的 8 位、16 位、32 位嵌入式微控制器在应用中,如工业过程控制、数字机床、电力系统、电网安全、电网设备监测和石油化工系统。就传统的工业控制产品而言,低端型采用的往往是 8 位单片机。但是随着技术的发展,32 位、64 位的处理器逐渐成为工业控制设备的核心,在未来几年内将获得长足的发展。

2. 交通管理

在车辆导航、流量控制、信息监测与汽车服务等方面,嵌入式系统技术已经获得了广泛的应用,内嵌 GPS 模块、GSM 模块的移动定位终端已经在各种运输行业中广泛使用。目前 GPS 设备已经从尖端产品进入了普通百姓的家庭,用户只需要几千元,就可以随时随地找到自己的位置。

3. 信息家电

信息家电将成为嵌入式系统最大的应用领域,冰箱、空调等的网络化、智能化将引领人们的生活步入一个崭新的空间。即使你不在家里,也可以通过电话线、网络对这些信息家电进行远程控制。在这些设备中,嵌入式系统将大有用武之地。

4. 家庭智能管理系统

水、电、煤气表的远程自动抄表,安全防火、防盗系统,其中嵌有的专用控制芯片将代替传统的人工检查,并实现更高、更准确和更安全的性能。目前在服务领域,如远程点菜器等已经体现了嵌入式系统的优势。

5. POS 网络及电子商务

公共交通无接触智能卡(Contactless Smart Card, CSC)发行系统、公共电话卡发行系统、



自动售货机……各种智能 ATM 终端将全面走入人们的生活，到时手持一卡就可以行遍天下。

6. 环境工程与自然

在很多环境恶劣、地况复杂的地区，嵌入式系统将实现无人监测，如水文资料实时监测、防洪体系及水土质量监测、堤坝安全监测、地震监测网、实时气象信息网以及水源和空气污染监测等。

7. 机器人

嵌入式芯片的发展将使机器人在微型化、高智能方面优势更加明显，同时会大幅度降低机器人的价格，使其在工业领域和服务领域获得更广泛的应用。

可见，学习和开发嵌入式系统有着十分重要的意义。

1.2 嵌入式操作系统

从上述嵌入式组成部分可以看出，嵌入式软件的核心是嵌入式实时多任务操作系统，即 RTOS(Real Time Operation System)。

在嵌入式系统开发中，为什么要采用嵌入式实时多任务操作系统呢？

这是因为，现代信息社会的高效性、协同性客观上要求软件的编写便于多人分工合作、编写的软件具有可重用性，软件日趋复杂等。所有这些迫切需要有一个屏蔽底层硬件的、功能强大的操作系统来支持。如下实例中数据采集终端要具有：采集、处理、键盘、LCD 显示、打印功能。在嵌入式系统开发中的传统编程如图 1-2 所示，而基于 RTOS 的编程如图 1-3 所示。

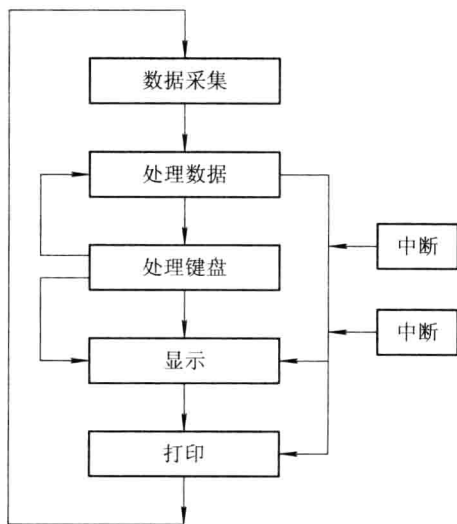


图 1-2 传统编程

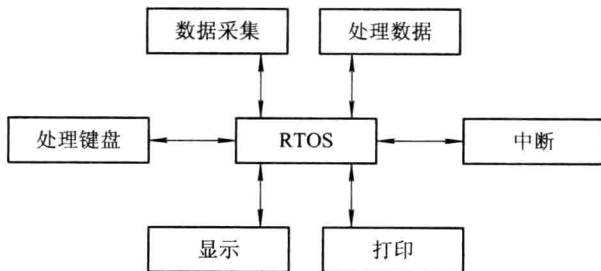


图 1-3 基于 RTOS 的编程

由图 1-2 和图 1-3 可见，传统的编程对于多任务的处理是基于顺序执行的，而基于 RTOS