

EMBEDDED
SYSTEM

嵌入式技术与应用丛书

基于ARM的 嵌入式系统开发

刘 岚 尹 勇 李京蔚 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

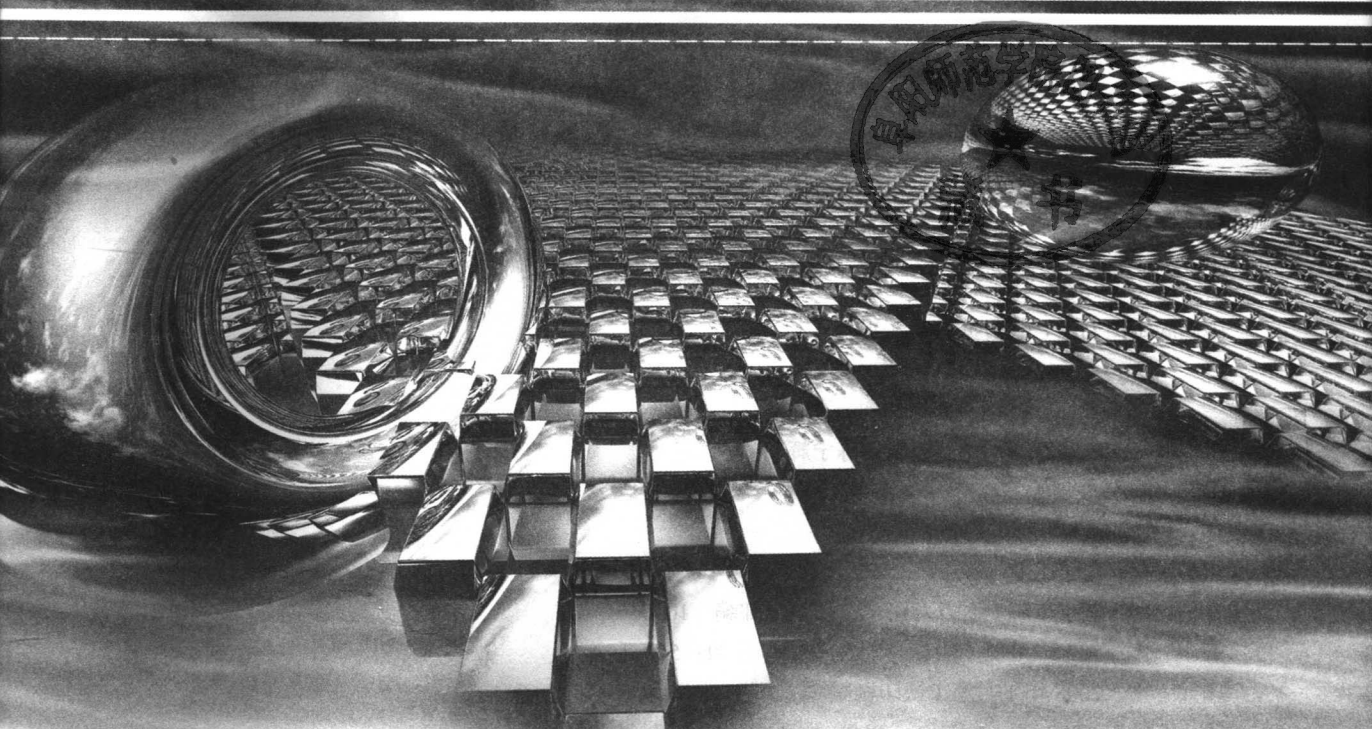


TP332/126

2008

基于ARM的 嵌入式系统开发

刘 岚 尹 勇 李京蔚 编著



北方工业大学图书馆



C00061472

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书在比较 51 单片机与嵌入式系统的基础上,介绍嵌入式系统的开发原理、思路和流程,并对利用汇编语言和 C 语言进行嵌入式系统开发进行了详细讲解。全书共分 9 章,内容包含嵌入式系统开发过程中的原理图设计、芯片选择、驱动程序设计、上层应用程序设计、嵌入式系统的体系结构、嵌入式系统的指令集、嵌入式系统的开发工具、嵌入式系统的硬软件开发、嵌入式实时操作系统的移植等,重点介绍了基于 S3C44B0 嵌入式系统的硬件单元设计和软件模块设计,并在此基础上介绍了实时操作系统 μ C/OS-II 在 S3C44B0 上的移植。对读者而言,只需要具备模拟、数字电路和单片机的基础,就能通过本书熟练掌握嵌入式系统的硬件和软件开发技术。

本书兼顾嵌入式系统开发的原理、关键芯片的使用、嵌入式系统的开发工具、具体项目的硬软件开发,可供通信、电子、控制及计算机专业的工程师参考,也可作为相关专业高等院校本科生和研究生学习嵌入式系统设计和开发的辅助教材。

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

基于 ARM 的嵌入式系统开发/刘岚,尹勇,李京蔚编著. —北京:电子工业出版社,2008.1

(嵌入式技术与应用丛书)

ISBN 978-7-121-05561-4

I. 基… II. ①刘…②尹…③李… III. 微处理器, ARM—系统设计 IV. TP332

中国版本图书馆 CIP 数据核字(2007)第 191385 号

责任编辑:高买花 特约编辑:陈宁辉

印 刷: 北京牛山世兴印刷厂

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: 22.5 字数: 576 千字

印 次: 2008 年 1 月第 1 次印刷

印 数: 4 000 册 定价: 38.00 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。

前 言

嵌入式技术是当前发展最快、应用最广、最有发展前景的应用技术之一。嵌入式系统是计算机软件与硬件的综合体，它是以应用为中心，以计算机技术为基础，软硬件可裁减，从而能够适应实际应用中对功能、可靠性、成本、体积、功耗等严格要求的专用计算机系统。嵌入式计算机在应用数量上远远超过了各种通用计算机，一台通用计算机的外部设备中包含了 5~10 个嵌入式微处理器，键盘、鼠标、软驱、硬盘、显示卡、显示器、Modem、网卡、声卡、打印机、扫描仪、数码相机、USB 集线器等均是由嵌入式处理器进行控制的。在工业制造、过程控制、通信、仪器、仪表、汽车、船舶、航空、航天、军事装备、消费类产品等领域，嵌入式计算机都有用武之地。

目前，嵌入式系统开发和设计方面的书籍已经很多了，然而，大多数书籍或是以讲概念、说原理为主，或是以翻译嵌入式微处理器的数据手册为主。一些使用性比较强的书籍则要求读者必须具有一定的专业知识，特别是需要掌握很多硬件知识和底层软件驱动知识才能看懂。为此，本书在体系结构和内容安排上注意满足不同层次读者的不同要求，积极为读者搭建学习平台，拓展应用空间。

本书以 51 系列单片机作为学习比照的对象，读者若具备模拟电路、数字电路、单片机、C/C++ 基础便可以立即进行嵌入式系统的学习，而不是像传统的嵌入式系统教程那样将大量篇幅放在原理和概念上。

全书共 9 章。前两章是对 ARM 处理器的一般介绍，内容简明易懂，将这些内容与 51 系列单片机对比学习，可以很快了解 ARM 处理器的基本原理。第 3 章介绍 ARM 微处理器的指令系统。第 4 章介绍 ARM 汇编程序设计，提供了大量的嵌入式系统汇编子程序及例子。由于嵌入式系统的 C 语言开发与标准 C 语言或 Turbo C 基本相同，因此第 5 章仅简单介绍嵌入式系统的 C 语言程序设计与标准 C 语言或 Turbo C 不同的地方。第 6 章讲述如何使用嵌入式系统的开发工具，并给出了具体、完整的开发实例。第 7 章讲述嵌入式系统的硬件模块设计，详尽细致，让读者知其然同时又知其所以然。第 8 章介绍嵌入式系统软件的设计方法，并以 S3C44B0 为参照，结合第 7 章的硬件开发过程，详细给出各个软件模块的开发和实现。在嵌入式系统中，一般都需要进行操作系统的移植，因此，第 9 章在介绍嵌入式实时操作系统基本概念的基础上，针对 S3C44B0 微处理器，详细介绍 μ C/OS-II 操作系统在 S3C44B0 上的移植。

通过阅读本书，读者可以了解嵌入式系统的基本概念和基本理论，能够进行嵌入式系统的硬件模块设计和开发，能够掌握嵌入式系统软件设计的基本方法，并能够使用 ADS 开发工具进行程序编辑、编译、链接和调试，成为一个初步懂得 ARM 指令并可以进行嵌入式系统项目开发和应用的工程师。

本书由武汉理工大学刘岚教授进行策划、指导和统稿，武汉理工大学的尹勇博士负责

854512407

第4章～第9章的编写，武汉铁路局机械化工务段的李京蔚女士负责第1章～第3章的编写。在编写过程中，参考了许多网络文章，并得到武汉理工大学嵌入式实验室师生的帮助和电子工业出版社的大力支持，在此一并感谢。

限于编者的水平和经验，书中难免存在错误和不妥之处，敬请广大读者批评指正。

交流请联系：yiyng_hust@126.com（编著者）或 gmholife@hotmail.com（策划编辑）。

编著者

2007年10月于武汉

目 录

第 1 章 嵌入式系统简介	(1)
1.1 嵌入式系统介绍	(1)
1.2 嵌入式处理器	(3)
1.2.1 嵌入式处理器分类	(3)
1.2.2 嵌入式处理器内核	(4)
1.3 嵌入式系统的组成	(5)
1.4 嵌入式系统的开发过程	(8)
1.5 嵌入式系统的应用	(8)
第 2 章 ARM 体系结构	(10)
2.1 ARM 公司简介	(10)
2.2 ARM 微处理器系列	(11)
2.2.1 ARM7 系列	(12)
2.2.2 ARM9 系列	(12)
2.2.3 ARM9E 系列	(13)
2.2.4 ARM10E 系列	(13)
2.2.5 SecurCore 系列	(14)
2.2.6 Intel 的 Xscale	(14)
2.2.7 Intel 的 StrongARM	(14)
2.3 ARM 微处理器结构	(14)
2.3.1 RISC 体系结构	(15)
2.3.2 ARM 处理器的工作状态	(15)
2.3.3 ARM 处理器模式	(16)
2.3.4 ARM 处理器的寄存器结构	(22)
2.3.5 处理器的存储器结构	(23)
2.3.6 异常	(28)
第 3 章 ARM 微处理器的指令系统	(28)
3.1 ARM 指令的寻址方式	(28)
3.1.1 立即寻址	(28)
3.1.2 寄存器寻址	(29)
3.1.3 寄存器间接寻址	(29)
3.1.4 基址寻址	(29)
3.1.5 相对寻址	(30)
3.1.6 多寄存器寻址	(30)
3.1.7 堆栈寻址	(30)



3.1.8	寄存器偏移寻址	(31)
3.2	ARM 指令格式	(32)
3.2.1	指令格式	(32)
3.2.2	指令集的机器码格式	(33)
3.2.3	指令条件码	(33)
3.3	ARM 指令集	(34)
3.3.1	数据处理指令	(34)
3.3.2	乘法与乘加指令	(40)
3.3.3	跳转指令	(43)
3.3.4	程序状态寄存器访问指令	(45)
3.3.5	加载/存储指令	(47)
3.3.6	批量数据加载/存储指令	(50)
3.3.7	数据交换指令	(52)
3.3.8	移位操作	(53)
3.3.9	软中断指令	(55)
3.3.10	协处理器指令	(56)
3.4	Thumb 指令及应用	(56)
第 4 章	ARM 汇编程序设计	(58)
4.1	ARM 伪指令	(58)
4.1.1	符号定义 (Symbol Definition) 伪指令	(58)
4.1.2	数据定义 (Data Definition) 伪指令	(60)
4.1.3	汇编控制 (Assembly Control) 伪指令	(64)
4.1.4	其他常用的伪指令	(66)
4.2	汇编语法	(71)
4.2.1	汇编程序源文件	(71)
4.2.2	汇编语句语法	(72)
4.3	汇编语言程序设计	(77)
4.3.1	汇编语言的程序结构	(77)
4.3.2	汇编语言的子程序调用	(78)
4.3.3	宏定义及其作用	(78)
4.3.4	数据比较跳转	(79)
4.3.5	循环	(79)
4.3.6	数据块复制	(80)
4.3.7	堆栈操作	(80)
4.3.8	查表操作	(80)
4.3.9	长跳转	(81)
4.3.10	对信号量的支持	(81)
4.3.11	片上特殊寄存器定义及应用	(81)

4.3.12	片外部件控制	(81)
4.4	汇编程序设计示例	(82)
第5章	ARM 的 C 语言程序设计	(95)
5.1	嵌入式系统中的 C 语言	(95)
5.1.1	概述	(96)
5.1.2	C 语言编程的优势	(97)
5.1.3	嵌入式 C 语言与 ANSI C 的主要区别	(97)
5.1.4	嵌入式 C 语言开发方法	(99)
5.2	嵌入式 C 程序的基本结构	(100)
5.3	ARM 的 C 语言编译器	(100)
5.3.1	C 语言程序中的元素: 头文件、源文件及预处理	(101)
5.3.2	ARM 的 C 语言编译器特性	(107)
5.3.3	ARM 的标识符	(108)
5.3.4	ARM 转义字符	(109)
5.3.5	ARM 基本数据类型	(109)
5.4	ARM 运行时库	(109)
5.4.1	库类型和选项	(110)
5.4.2	库路径结构	(112)
第6章	ARM 开发工具的使用	(112)
6.1	开发工具概述	(116)
6.2	ADS 开发工具	(116)
6.2.1	ADS 开发工具的组成及功能	(116)
6.2.2	ADS 软件的安装	(121)
6.3	ADS 的命令行工具	(121)
6.3.1	命令行工具介绍	(124)
6.3.2	ADS 命令行编译工具的用法	(125)
6.3.3	ADS 命令行链接工具 armlink 的用法	(131)
6.4	ADS 的集成开发环境	(132)
6.4.1	CodeWarrior 模块	(133)
6.4.2	AXD 模块	(134)
6.5	用 ADS 开发应用程序	(134)
6.5.1	新建一个工程	(140)
6.5.2	工程设置	(142)
6.5.3	工程编译	(145)
6.5.4	工程调试	(157)
第7章	ARM 硬件系统设计	(157)
7.1	硬件的选择	(157)
7.1.1	处理器的选择	(157)

7.1.2	外围芯片的选择	(158)
7.2	嵌入式硬件系统的结构	(159)
7.3	S3C44B0 芯片概述	(160)
7.3.1	简介	(160)
7.3.2	S3C44B0 芯片的特性	(161)
7.3.3	S3C44B0 芯片的内部功能框图	(164)
7.3.4	S3C44B0 芯片的引脚定义	(165)
7.4	单元电路设计	(168)
7.4.1	电源电路	(168)
7.4.2	晶振电路	(169)
7.4.3	看门狗与复位电路	(171)
7.4.4	存储单元电路	(172)
7.4.5	串行接口电路	(179)
7.4.6	JTAG 接口电路	(180)
7.4.7	I ² C 接口电路	(182)
7.4.8	网络接口电路	(183)
7.4.9	LCD 接口电路	(187)
7.4.10	触摸屏控制电路	(190)
7.4.11	I ² S 接口电路	(192)
7.4.12	USB 接口电路	(195)
7.4.13	PS/2 键盘接口电路	(199)
7.4.14	I/O 接口电路	(201)
7.4.15	A/D 转换电路	(202)
7.5	S3C44B0 最小系统	(203)
7.6	硬件电路板设计事项	(205)
7.7	硬件电路的调试	(207)
第 8 章	ARM 的软件开发	(210)
8.1	Bootloader 程序的编写	(210)
8.1.1	概述	(210)
8.1.2	Bootloader 简介	(211)
8.1.3	S3C44B0 的 Bootloader	(211)
8.2	LCD 程序的编写	(213)
8.2.1	字符显示	(213)
8.2.2	LCD 绘图函数	(215)
8.3	异步串行口编程	(221)
8.3.1	异步串口简介	(221)
8.3.2	串口函数	(223)
8.4	I/O 操作	(225)

8.4.1 I/O 概述	(225)
8.4.2 I/O 函数	(226)
8.5 I ² C 驱动编程	(227)
8.5.1 I ² C 简介	(227)
8.5.2 相关函数	(229)
8.6 A/D 转换编程	(234)
8.7 网卡编程	(235)
8.7.1 网卡编程简介	(235)
8.7.2 基本函数	(237)
8.8 I ² S 编程	(241)
8.8.1 I ² S 简介	(241)
8.8.2 相关函数	(242)
8.9 触摸屏编程	(244)
8.9.1 ADS7843 编程简介	(244)
8.9.2 触摸屏函数	(246)
8.10 PS/2 键盘编程	(248)
8.10.1 PS/2 协议介绍	(248)
8.10.2 PS/2 函数	(250)
8.11 USB Host 编程	(255)
8.11.1 USB Host 简介	(256)
8.11.2 USB 传输协议	(256)
8.11.3 Mass storage 协议	(258)
8.11.4 FAT 文件系统	(258)
8.11.5 U 盘的 FAT 结构	(261)
8.11.6 SL811 描述	(263)
8.11.7 相关函数	(265)
第 9 章 μ C/OS-II 操作系统在 S3C44B0 上的移植	(276)
9.1 概述	(276)
9.1.1 嵌入式系统软件构成	(277)
9.1.2 实时操作系统与非实时操作系统	(281)
9.1.3 嵌入式操作系统的选择	(283)
9.2 μ C/OS-II 操作系统简介	(284)
9.3 μ C/OS-II 操作系统在 S3C44B0 上的移植	(284)
9.4 μ C/OS-II 操作系统中应用程序的编写	(292)
附录 S3C44B0 的寄存器	(296)
参考文献	(349)



第1章 嵌入式系统简介

本章简要介绍嵌入式系统的一些基本概念、应用领域、组成及特点,使读者了解嵌入式系统的来源、发展状况以及在国民经济中的应用现状。本章从计算机系统的发展、单片机的使用现状入手,逐步引导读者进入嵌入式系统的殿堂。需要注意的是,读者首先需要具备单片机的基础知识,比如单片机的硬件基本构件、单片机的程序执行流程等,再进一步阅读本书才能起到事半功倍的效果。

1.1 嵌入式系统介绍

单片机功能单一、速度较慢、网络功能不健全,并经常采用串口通信来交换数据和信息,而串口通信在使用中存在抗干扰能力差、传输距离短和传输速率低等缺点。即使采用高速单片机来实现网络传输,由于需要额外的 ROM/RAM 空间,这对于一些内存有限且只能工作在单片方式的单片机来说会受到限制。在网络连接方面,现在已经有许多移植到单片机中的 TCP/IP 协议栈,其内装的 TCP/IP 协议栈过于简单,没有内置多任务操作系统,对于大量网络广播包数据、病毒攻击响应有困难,非常容易导致主机网络掉线。同时,在工业控制领域采用单片机控制,逻辑控制烦琐,增加了维修调试的环节,线路板驱动部件老化程度较高,如果出现问题将更换整块电路板,造成维修成本高。

从广义的概念上讲,嵌入式系统一般指非 PC 系统,即有计算机功能但又不称为计算机的设备或装置。从这种意义上讲,单片机系统也是一种嵌入式系统。由于单片机与嵌入式系统的芯片,比如 ARM,在功能结构上的差异,故应用层次上就有很大的不同。如果将单片机与 ARM 做一个比较,从主频方面比较,现在使用最普遍的 ARM7 已经达到 200MHz~300MHz 左右,ARM9 可达到 700MHz,这显然是单片机所无法相比的;ARM 采用的哈佛结构、RISC 指令集及 3~4 级流水线技术,对于传统的 51 等单片机都有其性能上无法比拟的优势,所以 ARM 系列芯片对于现在的多媒体、网络控制等嵌入式系统有很强的适用性,广泛应用于手机、PDA、医疗设备、汽车电子等领域。同时 ARM 可以支持很多现在流行的操作系统,如 WinCE, PalmOS, Linux, Embedded Linux, VxWorks 及 μ C/OS 等,它可用于多媒体技术、文件管理、内存管理及联网支持等方面,这都是普通单片机所不能达到的。然而对传统的简单控制而言,8 位机应该已经够用了,而这种简单的控制应用还大量存在,比如彩灯控制、温度控制、电子表、洗衣机等,这决定了“简单”的单片机永远无法退出它所占有的一席之地。所以 51 系列及同档次的单片机并没有退出应用市场,在今后很长的时间内都还会继续主导潮流,但在高端产品的应用领域就很难再发挥作用。

从狭义的概念上讲,嵌入式系统是以应用为中心,即以计算机技术为基础,并且软硬

件可裁减,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。一般情况下,嵌入式系统由嵌入式微处理器、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。其中,最突出的是嵌入式操作系统。

在一般的单片机系统中是没有操作系统的,在其中运行的只是一个循环等待程序,在需要进行控制时,通过中断或条件判断语句去执行相应的控制语句。也就是说,随时等待中断的发生并予以响应,然后再进入循环等待。在单片机程序中,将代码烧入 Flash 存储器,上电开始执行时,从内存地址 0000h 开始执行,在没有外部事件,比如外部中断引脚 1 电平或升降源发生变化时,程序就等待在语句“SJMP \$”处,一旦有了外部事件触发外部中断引脚 1,程序就跳到外部中断入口“LJMP 0050h”处,在该处是一条跳转语句,跳转到内存地址 0050h 处,执行中断需要执行的函数 Function1()、Function2()、Function3() 等。执行完中断程序的函数后,程序就停在语句“SJMP \$”处等待下一个中断的发生,周而复始。

然而,目前的控制系统,比如电子医疗保健产品中的电脑断层成像系统、心脏起搏器、监护仪等,所要完成的工作越来越复杂,需要执行的程序越来越庞大,需要管理的外设越来越多,如果还是采用循环等待或中断方式来执行任务,那么,复杂的任务和庞大的程序可能会使对整个系统的管理束手无策。在这种情况下,拥有多任务操作系统的嵌入式系统就打破局面、脱颖而出了。另外,在一个完整的、具有统一编程规范的操作系统基础上使用高级语言开发出的应用程序,还可以具有良好的可移植性,这使嵌入式系统在教育方面更具有广泛性。

在嵌入式系统中,一般都移植了专门的嵌入式操作系统,如 Embedded Linux, VxWorks 及 μ COS-II 等,这些操作系统中很多具有源代码开放、价格低廉甚至免费、功能强大又易于移植的特性,同时它们占用内存空间小、执行效率高、方便进行个性化定制和软件固化存储,还能很方便地支持网络文件系统且内嵌 TCP/IP 协议,使系统具有任务调度、同步机制、中断处理、文件管理等功能。就手机上所使用的嵌入式系统而言,一般手机具有下载图片、铃音等功能,这些图片或铃音都是以文件的形式存在的,因此要求手机具有文件管理的功能;有时候手机在通话时可能来了一条短消息,为了能同时处理这两项,要求手机具有任务调度和同步机制的功能;有些手机具有无线上网功能,因此要求手机具有 TCP/IP 协议栈,等等。所有这些,都是在嵌入式操作系统的基础上进行的,是传统的单片机所无法实现的。

另外,从硬件的角度考虑,单片机的引脚资源和可扩展的硬件资源非常有限,比如 AT89C51 单片机只有 40 个引脚,AT89C2051 单片机只有 20 个引脚,除去数据信号线引脚、地址信号线引脚、电源引脚、地引脚、晶振引脚和复位引脚等芯片必需的功能引脚外,用户能使用的引脚非常有限。就 AT89C2051 而言,由于它的 I/O 线很少,导致它无法外加数据 RAM 和程序 ROM,片内 Flash 存储器也少,使很多应用功能无法用它实现。

可以看出,单片机的内存资源、定时器/计数器资源、中断资源和其他 I/O 资源非常少,其 CPU 工作频率也非常低,即使可以在单片机中移植一些精巧的嵌入式操作系统,但其非常紧缺的资源可能完全用在操作系统的运行上了,再想应对用户需要的各种控制和管理就



会显得力不从心。

1.2 嵌入式处理器

1.2.1 嵌入式处理器分类

嵌入式系统的核心部件是各种类型的嵌入式处理器，目前，据不完全统计，全世界嵌入式处理器的品种总量已经超过 1000 多种，流行体系结构有 30 多个系列。根据其发展现状来看，嵌入式计算机可以分成下面几类。

1. 嵌入式微处理器 (Embedded Microprocessor Unit, EMPU)

嵌入式微处理器的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母板功能，这样可以大幅度减小系统体积和功耗。为了满足嵌入式应用的特殊要求，嵌入式微处理器虽然在功能上和标准微处理器基本一样，但在工作温度、抗电磁干扰、可靠性等方面一般都分别做了各种增强。与工业控制计算机相比，嵌入式微处理器具有体积小、重量轻、成本低、可靠性高的优点，但是在电路板上必须包括 ROM、RAM、总线接口、各种外设等器件，从而降低了系统的可靠性及技术保密性。嵌入式微处理器目前主要有 Am186/88、386EX、SC-400、Power PC、68000、MIPS、ARM 系列等。近年来，德国、日本的一些公司也开发出了“火柴盒”式、名片大小的嵌入式计算机系列 OEM 产品。

2. 嵌入式微控制器 (Microcontroller Unit, MCU)

嵌入式微控制器又称单片机，顾名思义，就是将整个计算机系统集成到一块芯片中。嵌入式微控制器一般以某一种微处理器内核为核心，芯片内部集成 ROM/EPROM、RAM、总线、总线逻辑、定时器/计数器、WatchDog、I/O、串行口、脉宽调制输出、A/D、D/A、Flash RAM、EEPROM 等各种必要功能和外设。微控制器的片上外设资源一般比较丰富，适合于控制应用，因此称微控制器。嵌入式微控制器目前的品种和数量最多，比较有代表性的通用系列包括 8051、P51XA、MCS-251、MCS-96/196/296、C166/167、MC68HC05/11/12/16、68300 等。

3. 嵌入式 DSP 处理器 (Embedded Digital Signal Processor, EDSP)

嵌入式 DSP 处理器对系统结构和指令进行了特殊设计，使其适合于执行 DSP 算法，编译效率及指令执行速度较高。在数字滤波、FFT、谱分析等方面 DSP 算法正在大量引入嵌入式领域，DSP 应用正在从以普通指令实现 DSP 功能的通用单片机，过渡到采用嵌入式 DSP 处理器阶段。嵌入式 DSP 处理器比较有代表性的产品是 Texas Instruments 的 TMS320 系列和 Motorola 的 DSP56000 系列。TMS320 系列处理器包括用于控制的 C2000 系列，用于移动通信的 C5000 系列，以及性能更高的 C6000 系列和 C8000 系列。DSP56000 系列目前已经发展成为 DSP56000，DSP56100，DSP56200 和 DSP56300 等几个不同系列的处理器。

4. 嵌入式片上系统 (System on Chip, SoC)

随着 EDI 的推广和 VLSI 设计的普及化, 以及半导体工艺的迅速发展, 在一个硅片上实现一个更为复杂的系统的时代已来临, 这就是嵌入式片上系统 (SoC)。各种通用处理器内核将作为嵌入式片上系统设计公司的标准库, 与许多其他嵌入式系统外设一样, 成为 VLSI 设计中一种标准的器件。它用标准的 VHDL 等语言描述, 存储在器件库中, 用户只需定义出其整个应用系统, 仿真通过后就可以将设计图交给半导体工厂制作样品。这样除个别无法集成的器件以外, 整个嵌入式系统大部分均可集成到一块或几块芯片中, 应用系统电路板将变得很简洁。嵌入式片上系统可以分为通用和专用两类, 通用系列包括 Infineon 的 TriCore, Motorola 的 M-Core, 某些 ARM 系列器件, Echelon 和 Motorola 联合研制的 Neuron 芯片等。专用嵌入式片上系统一般专用于某个或某类系统中, 不为一般用户所知。一个有代表性的产品是 Philips 的 Smart XA, 它将 XA 单片机内核和支持超过 2048 位复杂 RSA 算法的 CCU 单元制作在一块硅片上, 形成一个可加载 Java 语言或 C 语言的专用嵌入式片上系统, 可用于公众互联网安全方面。

1.2.2 嵌入式处理器内核

当前的嵌入式微处理器目前主要有 Am186/88、386EX、SC-400、Power PC、68000、MIPS、ARM 系列等, 它们都以 RISC (Reduced Instruction Set Computer) 的 MCU 架构为主, 其中 RISC 代表 MCU 的所有指令。RISC 指令集的每条指令简单, 执行的动作少, 但整体的时钟速度可以很高, 通常可以提高 CPU 性能。与 RISC 相对应的是 CISC, 即复杂指令集计算机, 指的是另一种 CPU 指令集。CISC 指令集的每条指令复杂, 功能丰富, 相对于 RISC 指令集, 执行同一个功能所需的指令更少, 而执行每条指令的时间会更长。嵌入式领域全部是 RISC 指令集的处理器内核。处理器内核是一个设计, 并不是一个芯片。嵌入式内核的设计一般追求高速度、低功耗、易于集成。目前, 嵌入式处理器内核主要有如下几种。

1. ARM (Advanced RISC Machines)

ARM 公司是全球领先的 16 位/32 位 RISC 微处理器知识产权设计供应商, 该公司自己不生产芯片, 只设计内核。ARM 内核的设计技术被授权给数百家的半导体厂商, 制成不同的 SoC 芯片。ARM 公司通过转让高性能、低成本、低功耗的 RISC 微处理器、外围和系统芯片设计技术给合作伙伴, 使这些合作伙伴能够用这些技术来生产各具特色的芯片。ARM 已成为移动通信、手持设备、多媒体数字消费嵌入式解决方案的 RISC 标准。ARM 处理器有三大特点: 小体积、低功耗、低成本而高性能。ARM 处理器是 16 位/32 位双指令集。ARM 处理器目前有 5 个系列产品: ARM7、ARM9、ARM9E、ARM10 和 SecurCore。其中 ARM7 是低功耗的 32 位内核, 最适合应用于对价位和功耗敏感的产品, 它又分为应用于实时环境的 ARM7TDMI、ARM7TDMIS, 以及适用于开放平台的 ARM720T 和适用于 DSP 运算及支持 Java 的 ARM7EJ 等几种。其中, ARM7TDMI 处理器是 ARM7 处理器系



列成员之一，是目前应用最广的 32 位高性能嵌入式 RISC 处理器。

2. MIPS (Microprocessor without Interlocked Pipeline Stages)

MIPS 是一种处理器内核标准，它是由 MIPS 技术公司开发的。MIPS 技术公司是一家设计制造高性能、高档次的嵌入式 32 位和 64 位处理器的厂商，在 RISC 处理器方面占有重要地位。MIPS 技术公司既开发 MIPS 处理器结构，又自己生产基于 MIPS 的 32 位/64 位芯片。

3. PowerPC

PowerPC 是 IBM 和 Motorola 公司共有的内核。PowerPC 架构的特点是伸缩性好，方便灵活。PowerPC 处理器品种很多，既有通用的处理器，又有嵌入式控制器和内核，应用范围非常广泛，从高端的工作站、服务器到桌面计算机系统，从消费类电子产品到大型通信设备等各个方面，如中兴通信、华为等公司在其通信产品中就大量采用了 Motorola 的 PowerPC 家族的系列嵌入式处理器。MPC860 和 MPC8260 是其最经典的两款 PowerPC 内核的嵌入式处理器。目前基于 PowerPC 内核的处理器有 IBM 公司开发的 PowerPC 405 嵌入式处理器和 Motorola 开发的 MPC823e 微处理器。

4. 68000/Coldfire

68000/Coldfire 是 Motorola 公司独有的内核。68000 内核是最早在嵌入式领域广泛应用的内核，其著名的代表芯片是 68360。Coldfire 继承了 68000 的特点并与它兼容，最近，Motorola 公司已经发布第五版本内核，Coldfire 内核中集成了 DSP 模块、CAN 总线模块及一般嵌入式处理器所集成的外设模块，从而形成了一系列的嵌入式处理器，在工业控制、机器人研究、家电控制等领域被广泛采用。

本书以 ARM 系列微处理器为主进行叙述，在本书的各个章节中，如果没有特殊说明，所采用的微处理器都是基于 ARM 的。

1.3 嵌入式系统的组成

嵌入式系统一般指非 PC 系统，它包括硬件和软件两部分。硬件包括处理器/微处理器、存储器、外设器件和 I/O 端口、图形控制器等。软件部分包括操作系统软件（要求实时和多任务操作）和应用程序。有时设计人员把这两种软件组合在一起，应用程序控制着系统的运作和行为，而操作系统控制着应用程序编程与硬件的交互作用。

典型的嵌入式系统的体系结构可以分成 4 个部分：嵌入式处理器、嵌入式外围设备、嵌入式操作系统和嵌入式应用软件，如图 1.1 所示。

ARM 嵌入式系统硬件平台一般包括一个以 ARM 为内核的处理器、存储器和必要的外部接口与设备。在实际应用中，嵌入式系统硬件配置十分精简，除了 CPU 和基本外围



电路外，其余硬件都可以根据成本和功能进行裁减。

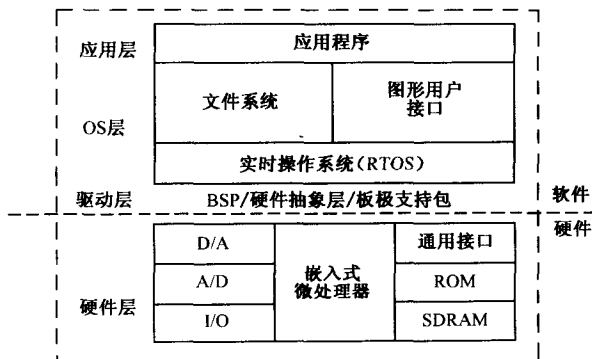


图 1.1 典型嵌入式系统的体系结构

嵌入式外围接口电路和设备接口包括：

- 存储器：静态易失性存储器（RAM、SRAM）、动态存储器（DROM）、非易失性存储器（MASK ROM、EPROM、EEPROM、Flash）、硬盘、软盘、CDROM 等。
- 通信接口：RS232、USB、IrDA、SPI（串行外围设备接口）、I²C、CAN 总线接口、蓝牙、以太网、IEEE 1394 和通用可编程接口等。
- I/O 设备：LCD、CRT、键盘、鼠标等。
- 电源及辅助设备。

典型嵌入式系统的硬件组成如图 1.2 所示。

嵌入式系统软件由芯片级/板级初始化 BSP、操作系统、GUI 文件系统设备驱动应用程序 API 和相应的用户应用程序构成，如图 1.3 所示。

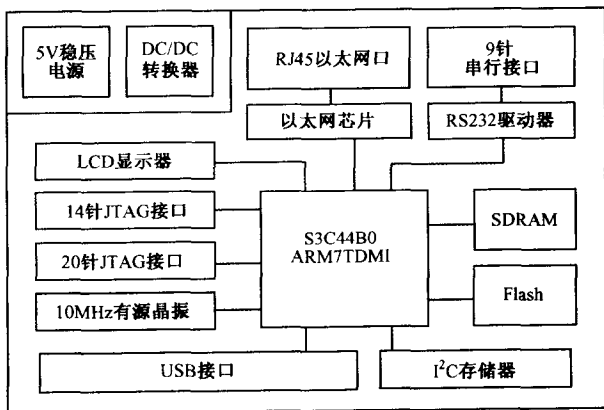


图 1.2 典型嵌入式系统的硬件组成

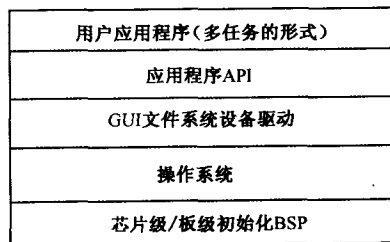


图 1.3 典型嵌入式系统软件的组成

为了使嵌入式系统的开发更加方便和快捷，需要有专门负责管理存储器分配、中断处理、任务调度等功能的软件模块，这就是嵌入式操作系统。操作系统是一段嵌入在目标代码中的软件，用户的其他应用程序都建立在操作系统之上。嵌入式操作系统通常包括与硬



件相关的底层驱动软件、系统内核、设备驱动接口、通信协议、图形界面、标准化浏览器等。嵌入式操作系统具有通用操作系统的基本特点，能够有效地管理越来越复杂的系统资源；能够把硬件虚拟化，使开发人员从繁忙的驱动程序移植和维护中解脱出来；能够提供库函数、驱动程序、工具集及应用程序。

芯片级和板级初始化引导代码是嵌入式系统上电复位后第一个执行的代码，当系统加电开始运行或系统复位时，代码开始执行的地址对 ARM 来说是 0x00000000H。其主要目的是执行系统自检、系统初始化（硬件、软件）及引导操作系统，最后进入用户的 main() 函数。

操作系统将 CPU 时间、中断、I/O、定时器等资源都包装起来，留给用户一个标准的 API，并根据各个任务的优先级，合理地在不同任务之间分配 CPU 时间。

与通用操作系统相比较，嵌入式操作系统在系统实时性、硬件依赖性、软件固化性及应用专用性等方面，具有以下鲜明的特点：

- (1) 代码占用空间小：适合在嵌入式系统的有限存储空间中运行。
- (2) 面向应用，可裁减和移植：系统功能可根据需求裁减、调整和生成，以满足最终产品的设计需求，可进一步缩小代码占用空间。
- (3) 实时性强：软件代码要求在 ROM 中固态存储，以提高速度，软件代码要求高质量、高可靠性和实时性。
- (4) 可靠性高：嵌入式系统可无须人工干预独立运行，并处理各类事件和故障。
- (5) 专用性强：嵌入式操作系统和硬件的结合非常紧密，一般要针对硬件进行系统的移植。

嵌入式操作系统一般是实时系统（Real Time System）。实时系统是一种能够在指定或确定时间内完成的系统功能，并且对外部事件和内部事件在同步或异步时间内能做出及时响应的系统。在实时系统中，操作的正确性不仅依赖于逻辑设计的正确程度，而且与这些操作进行的时间有关，也就是说，实时系统对逻辑和时序的要求非常严格，如果逻辑和时序控制出现偏差将会产生严重后果。

嵌入式操作系统一般也是多任务（Multi-Task）系统。多任务的并发执行通常依赖于一个多任务操作系统（OS），多任务操作系统的核心是系统调度器，它使用任务控制块（TCB）来管理任务调度功能。TCB 包括任务的当前状态、优先级、要等待的事件或资源、任务程序码的起始地址、初始堆栈指针等信息。调度器在任务被激活时，要用到这些信息。此外，TCB 还被用来存放任务的“上下文”。任务的上下文就是当一个执行中的任务被停止时，所要保存的所有信息。通常，上下文就是计算机当前的状态，也即各个寄存器的内容。当发生任务切换时，当前运行任务的上下文被存入 TCB，并将要被执行的任务的上下文从它的 TCB 中取出，放入各个寄存器中。选择多任务还是单任务方式依赖于软件的体系是否庞大，绝大多数手机程序都是多任务的，但也有一些小灵通的协议栈是单任务的，没有操作系统，它们的主程序轮流调用各个软件模块的处理程序，模拟多任务环境。

图形化用户界面（Graphic User Interface, GUI）、文件系统、USB 驱动、TCP/IP 协议栈、键盘、鼠标、打印机等驱动程序在嵌入式操作系统之上运行，而与用户交互的应用软件则通过调用文件系统和 GUI 的功能调用（API）来实现。对于大多数嵌入式系统，

