



北京市高等教育精品教材立项项目

高等院校计算机精品教材系列

嵌入式系统原理 及应用开发



陈渝 主编
韩超 李明 编著



机械工业出版社
CHINA MACHINE PRESS

北京市高等教育精品教材立项项目
高等院校计算机精品教材系列

嵌入式系统原理及应用开发

- [4] 陈章龙, 庞志强, 陈时亮. 嵌入式系统原理及应用开发[M]. 北京: 北京航空航天大学出版社, 2005.
- [7] Intel Co. PXA255 数据手册. 陈渝 主编. Processor Developer's Manual, 2003.
- [8] Intel Co. PXA255 数据手册. 韩超 李明 编著. Processor Family Developer's Manual, 2004.
- [9] Andrew N Sloss. ARM 嵌入式系统开发——软件设计与优化[M]. 北京: 北京航空航天大学出版社, 2005.
- [10] 刘森. 嵌入式系统接口设计与Linux驱动程序开发[M]. 北京: 北京航空航天大学出版社, 2005.
- [11] 孙天洋, 袁文荣, 张海峰. 嵌入式系统原理及应用开发——基于ARM9[M]. 北京: 北京航空航天大学出版社, 2005.
- [12] Linux kernel 文档. <http://www.kernel.org/>. 2008. 5. 林强出
- [13] 北京航空航天大学. 嵌入式系统原理及应用开发[M]. 北京: 北京航空航天大学出版社, 2005.
- [14] 赵朝. Linux 内核完全剖析[M]. 北京: 机械工业出版社, 2005.
- [15] SkyEye 网站. <http://www.skyeye.com/>. ISBN 978-7-111-23424-8. 2005.
- [16] 魏永明. 嵌入式系统原理及应用开发[M]. 北京: 中国电力出版社, 2002.
- [17] 嵌入式系统原理及应用开发[M]. 北京: 机械工业出版社, 2005.

机械工业出版社

嵌入式系统是一个涉及多方面知识的交叉学科，目前 ARM 和 Linux 是嵌入式系统发展中的两个重点方向。本书是关于嵌入式系统的基础教程，立足于基于 ARM 体系结构的 Linux 开发。全书共 12 章，包含了嵌入式系统发展方向、ARM 体系结构、嵌入式处理器、嵌入式硬件系统、嵌入式软件结构、基于 GNU Tools 的程序开发、Linux 内核移植、Linux 驱动程序、Linux 应用开发等方面的内容，组成了一个从硬件到软件的完整的知识结构。

本书可作为高等院校计算机、自动控制、电子等相关专业本科生及研究生教材，也适用于工程技术人员参考。

图书在版编目 (CIP) 数据

嵌入式系统原理及应用开发/陈渝主编. —北京: 机械工业出版社, 2008. 2

北京市高等教育精品教材立项项目. 高等院校计算机精品教材系列

ISBN 978 - 7 - 111 - 23424 - 1

I. 嵌... II. 陈... III. 微型计算机 - 系统开发 - 高等学校 - 教材 IV. TP360. 21

中国版本图书馆 CIP 数据核字 (2008) 第 016953 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 赵 慧 责任编辑: 唐德凯

责任校对: 张莉娟 责任印制: 邓 博

北京京丰印刷厂印刷

2008 年 4 月第 1 版 · 第 1 次印刷

184mm × 260mm · 23.5 印张 · 582 千字

0 001—5 000 册

标准书号: ISBN 978 - 7 - 111 - 23424 - 1

定价: 38.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

销售服务热线电话: (010) 68326294

购书热线电话: (010) 88379639 88379641 88379643

编辑热线电话: (010) 88379739

封面无防伪标均为盗版

前 言

本书的特点

嵌入式系统是 21 世纪具有广阔发展前景的技术方向之一。随着微电子技术、网络技术等信息技术的飞速发展，对嵌入式应用的设计要求越来越复杂，已有的基于 8/16 位单片机的嵌入式应用已经逐渐让位给基于 32 位嵌入式微处理器的嵌入式应用。由于在种类繁多的 32 位嵌入式 CPU 中，基于 ARM CPU 内核的嵌入式微处理器所占的比例最高，而且它基于 RISC 架构，利于学习和掌握。因此，本书选择基于 ARM CPU 内核的嵌入式系统作为本书的参考硬件平台。针对上述发展现状，通过对嵌入式系统开发的实践过程，总结出嵌入式系统开发需要掌握的知识和经验，编写了这本书。与以往单纯介绍嵌入式系统知识的书籍相比，本书更注重理论性和实践性相结合，具有如下特点：

第一，本书对 ARM 为核心的 CPU、外设和开发板进行了全面深入的讲解，从而可以真正从实践中了解和掌握嵌入式系统的硬件开发知识。

第二，本书选择 Linux 作为我们研究的嵌入式操作系统。Linux 是功能强大的操作系统，横跨了服务器、台式机和嵌入式系统三个硬件领域。由于开放源代码，读者可以深入地分析和了解它们的实现细节，从而可以更深入地了解嵌入式操作系统和系统软件的具体开发和实现过程。

第三，如果要开发面向嵌入式领域的系统级软件或操作系统，选择基于 GPL 协议和 Open Source 协议的开发工具有很多优势。目前开源工具（GNU Tools）支持的嵌入式 CPU 种类是最多的，远远超过其他的嵌入式开发工具，而且目前还有多种高质量的集成开发环境（IDE）的支持，这样开发人员除了直接使用命令行工具还有更多的选择空间。因此，本书选择 GNU Tools 等高质量软件作为嵌入式开发工具。本书介绍的操作系统和系统软件实例都是用 GNU Tools 开发的。

第四，本书配有实验指导书和光盘，光盘上提供嵌入式软件开发工具和硬件模拟器、嵌入式操作系统和本书作者实现的软件实例。读者在阅读本书的时候，可以配合实验指导书和光盘提供的软件来进行实践。

内容的组织

本书由 12 章构成。第 1 章为绪论，介绍了嵌入式系统的一些基本概念，包括嵌入式系统的起源、发展过程、分类以及当前嵌入式系统的几个组成部分等内容；第 2 章到第 5 章为硬件部分，主要介绍了嵌入式系统硬件的组成原理、ARM 体系结构、ARM 处理器、基于 ARM 处理器的嵌入式开发板等内容；第 6 章到第 12 章介绍了在嵌入式系统开发过程中需要使用的各种开发工具和开发手段，涉及操作系统移植、驱动开发、应用程序开发等。具体的

内容如下：

第1章，介绍了嵌入式系统的一些基本概念和基本知识；

第2章，介绍了嵌入式硬件系统的组成原理、最小系统组成、外设等；

第3章，介绍了 ARM 体系结构、ARM 系列内核分类、ARM 指令集和 ARM 的 MMU 设计等部分；

第4章，介绍了 ARM 嵌入式处理器的基本概念与类型，并以两种 ARM 处理器：S3C44B0X 和 PXA 2xx 为代表，介绍 ARM 处理器的结构和功能部件；

第5章，介绍了嵌入式系统扩展的逻辑组成和基于 S3C44B0X 处理器的嵌入式系统以及基于 PXA2xx 处理器的嵌入式系统；

第6章，讲述了嵌入式软件开发的基础知识，包括嵌入式的开发语言、程序开发流程、嵌入式调试工具以及 C 语言程序结构等几方面的内容；

第7章，讲述了在 Linux 下的 GNU Tools 的使用，特别是 ld、gcc、gdb、make 的使用细节；

第8章，介绍了嵌入式 Linux 构建的原理和流程，包括 BootLoader 的实现原理、Linux 内核的编译过程、根文件系统和具体文件系统的描述等内容；

第9章，讲述了 Linux 内核的移植原理，介绍了标准 Linux 2.6 在具有 MMU 的 ARM 处理器上的移植和 μ Clinux 2.4 在无 MMU 的 ARM 处理器上的移植；

第10章，介绍了 Linux 的内核驱动程序的基础知识，不涉及具体硬件的驱动程序，主要包含内核模块的功能、字符设备、块设备和网络设备的驱动程序框架等内容；

第11章，介绍了 Linux 的驱动程序开发中主要的 3 种驱动程序：串口驱动、帧缓冲驱动和以太网驱动；

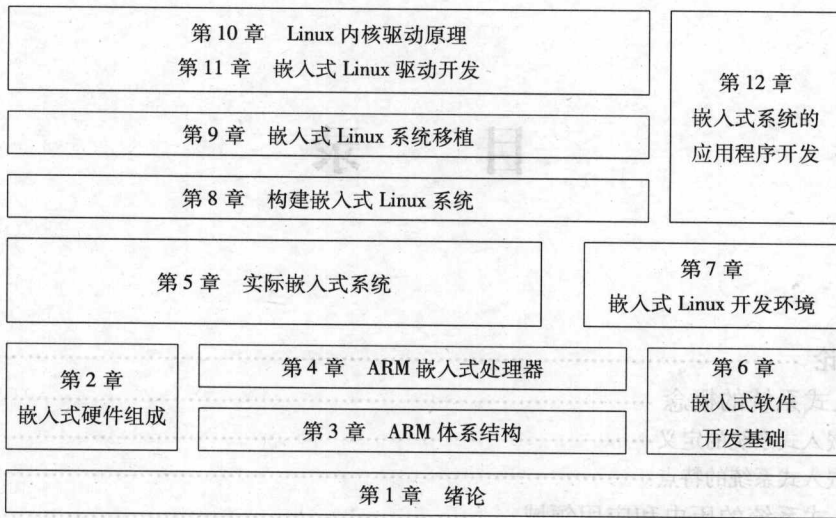
第12章，介绍了嵌入式系统的应用程序开发，主要包括嵌入式 GUI 系统和网络应用两个方面。嵌入式 GUI 的内容涉及嵌入式 GUI 系统、GUI 系统下层的移植和 GUI 系统上层的应用程序开发；网络应用的内容主要包含 Linux 应用层面中的网络编程。

阅读建议

为了更好地阅读本书，我们希望读者熟悉 C 语言，对计算机体系结构、ARM 体系结构和汇编指令有一定的了解，能够掌握 Linux 的基本使用方法。由于本书中部分内容与硬件相关，我们在提到特定硬件时，给出了必要的信息。

本书的大多数实验都是在 Linux 上运行的，所以熟悉使用 Linux 是能够实践本书讲解内容的基础。如果您是嵌入式系统的初学者，建议读者一边阅读本书，一边参照配套的实验教材中的内容在 Linux 系统上进行各种实践。如果没有真实的开发板硬件，则可以用 ARM 嵌入式硬件模拟器在 PC 上进行实践。

本书的各个章节有一定的联系和相关性，其关系图如下所示：



各章之间的关系图

本书的组织和编写工作主要由陈渝、韩超和李明完成，韩超和陈渝完成全书的统稿。此外，参与本书编写的还有谢劲松、王月龙、刘永安、吴德新、冯学青、邓祺和李鹏飞。北京亚嵌教育研究中心的老师和学员在教学实践过程中也对本书的部分章节提出了很好的建议，在此一并表示感谢。

虽然作者对本书进行了认真地检查，但书中肯定还有不足之处，恳请读者批评指正。

编者

目 录

前言

第1章 绪论	1
1.1 嵌入式系统的概念	1
1.1.1 嵌入式系统的定义	1
1.1.2 嵌入式系统的特点	1
1.2 嵌入式系统的历史和应用领域	3
1.3 嵌入式系统的组成	5
1.3.1 微处理器	5
1.3.2 外围硬件	6
1.3.3 嵌入式操作系统	6
1.3.4 应用程序	6
1.3.5 开发环境	6
习题	7
第2章 嵌入式硬件组成	8
2.1 嵌入式系统组成原理	8
2.1.1 计算机系统的一般运行原理	8
2.1.2 嵌入式系统与通用计算机硬件组成的异同	11
2.1.3 嵌入式系统硬件结构	12
2.2 嵌入式最小系统	14
2.2.1 嵌入式最小系统的概念	14
2.2.2 实际的嵌入式最小系统	17
2.3 嵌入式系统扩展	18
2.3.1 内存类芯片	19
2.3.2 网络芯片	21
2.3.3 USB 芯片	21
习题	22
第3章 ARM 体系结构	23
3.1 ARM 微处理器基础	23
3.1.1 ARM 的运行模式	24
3.1.2 ARM 的工作状态	25
3.1.3 ARM 的寄存器组织	26
3.1.4 ARM 的异常	32
3.1.5 ARM 的存储器格式	36
3.1.6 ARM 的特点和优势	37

3.2	ARM 的内核系列	39
3.2.1	ARM 内核分类和系列	40
3.2.2	ARM 体系结构版本	50
3.3	ARM 微处理器的指令系统	54
3.3.1	ARM 指令系统概述	54
3.3.2	ARM 的寻址模式	56
3.3.3	ARM 的指令集	59
3.3.4	Thumb 的指令集	65
3.4	ARM 内存管理单元	67
3.4.1	ARM MMU 概述	67
3.4.2	ARM MMU 功能与操作	68
3.4.3	ARM MMU 的地址转换	72
	习题	79
第4章	ARM 嵌入式处理器	80
4.1	ARM 微处理器概述	80
4.1.1	ARM 核心微处理器的分类和特点	80
4.1.2	ARM 核心微处理器的使用	88
4.2	S3C44BOX 处理器	90
4.2.1	概述	90
4.2.2	处理器组成与框图	91
4.2.3	ARM7TDMI 核心	94
4.2.4	S3C44BOX 重点部件特点	95
4.3	PXA25x/PXA27x 处理器	114
4.3.1	概述	114
4.3.2	处理器组成与框图	116
4.3.3	XScale 核心	119
4.3.4	重点部件特点	120
	习题	131
第5章	实际嵌入式系统	132
5.1	嵌入式系统的硬件扩展	132
5.1.1	内部模块扩展	133
5.1.2	总线扩展	133
5.1.3	GPIO 扩展	133
5.1.4	综合扩展	134
5.2	S3C44BOX 系统	135
5.2.1	系统概述	135
5.2.2	系统组成	137
5.2.3	系统总结	151
5.3	PXA 2xx 系统	152
5.3.1	系统概述	152
5.3.2	系统组成	153

习题	154
第 6 章 嵌入式软件开发基础	156
6.1 嵌入式软件开发语言	156
6.2 嵌入式软件开发流程	157
6.2.1 嵌入式应用程序生成的特点	157
6.2.2 编译	158
6.2.3 汇编	158
6.2.4 链接	158
6.2.5 加载程序	159
6.3 调试工具和方法	159
6.3.1 嵌入式调试的特点	159
6.3.2 嵌入式开发的常用调试方式	160
6.4 C 语言程序的结构	161
6.4.1 C 语言目标文件中的段	161
6.4.2 执行程序内部细节	162
6.4.3 目标文件各段的链接	165
6.4.4 C 语言程序的执行	166
6.5 仿真环境	168
习题	169
第 7 章 嵌入式 Linux 开发环境	170
7.1 GNU Tools	170
7.1.1 GNU Tools 综述	170
7.1.2 GCC 编译与链接	173
7.1.3 GCC 工具介绍	178
7.1.4 C 语言库	185
7.1.5 ELF 文件格式	186
7.2 项目管理	188
7.2.1 工程管理工具 Make 与 Makefile	188
7.2.2 版本控制	192
7.3 调试器	193
7.3.1 GDB 调试基础	193
7.3.2 远程 GDB 调试	197
7.4 模拟器	198
习题	198
第 8 章 构建嵌入式 Linux 系统	200
8.1 嵌入式 Linux 系统的构建流程	200
8.1.1 嵌入式 Linux 的组成	200
8.1.2 嵌入式 Linux 的构建	201
8.2 Bootloader	202
8.2.1 嵌入式 Linux 的引导方式	202
8.2.2 Bootloader 的作用	203

8.3	Linux 内核	203
8.3.1	Linux 内核配置和编译的流程	203
8.3.2	Linux 内核配置	204
8.3.3	Linux 内核的编译	205
8.4	根文件系统	206
8.4.1	Linux 下的文件系统	206
8.4.2	嵌入式 Linux 的文件系统介绍	209
8.4.3	Linux 的根文件系统	215
	习题	216
第 9 章	嵌入式 Linux 系统移植	217
9.1	Bootloader	217
9.1.1	Linux Bootloader 概述	217
9.1.2	Bootloader 的 stage1	220
9.1.3	Bootloader 的 stage2	221
9.2	基于 ARM-with MMU 的 Linux 移植	224
9.2.1	Linux 内核概述	224
9.2.2	标准 Linux 移植概述	226
9.2.3	基于 PXA 2xx 嵌入式系统的 ARM-Linux 移植	235
9.3	基于 ARM-noMMU 的 μ CLinux 移植	248
9.3.1	μ CLinux 的特点	248
9.3.2	μ CLinux 移植概述	251
9.3.3	S3C44B0X 系统的 μ CLinux 的移植	256
	习题	265
第 10 章	Linux 内核驱动原理	266
10.1	Linux 驱动程序概述	266
10.2	Linux 驱动程序的相关知识	268
10.2.1	内核模块的作用	268
10.2.2	模块的编程结构和使用	269
10.2.3	内核模块编写的注意事项	270
10.2.4	Linux 设备文件	272
10.2.5	Linux 对设备文件的操作	273
10.2.6	设备文件相关的数据结构	277
10.2.7	内核空间和用户空间的交互	279
10.3	驱动程序的结构	281
10.3.1	字符驱动程序框架	281
10.3.2	驱动程序的操作	285
10.3.3	带中断的驱动程序	286
10.3.4	块设备的驱动程序框架	288
10.3.5	网络设备的驱动程序框架	291
	习题	293
第 11 章	嵌入式 Linux 驱动开发	294

11.1	串口及控制台驱动	294
11.1.1	串口及控制台驱动的作用	294
11.1.2	串口驱动程序	295
11.1.3	控制台功能的实现	302
11.2	Framebuffer 驱动	305
11.2.1	LCD 驱动原理	306
11.2.2	Framebuffer 驱动结构	311
11.2.3	虚拟显示缓冲驱动分析	316
11.3	以太网驱动	324
11.3.1	Linux 以太网的应用	324
11.3.2	Linux 网络驱动	327
11.3.3	回环网络设备驱动程序分析	333
11.4	Linux 驱动程序总结	335
	习题	336
第 12 章 嵌入式系统的应用程序开发		337
12.1	嵌入式 GUI 系统	337
12.1.1	嵌入式 GUI 的概述	337
12.1.2	GUI 系统的设计原理	338
12.1.3	GUI 系统的移植	342
12.1.4	GUI 系统的应用开发	344
12.2	嵌入式网络应用	345
12.2.1	网络基础知识	345
12.2.2	TCP/IP 知识	346
12.2.3	嵌入式 Linux 的网络应用	349
	习题	356
附录		357
附录 A	ARM 指令集	357
附录 B	嵌入式系统网络资源	360
附录 C	常用逻辑符号对照表	363
参考文献		365

第 1 章 绪 论

本章介绍了嵌入式系统一些基本概念，包括嵌入式系统的起源、发展过程、分类以及当前嵌入式系统的几个组成部分。

通过本章的学习，将掌握并理解如下一些重要问题：

- ◇ 嵌入式系统的概念和特点
- ◇ 电子和计算机两大领域在嵌入式系统的结合
- ◇ 嵌入式系统的组成

1.1 嵌入式系统的概念

1.1.1 嵌入式系统的定义

IEEE（电气和电子工程师协会）对嵌入式系统定义为：嵌入式系统是“控制、监视或者辅助装置、机器和设备运行的装置”（devices used to control, monitor, or assist the operation of equipment, machinery or plants）。这种定义来自应用的角度，从中可以看出嵌入式系统是软件和硬件的综合体，还可以涵盖机械辅助设备附属装置。

一个普遍被认同的“嵌入式系统”的定义是：以应用为中心、以计算机技术为基础、软件硬件可裁剪、适应应用系统对功能、可靠性、成本、体积、功耗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、嵌入式操作系统以及用户的应用程序等四个部分组成，用于实现对其他设备的控制、监视或管理等功能。

1.1.2 嵌入式系统的特点

从本质上看，嵌入式系统属于计算机系统的一种具体体现。但是它与通用的计算机系统存在差异。相比通用的计算机系统，嵌入式系统主要具有以下一些特点：

（1）特定的设计

嵌入式系统通常是面向特定应用的，嵌入式处理器与通用计算机的处理器最大不同之处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。它通常都具有低功耗、体积小、集成度高等特点，能够把通用计算机 CPU 和许多由板卡完成的任务集成在处理器芯片内部，从而有利于嵌入式系统设计趋于小型化，移动能力大大增强，与网络的结合也越来越紧密。

（2）各学科结合的产物

嵌入式系统是先进的计算机技术、半导体技术和电子技术与各个行业的具体应用相结合的产物，这就决定了它必然是一个技术密集、资金密集、高度分散、不断创新的知识集成系统。

（3）高效的设计

嵌入式系统的硬件和软件都必须高效率地设计，量体裁衣、去除冗余，力争在同样的硅片面积上实现更高的性能，这样的处理器与同类产品相比才更具有竞争力。

(4) 与具体应用相结合

嵌入式系统和具体应用有机地结合在一起，它的升级换代也是和具体产品同步进行的，因此嵌入式系统产品一旦进入市场，具有较长的生命周期。

(5) 存储特性

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自身内，而不是存储于磁盘等载体中。

(6) 交叉开发

嵌入式系统本身不具备自举开发能力，即使设计完成以后用户通常也不能对其中的程序功能进行修改，必须有一套开发工具和环境才能进行开发。

嵌入式系统的交叉开发主要是指嵌入式系统的软件开发方式，这是嵌入式系统区别于通用计算机软件开发的最主要特性。实质上，任何系统的软件开发都包括两个最基本的部分：程序的生成和程序的载入。

对于通用计算机的软件开发，这两个步骤相对比较简单：

程序的生成：用一定的编译工具将各种计算机语言转换成计算机可以执行的二进制代码，一般包括编译和链接等步骤，常用的工具包括 Windows 下的 Visual C++ 或 Linux 下的 gcc 等。

程序的载入：程序的载入是指将程序放入需要执行的地方。一般来说，通用计算机编译工具生成的可执行代码就在开发的计算机上，如果在本机使用可以直接运行。如果需要放到其他计算机运行，可以使用网络、移动存储设备直接复制，相对复杂的情况也不过是将生成的二进制代码封装成安装程序。实质上，通用计算机的程序装入的本质，就是将程序复制到执行的计算机，复制程序的过程可以使用通用计算机的任何工具。这是因为开发程序和运行程序的计算机虽然可能在配置和使用方式上存在差异（例如：一个是工作站，一个是个人电脑），但它们本质是一样的（具有相同的体系结构，程序基本兼容）。图 1-1 说明了通用计算机程序生成和载入的过程。

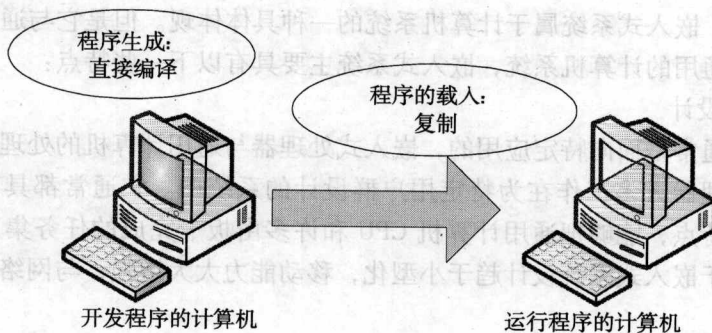


图 1-1 计算机系统程序开发

对于嵌入式系统的软件的开发，这两个步骤要相对复杂。

程序的生成：嵌入式系统程序的生成一般使用交叉编译的模式（主机-目标机模式），主

机一般使用通用计算机,这时就需要使用交叉编译工具,即在主机上完成程序的编写后,使用主机开发工具(编译链接工具)将程序转换成目标机可执行的二进制代码。这种二进制代码是不能在主机运行的。也就是说,在开发过程中,开发工具是在主机运行的程序,生成的程序是在目标机运行的程序。

程序的载入:由于在嵌入式系统的软件开发中,程序生成的地方和程序运行的地方不一致,因此程序的载入比较复杂。程序的载入方式在很大程度上取决于目标机的状态。在程序开发的初级阶段,目标机只是一些电子元器件的组合,程序的载入使用简单的烧写方式,将二进制代码烧写到目标机的固化存储器(Flash)中,由目标机上电后自动运行;在目标机具有一定的软件后,目标机就能形成一个可以运作的系统,然后使用各种通信方式将主机的目标代码复制到目标机。当然,能使用何种方式取决于目标机具有何种通信手段。图1-2说明了嵌入式系统程序生成和载入的过程。

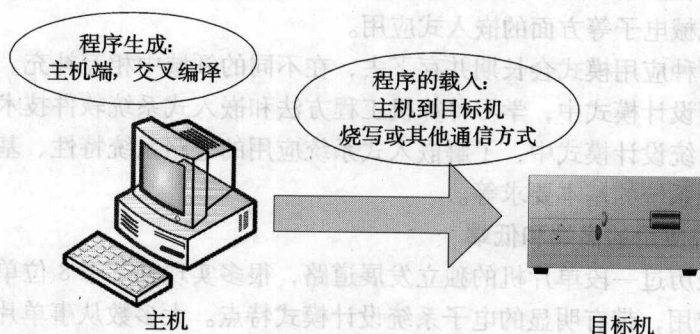


图1-2 嵌入式系统程序开发

1.2 嵌入式系统的历史和应用领域

嵌入式系统是一个交叉学科,它的起因、发展都和电子专业与计算机专业有着很强的相关性。根据嵌入式系统的概念,它是一种计算机系统,但是它又和传统的计算机系统有着很大的差异。嵌入式系统的物质基础一般都来自一些电子元器件。嵌入式系统工程师的一个主要任务就是将这些分离的电子元器件构件成一个可以实现一定逻辑功能的计算机系统。在嵌入式系统发展的历史过程中,由于嵌入式应用具有多样性的特点,在客观上形成的两种相对独立的应用模式。

1. 单片机模式和计算机模式

嵌入式系统起源于微型计算机时代,但很快就进入独立发展的单片机时代。在单片机时代,嵌入式系统以器件形态迅速进入到传统电子技术领域中,以电子技术应用工程师为主体,实现传统电子系统的智能化,而计算机专业的工程师并没有真正进入单片机应用领域。

电子技术应用工程师习惯于电子技术的设计和应用模式,以这种方式从事单片机的开发。这种应用模式最重要的特点是:软、硬件的实现具有底层性和随意性;对象系统专业技术具有密切相关性;缺少计算机工程设计方法。

在嵌入式系统的单片机时代，计算机专业在嵌入式系统领域起的作用不大。随着后 PC 时代的到来，嵌入式系统软、硬件技术有了很大的提升，这为计算机专业的人士从事嵌入式系统的开发工作提供了很好的条件，例如：很多计算机的软件和操作系统的设计方法可以应用到嵌入式系统中。随着计算机专业人士进入嵌入式领域，形成的计算机应用模式，带有明显的计算机的工程应用特点，即基于嵌入式系统软、硬件平台，以网络应用、多媒体、商务电子等为主的非嵌入式底层应用。

2. 两种应用模式的并存与互补

嵌入式系统最大、最广、最底层的应用是传统电子技术领域的智能化改造。因此，以熟悉具体对象专业的电子技术队伍为主，用最少的嵌入式系统软、硬件开销，以 8 位机为主，带有浓厚的电子系统设计色彩的电子系统应用模式会长期存在下去。

计算机专业人士会愈来愈多地介入嵌入式系统应用，但由于具体对象专业知识的隔阂，其应用领域会集中在网络应用、多媒体、商务电子等方面，不可能替代原来电子工程师在控制、仪器仪表、机械电子等方面的嵌入式应用。

客观存在的两种应用模式会长期并存下去，在不同的领域中相互补充。电子系统设计模式应从计算机应用设计模式中，学习计算机工程方法和嵌入式系统软件技术；计算机应用设计模式应从电子系统设计模式中，了解嵌入式系统应用的电路系统特性、基本的外围电路设计方法和实际对象系统的基本要求等。

3. 嵌入式系统应用的高端和低端

嵌入式系统经历过一段单片机的独立发展道路，很多实现都基于 8 位单片机，实现最底层的嵌入式系统应用，带有明显的电子系统设计模式特点。大多数从事单片机应用的开发人员，都是具体对象系统领域中的电子系统工程师，随着单片机的出现，嵌入式系统立即脱离了计算机专业领域，并以“智能化”器件身份进入电子系统领域，也没有引入“嵌入式系统”概念。因此，对于很多从事单片机应用的人，并不了解单片机与嵌入式系统的关系，在谈到“嵌入式系统”时，往往理解成计算机专业领域的，基于 32 位嵌入式处理器，实现网络应用、多媒体、商务电子等功能。

“单片机”是典型的、独立发展起来的嵌入式系统，从学科建设的角度出发，应该把它统一成“嵌入式系统”。考虑到原来单片机的电子系统底层应用特点，可以把嵌入式系统应用分成高端与低端，把原来的单片机应用理解成嵌入式系统的低端应用，主要特性表现为它的底层性以及对象系统的紧耦合。图 1-3 说明了计算机学科和电子学科在嵌入式系统中的融合。

在嵌入式系统发展的初期，工程师一般都会独自完成系统的软硬件设计，包括从电子元器件的选择、逻辑电路的设计、PCB 板的设计直到软件的编写。随着嵌入式系统的发展，其本身的功能越来越强大，用户的需求也越来越高，很难依靠个人力量完成整个系统的开发。

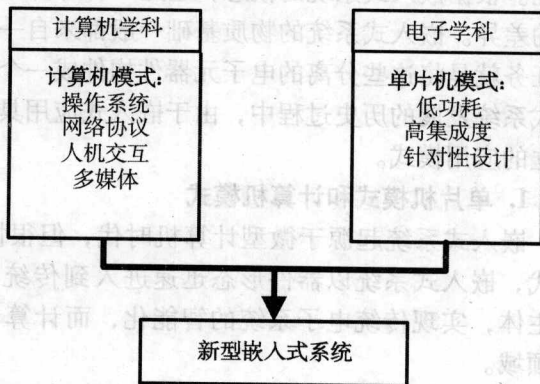


图 1-3 计算机模式和单片机模式在嵌入式系统的融合

因此，在当今的嵌入式设计中，也产生了阶段性的开发方式和多级产业链。电子专业和计算机专业在嵌入式系统中产生了融合，它们可以分别完成嵌入式系统的下层和上层设计。例如，由电子工程师完成嵌入式系统从处理器到外围电路的设计和实现，由计算机工程师构建程序开发环境、实现或移植操作系统以及完成具体的应用软件开发。此外，由于嵌入式系统与客观的对象有着很强的交互性，它与工业控制、通信的等学科的联系也非常紧密。

1.3 嵌入式系统的组成

嵌入式系统一般指非桌面系统和服务器系统的计算机系统。根据计算机系统的概念，它可以分成硬件部分和软件部分。硬件部分包括嵌入式处理器（MPU）或控制器（MCU）以及外围电路；软件部分包括嵌入式操作系统（Embedded Operating System）和应用程序。综合物理和逻辑上的模块划分，以及开发中的步骤，嵌入式系统可以划分成以下几个部分：

- 微处理器
- 外围硬件
- 嵌入式操作系统
- 应用程序
- 开发环境

图 1-4 说明了嵌入式系统各个组成部分以及它们之间的关系。

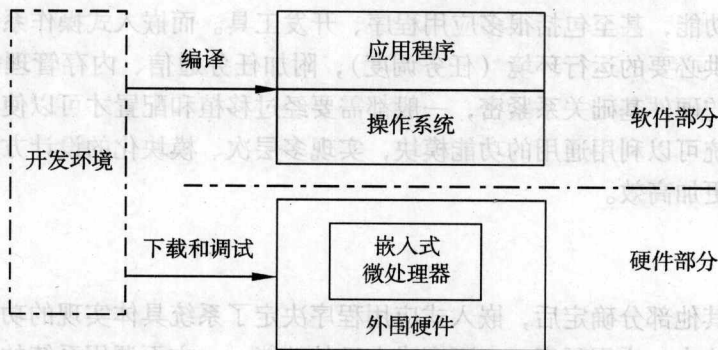


图 1-4 嵌入式系统的组成

1.3.1 微处理器

微处理器是嵌入式系统的核心。嵌入式微处理器一般都具有较高的集成度。PC 的处理器一般仅具有 CPU（中央处理单元，包含控制器和运算器）的功能，而大多数嵌入式微处理器不仅包括 CPU 核心，也包括丰富的片内部件。从功能结构上，嵌入式微处理器同时集成了 PC 机 CPU 和主板-芯片组的功能，例如：嵌入式处理器中大多包含内存管理器、中断控制器、定时器，而在 PC 系统中，芯片组的北桥包含了内存管理单元，南桥包含中断控

制、定时器等单元。

嵌入式处理器都包含 CPU 核心与片内部件两个部分。从基本原理上看，嵌入式微处理器的 CPU 核心决定了系统需要生成哪种格式的二进制代码才可以执行，而嵌入式系统的片内外设则决定系统有哪些基本资源可以使用。

1.3.2 外围硬件

外围硬件是嵌入式系统处理器以外的硬件，它为嵌入式处理器提供了运行条件并增加部分功能。在运行条件方面，最基础的是时钟和电源部件，它们不可能集成到处理器内部，但它们对于处理器运转是必需的。虽然嵌入式处理器的集成度一般都很高，但也不能把所有的功能都集成到处理器内部，因此外围硬件也提供了在系统中需要、但微处理器内部不具有的功能模块，例如：内存、各种通信端口、AD/DA 等。

在通用计算机系统（例如 PC）中，处理器之外的外围硬件是标准化的，与 CPU 的接口也是一致的。而嵌入式系统的外围硬件则是很灵活的，而且在很大程度上取决于核心微处理器的结构。同时，嵌入式系统的外围硬件也会影响到系统程序的编写。

嵌入式系统微处理器和外围硬件构成了系统的硬件基础。

1.3.3 嵌入式操作系统

在嵌入式系统中，操作系统并不是一个必要部件。很多嵌入式系统可以在没有操作系统的情况下运行。嵌入式操作系统和嵌入式应用程序关系紧密，在很多系统中，嵌入式操作系统和应用程序是被编译-链接在一起的，它们之间的差别仅仅是逻辑上的。

在通用计算机上，普遍使用的操作系统有 Windows、Linux 等，它们提供了从内核到系统级应用的完整管理功能，甚至包括很多应用程序、开发工具。而嵌入式操作系统的作用一般只是为应用程序提供必要的运行环境（任务调度），附加任务通信、内存管理等功能。嵌入式操作系统和系统的硬件基础关系紧密，一般都需要经过移植和配置才可以使用。嵌入式操作系统使嵌入式系统可以利用通用的功能模块，实现多层次、模块化的设计方式，这可以让嵌入式系统的开发更加高效。

1.3.4 应用程序

当嵌入式系统的其他部分确定后，嵌入式应用程序决定了系统具体实现的功能。

在通用计算机系统中，应用程序一方面完成自己的功能，一方面调用系统的功能。系统调用一般都是由操作系统提供的。

在嵌入式系统中，应用程序的层级远远少于通用计算机。由于控制硬件是嵌入式系统基本的操作，因此嵌入式应用程序依然与系统的硬件基础关系密切。尤其在没有操作系统的情况下，嵌入式的应用程序需要直接访问寄存器或者设备的地址来操作硬件。

1.3.5 开发环境

开发环境并不是嵌入式系统产品的一部分，但是它在嵌入式系统产品的开发过程中起着至关重要的作用。不同嵌入式系统的开发环境差异是很大的。嵌入式系统的开发环境包括几个环节：嵌入式的交叉编译环境、主机到目标机的程序载入环境、主机到目标机的调试环境