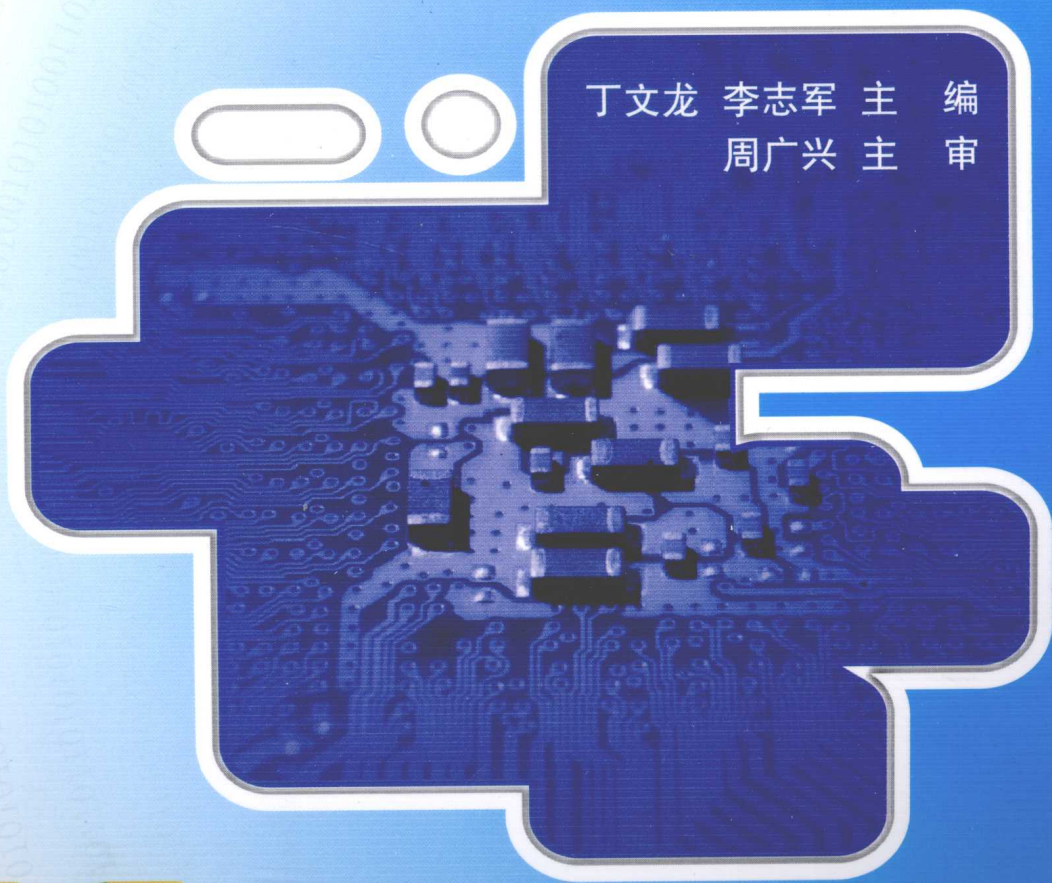




21世纪全国本科院校电气信息类**创新型**应用人才培养规划教材

ARM嵌入式系统 基础与开发教程

丁文龙 李志军 主 编
周广兴 主 审



北京大学出版社
PEKING UNIVERSITY PRESS

21 世纪全国本科院校电气信息类创新型应用人才培养规划教材

ARM 嵌入式系统基础与开发教程

主 编 丁文龙 李志军
副主编 刘建霞 张广兰
主 审 周广兴



北京大学出版社
PEKING UNIVERSITY PRESS

内 容 简 介

本书全面、系统地介绍 ARM7TDMI 处理器的体系结构、编程模型、指令系统,以及 ADS1.2、RealView MDK、PROTEUS 三个开发环境。同时,还以基于 ARM7TDMI 的应用处理器 LPC2124 为核心,详细介绍了 ARM 嵌入式系统设计、相关接口技术、嵌入式软件设计等内容。接口技术的内容涵盖了 GPIO、中断、定时器、UART、RTC、A/D、I²C、SPI、PWM,并列举了大量实例。本书还介绍了 μ C/OS-II 软件体系结构、文件结构及移植规划等方面的内容。在第 9 章提供了一个综合实例。

本书从教学和应用的角度出发,具有较强的系统性和实用性。内容充实,图文并茂,选材精炼,论述由浅入深,每章除配有大量的例子和思考与练习题外,还配备了一定量的实例分析。

本书可作为高等院校计算机科学与技术、通信工程、电子信息工程、自动化等本科专业的嵌入式系统课程教材,也可作为嵌入式应用工程技术人员的参考书。

图书在版编目(CIP)数据

ARM 嵌入式系统基础与开发教程/丁文龙,李志军主编. —北京:北京大学出版社,2010.6
(21 世纪全国本科院校电气信息类创新型应用人才培养规划教材)
ISBN 978-7-301-17318-3

I. ①A… II. ①丁…②李… III. ①微处理器—系统设计—高等学校—教材 IV. ①TP332
中国版本图书馆 CIP 数据核字(2010)第 104939 号

书 名: ARM 嵌入式系统基础与开发教程

著作责任者: 丁文龙 李志军 主编

策划编辑: 程志强

责任编辑: 程志强

标准书号: ISBN 978-7-301-17318-3/TP·1110

出版者: 北京大学出版社

地 址: 北京市海淀区成府路 205 号 100871

网 址: <http://www.pup.cn> <http://www.pup6.com>

电 话: 邮购部 62752015 发行部 62750672 编辑部 62750667 出版部 62754962

电子邮箱: pup_6@163.com

印 刷 者: 三河市北燕印装有限公司

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 19.75 印张 459 千字

2010 年 6 月第 1 版 2010 年 6 月第 1 次印刷

定 价: 36.00 元

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

举报电话: 010-62752024

电子邮箱: fd@pup.pku.edu.cn

前 言

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。随着现代社会信息化进程的加快，嵌入式系统被广泛地应用于军事、家用、工业、商业、办公、医疗等社会各个领域。

学习嵌入式系统要以应用为导向，因此，建议学习者首先选择一款主流芯片，以点带面、循序渐进地学习。目前，以 ARM 为核心的嵌入式技术逐渐成为我国嵌入式教学的主流。ARM 作为一种 32 位的高性能、低成本的嵌入式 RISC 微处理器，已得到最广泛的应用。NXP 公司推出的基于 ARM7TDMI(-S)的 32 位微控制器 LPC2000 系列，具有很高的性价比，其接口模块丰富，使用范围广，在 PROTEUS 环境中可仿真，相比其他的嵌入式处理器更便于学习和掌握。 μ C/OS-II 是著名的源代码公开的嵌入式实时操作系统，是专门为嵌入式系统的应用而设计的，主体代码采用 ANSI(美国国家标准学会)C 语言编写，十分易于应用和移植。 μ C/OS-II 的特点在于源代码小且完全公开，有详尽的解释，科研和教学可免费使用。正因为其小，才便于学习、研究、理解和掌握，也因为其小，更便于广大的低端用户和小系统使用。

本书以 NXP 公司的 LPC2000 系列 ARM 微控制器为平台，介绍了嵌入式系统开发的各个主要环节，将嵌入式软硬件理论讲解和嵌入式应用实例融合在一起，侧重实践，辅以代码加以讲解，从分析的角度来学习嵌入式开发的各种技术。在学习本书之前，读者最好具有数字电路、单片机接口、操作系统、C 语言编程等基础知识。

全书共 9 章，各章分别包括以下内容。

第 1 章嵌入式系统概述，主要讲述嵌入式系统的基础知识，介绍了嵌入式系统的定义、特点、应用、组成结构以及开发的基本流程，并讲述了嵌入式系统的发展趋势，最后给出了学习嵌入式系统的建议。

第 2 章 ARM 体系结构，介绍了 ARM 体系结构的发展和 ARM 系列的产品，讲解了 ARM7 编程模型，如 ARM 处理器的工作状态、工作模式、内部寄存器、异常处理的机制原理等，为后面进行 ARM 开发打好基础。

第 3 章 ARM7TDMI(-S)指令系统，详细介绍了 ARM 处理器的 8 种寻址方式和 ARM 指令集中的基本指令及使用方法。ARM 指令集是基于 ARM 的嵌入式编程的基础。

第 4 章基于 ARM 的嵌入式软件设计，介绍了 ARM 程序设计的过程与方法，包括伪指令的使用、汇编语言编程、嵌入式 C 语言编程、C 语言和汇编语言混合编程等内容，最后对 ARM 程序结构进行了分析。

第 5 章嵌入式系统常用开发工具，介绍了 ADS1.2、RealView MDK、PROTEUS 工具软件的使用。



第 6 章基于 LPC2000 系列的嵌入式应用开发实例, 详细介绍了 LPC2000 系列处理器的内部功能模块, 并对每个功能模块列举了应用实例, 采用 ADS1.2 或 RealView MDK 开发环境进行应用程序设计, 利用 PROTEUS 软件实现功能仿真。

第 7 章嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$ 基础, 首先对嵌入式操作系统做了概述并列出了典型的嵌入式操作系统, 然后介绍了 $\mu\text{C}/\text{OS-II}$ 的特点及软件体系结构和文件结构, 最后重点介绍了 $\mu\text{C}/\text{OS-II}$ 的内核结构。

第 8 章 $\mu\text{C}/\text{OS-II}$ 在 ARM7 上的移植, 详细介绍了 $\mu\text{C}/\text{OS-II}$ 的移植规划及其具体文件编写, 通过应用实例介绍了如何移植 $\mu\text{C}/\text{OS-II}$ 到 LPC2000 处理器上。

第 9 章综合实例——超声波测距仪设计, 通过一个综合实例, 综合了几个常用的接口功能, 通过本章读者可以提高利用已学知识解决实际问题的能力。

本书各章开始之前都配有“本章教学目的与要求”、“本章知识结构”、“引言”3 部分内容, 便于读者认清学习重点, 形成清晰的知识结构体系; 另外各章后均给出了适量的思考与练习题, 通过这些思考与练习题使读者进一步加深对已学知识的理解和记忆。

需要说明的是: 本书中有些电路原理图(如图 5.38)是用仿真设计软件绘制的, 为保持这些图与软件生成的实际图样的一致性且便于读者对照阅读, 在编写过程中编者未将其中不规范的图形符号、文字符号给予订正。

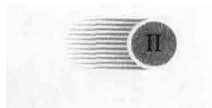
本书第 1、3 章由黑龙江科技学院计算机与信息工程学院的丁文龙编写, 第 6 章由黑龙江科技学院电气与信息工程学院的李志军编写, 第 4、5 章由黑龙江科技学院电气与信息工程学院的刘建霞编写, 第 2、7、8、9 章由黑龙江科技学院信息网络中心的张广兰编写。全书由丁文龙统稿, 由黑龙江科技学院计算机与信息工程学院的周广兴主审。

本书是编者多年来教学实践经验的总结, 也是编者从事嵌入式应用科研工作经验的总结。编写时注重实践操作部分, 尽量避免烦琐、高深的理论介绍, 强调培养学生的动手能力。在本书的编写过程中, 不少一线任课教师提出了许多宝贵的意见, 北京大学出版社对教材的出版给予了大力协作, 对此深表感谢! 本书的编写参考了许多相关的书籍和资料, 编者在此对这些参考文献的作者和出版者表示感谢!

由于编者水平有限, 疏漏之处在所难免, 恳请广大读者批评指正。

编 者

2010 年 3 月

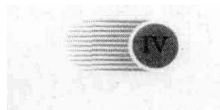


目 录

第 1 章 嵌入式系统概述	1	3.2 ARM 处理器指令概述	40
1.1 嵌入式系统	2	3.2.1 ARM 指令的特点	40
1.1.1 嵌入式系统定义	2	3.2.2 ARM 指令集分类	40
1.1.2 嵌入式系统特点	3	3.2.3 指令格式与条件码	40
1.1.3 嵌入式系统应用	4	3.3 ARM 指令集	42
1.2 嵌入式系统组成结构	5	3.3.1 加载/存储指令	42
1.2.1 嵌入式系统硬件平台	5	3.3.2 数据处理指令	46
1.2.2 嵌入式系统软件结构	8	3.3.3 乘法/乘加指令	53
1.3 嵌入式系统开发基本流程	9	3.3.4 跳转指令	55
1.4 嵌入式系统发展趋势	11	3.3.5 程序状态寄存器访问指令	58
1.5 学习嵌入式系统的建议	12	3.3.6 协处理器指令	60
本章小结	13	3.3.7 异常中断指令	61
思考与练习	13	本章小结	64
第 2 章 ARM 体系结构	14	思考与练习	64
2.1 ARM 简介	15	第 4 章 基于 ARM 的嵌入式软件设计	65
2.1.1 ARM 公司简介	15	4.1 嵌入式系统程序设计方法	66
2.1.2 ARM 体系结构的发展	16	4.1.1 嵌入式软件特点	66
2.1.3 ARM 处理器系列简介	18	4.1.2 嵌入式软件设计流程	67
2.2 ARM7 编程模型	21	4.2 ARM 汇编语言伪指令	68
2.2.1 ARM7TDMI 功能图	21	4.2.1 指导性伪指令	68
2.2.2 ARM 处理器工作状态	22	4.2.2 数据定义伪指令	71
2.2.3 ARM 处理器工作模式	23	4.2.3 输出报告型伪指令	76
2.2.4 ARM 内部寄存器	23	4.2.4 符号定义伪指令	76
2.2.5 指令长度及数据类型	28	4.2.5 汇编语言控制伪指令	82
2.2.6 存储器格式	28	4.2.6 与 ARM 汇编语言组合的 伪指令	87
2.2.7 3 级流水线	29	4.3 ARM 汇编语言程序设计	90
2.2.8 异常处理	30	4.3.1 ARM 汇编语言规范	90
本章小结	34	4.3.2 ARM 汇编语言程序结构	92
思考与练习	34	4.3.3 ARM 汇编程序设计实例	93
第 3 章 ARM7TDMI(-S)指令系统	36	4.4 ARM C 语言程序设计	94
3.1 ARM 处理器寻址方式	37	4.4.1 预处理伪指令	94



4.4.2	函数及函数库.....	97	6.2.2	片外存储器.....	153
4.4.3	ARM C 语言程序设计实例.....	98	6.2.3	存储器映射.....	154
4.5	ARM C 语言与汇编语言混合编程.....	99	6.2.4	存储器重映射及引导块.....	155
4.6	ARM 程序结构分析.....	104	6.2.5	系统启动代码分析.....	156
	本章小结.....	104	6.3	GPIO 端口功能及应用.....	157
	思考与练习.....	105	6.3.1	引脚连接模块寄存器.....	158
第 5 章	嵌入式系统常用开发工具.....	106	6.3.2	GPIO 寄存器描述.....	160
5.1	ARM 开发工具 ADS1.2.....	107	6.3.3	应用实例.....	162
5.1.1	ADS1.2 集成开发环境的组成.....	107	6.4	中断控制功能及应用.....	170
5.1.2	工程的创建、编译和链接.....	110	6.4.1	向量中断控制器概述.....	171
5.1.3	使用 AXD 调试代码.....	116	6.4.2	VIC 寄存器描述.....	172
5.2	ARM 开发工具 RealView MDK.....	118	6.4.3	VIC 基本使用方法.....	176
5.2.1	RealView MDK 简介.....	118	6.4.4	外部中断.....	177
5.2.2	RealView MDK 软件的特点.....	118	6.4.5	外部中断寄存器描述.....	178
5.2.3	ULINK2 仿真器简介.....	119	6.4.6	应用实例.....	179
5.2.4	工程的创建、编译和链接.....	120	6.5	定时器功能及应用.....	184
5.2.5	工程的调试.....	131	6.5.1	定时器概述.....	184
5.2.6	映像文件下载.....	132	6.5.2	寄存器描述.....	185
5.3	ARM 仿真设计软件 PROTEUS.....	132	6.5.3	应用实例.....	189
5.3.1	ISIS 编辑环境.....	133	6.6	UART 串口通信功能及应用.....	193
5.3.2	PROTEUS 设计与仿真流程.....	135	6.6.1	UART0 结构.....	193
5.3.3	PROTEUS 设计与仿真实例.....	136	6.6.2	UART0 寄存器描述.....	195
	本章小结.....	145	6.6.3	UART0 基本操作.....	199
	思考与练习.....	145	6.6.4	应用实例.....	201
第 6 章	基于 LPC2000 系列的嵌入式应用开发实例.....	146	6.7	RTC 功能及应用.....	204
6.1	LPC2000 系列处理器简介.....	147	6.7.1	RTC 概述.....	204
6.1.1	LPC2000 系列处理器特性.....	147	6.7.2	RTC 寄存器描述.....	205
6.1.2	LPC2000 系列处理器结构.....	147	6.7.3	RTC 基本操作.....	208
6.1.3	LPC2000 系列处理器引脚配置.....	149	6.7.4	应用实例.....	209
6.2	存储器管理.....	153	6.8	A/D 转换器功能及应用.....	216
6.2.1	片内存储器.....	153	6.8.1	A/D 转换器概述.....	216
			6.8.2	A/D 寄存器描述.....	216
			6.8.3	A/D 基本操作.....	218
			6.8.4	应用实例.....	219
			6.9	I ² C 接口功能及应用.....	223
			6.9.1	I ² C 接口描述.....	223
			6.9.2	I ² C 寄存器描述.....	224
			6.9.3	I ² C 操作模式.....	227



6.9.4 应用实例	233	8.1.1 编译器选择	274
6.10 SPI 接口功能及应用	235	8.1.2 ARM7 工作模式选择	274
6.10.1 SPI 接口描述	236	8.1.3 支持的指令集	274
6.10.2 SPI 寄存器描述	237	8.2 移植 $\mu\text{C}/\text{OS-II}$	274
6.10.3 SPI 操作模式	240	8.2.1 编写 OS_CPU.H	274
6.10.4 应用实例	243	8.2.2 编写 OS_CPU_C.C	276
6.11 PWM 功能及应用	246	8.2.3 编写 OS_CPU_A.S	280
6.11.1 PWM 概述	246	8.2.4 关于中断及时钟节拍	283
6.11.2 PWM 寄存器描述	248	8.3 应用实例——移植 $\mu\text{C}/\text{OS-II}$ 到 LPC2000	285
6.11.3 PWM 基本操作	253	8.3.1 编写启动代码	285
6.11.4 应用实例	254	8.3.2 挂接 SWI 软件中断	285
本章小结	256	8.3.3 中断及时钟节拍中断	286
思考与练习	256	8.3.4 编写应用程序	286
第 7 章 嵌入式实时操作系统$\mu\text{C}/\text{OS-II}$ 基础	257	本章小结	288
7.1 嵌入式操作系统概述	258	思考与练习	289
7.1.1 嵌入式操作系统	258	第 9 章 综合实例——超声波 测距仪设计	290
7.1.2 典型的嵌入式操作系统	258	9.1 项目功能描述	291
7.2 嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$	260	9.1.1 实现功能	291
7.2.1 $\mu\text{C}/\text{OS-II}$ 概述	260	9.1.2 主要性能指标	291
7.2.2 $\mu\text{C}/\text{OS-II}$ 特点	261	9.2 总体设计	291
7.2.3 $\mu\text{C}/\text{OS-II}$ 软件体系结构	261	9.2.1 超声波测距原理	291
7.2.4 $\mu\text{C}/\text{OS-II}$ 文件结构	263	9.2.2 超声波速度的确定	292
7.3 $\mu\text{C}/\text{OS-II}$ 内核结构	264	9.2.3 超声波测距仪总体设计	292
7.3.1 临界段的处理	264	9.3 硬件电路设计	292
7.3.2 任务和任务状态	265	9.3.1 元器件的选择	293
7.3.3 任务控制块	268	9.3.2 发射电路设计	294
7.3.4 任务调度	268	9.3.3 接收电路设计	294
7.3.5 中断服务	269	9.3.4 语音播报电路设计	295
7.3.6 时钟节拍与时间管理	270	9.3.5 LCD 显示电路设计	297
7.3.7 $\mu\text{C}/\text{OS-II}$ 的初始化	271	9.4 程序设计	300
7.3.8 $\mu\text{C}/\text{OS-II}$ 的启动	272	9.5 测试结果	303
本章小结	272	本章小结	305
思考与练习	272	思考与练习	305
第 8 章 $\mu\text{C}/\text{OS-II}$ 在 ARM7 上的移植	273	参考文献	306
8.1 移植规划	274		

第 1 章

嵌入式系统概述

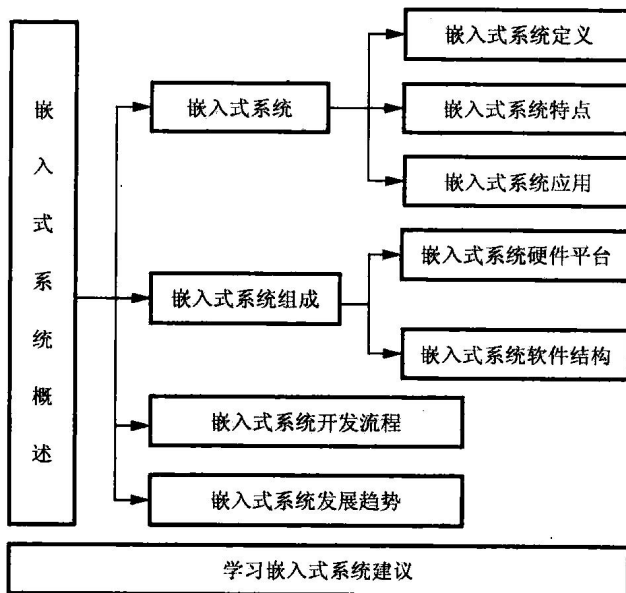


本章教学目的与要求

- 掌握嵌入式系统的定义及特点
- 掌握嵌入式系统的组成结构
- 了解嵌入式系统开发的基本流程
- 了解嵌入式系统的发展趋势



本章知识结构





引言

经过几十年的发展，嵌入式系统在很大程度上改变了人们的生活、工作和娱乐方式，而且这些改变还在加速。嵌入式系统具有很多种类，每类都具有自己独特的个性。图 1.1 所示为常见的嵌入式系统应用实例。



图 1.1 常见的嵌入式系统应用实例

本章主要介绍嵌入式系统的基础知识，从它的定义、特点、应用领域、组成结构、开发的基本流程、发展趋势等方面系统地进行介绍，使读者能够建立起嵌入式系统的整体概念。

1.1 嵌入式系统

嵌入式系统的应用日益广泛，即使不可见，嵌入式系统也无处不在。随着信息化、智能化、网络化的发展，嵌入式系统技术也将获得广阔的发展空间。美国著名未来学家尼葛洛庞帝 1999 年 1 月访华时曾预言，4~5 年后嵌入式智能(计算机)工具将是继 PC(个人计算机)和因特网之后最伟大的发明。

随着现代社会信息化进程的加快，嵌入式系统被广泛地应用于军事、家用、工业、商业、办公、医疗等社会各个方面，表现出很强的投资价值。从国际范围来看，作为数字化电子信息产品核心的嵌入式系统，目前其硬件和软件开发工具市场已经突破 2000 亿美元，嵌入式系统带来的全球工业年产值更是达到了 10000 亿美元，随着全球经济的持续增长以及信息化的加速发展，嵌入式系统市场必将进一步增长。

1.1.1 嵌入式系统定义

嵌入式系统本身是一个相对模糊的定义，它的应用发源于微型计算机，发展于单片机。那么，嵌入式系统的定义是怎样的呢？以下是几种嵌入式系统的定义。

1. IEEE(美国电气和电子工程师协会)的定义

嵌入式系统是“用来控制、监视或者辅助操作机器、装置、工厂等大规模系统的设备”(原文为 devices used to control, monitor, or assist the operation of equipment, machinery or plants)。

这主要是从应用上加以定义的,从中可以看出嵌入式系统是软件和硬件的综合体,还可以涵盖机械等附属装置。

2. 我国微机学会的定义

嵌入式系统是以嵌入式应用为目的的计算机系统。并且还将其分为系统级、板级和片级。系统级包括各类工控设备、PC104 模块等;板级包括各类中央处理器(Central Processing Unit, CPU)主板和贴牌生产(Original Equipment Manufacture, OEM)产品;片级包括各种以单片机、数字信号处理(Digital Signal Processing, DSP)、微处理器为核心的设备。

3. 从系统角度的定义

嵌入式系统是设计完成复杂功能的硬件和软件,并使其紧密耦合在一起的计算机系统。

4. 在中国嵌入式系统领域比较认同的定义

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。

这是从技术的角度来定义的,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的应用程序等 4 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。

“嵌入式”反映了这些系统通常是更大系统中的一个完整部分,嵌入的系统中共存多个嵌入式系统。

1.1.2 嵌入式系统特点

嵌入式系统是针对具体应用设计的“专用计算机系统”,硬件和软件都必须高效率地设计,它具有以下显著特点:

(1) 系统内核小。由于嵌入式系统一般是应用于小型电子装置的,系统资源相对有限,所以其内核较之传统的操作系统要小得多。比如 $\mu\text{C}/\text{OS-II}$ 的内核最小可裁剪到 2KB。

(2) 专用性强。嵌入式系统的个性化很强,其中的软件和硬件的结合非常紧密,一般要针对硬件进行系统的移植,即使在同一品牌、同一系列的产品中也需要根据系统硬件的变化和增减不断进行修改。同时针对不同的任务,往往需要对系统进行较大更改,程序的编译下载要和系统相结合,这种修改和通用软件的“升级”完全是两个概念。

(3) 系统精简。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,同时也利于实现系统安全。

(4) 软件固化。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 CPU 本身中,而不是存储于磁盘等载体中。软件固化是嵌入式软件的基本要求,同时要求软件代码具有高质量、高可靠性和高实时性。



(5) 嵌入式软件开发要想走向标准化,就必须使用多任务的操作系统。嵌入式系统的应用程序可以没有操作系统直接在芯片上运行;但是为了合理地调度多任务,利用系统资源、系统函数以及和专家库函数接口,用户必须自行选配实时操作系统(Real Time Operating System, RTOS)开发平台,这样才能保证程序执行的实时性、可靠性,并减少开发时间,保障软件质量。

(6) 嵌入式系统开发需要开发工具和环境。由于其本身不具备自主开发能力,即使设计完成以后用户通常也是不能对其中的程序功能进行修改的,必须有一套开发工具和开发环境才能进行开发,这些工具和环境一般是基于通用计算机上的软硬件设备以及各种逻辑分析仪、示波器等。开发时往往有主机和目标机的概念,主机用于程序的开发,目标机作为最后的执行机,开发时需要交替结合进行。

1.1.3 嵌入式系统应用

嵌入式系统具有体积小、性能好、功耗低、可靠性高以及面向行业应用的突出特征,目前广泛应用于各种领域。

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,网络化是提高生产效率和产品质量、减少人力资源的主要途径,如工业过程控制、数控机床、电力系统、电网安全、电网设备监测、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,在未来几年内必将获得长足的发展。

2. 交通管理

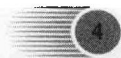
在车辆导航、流量控制、信息监测与汽车服务方面,嵌入式系统技术已经获得了广泛的应用,内嵌 GPS(Global Position System, 全球定位系统)模块, GSM(Global System for Mobile Communication, 全球移动通信系统)模块的移动定位终端已经在各种运输行业获得了成功的使用。目前 GPS 设备已经从尖端产品进入了普通百姓的家庭,只需要几千元,就可以随时随地找到自己的位置。

3. 信息家电

这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。即使不在家里,也可以通过电话线、网络进行远程控制。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. 销售点(Point Of Sales, POS)网络及电子商务

公共交通非接触式智能卡(Contactless Smart Card, CSC)发行系统, 公共电话卡发行系统, 自动售货机, 各种智能异步传送模式(Asynchronous Transfer Mode, 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

水文资料实时监测, 防洪体系及水土质量监测、堤坝安全, 地震监测网, 实时气象信息网, 水源和空气污染监测。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。

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。

在这些应用中, 可以着重于在控制方面的应用。就远程家电控制而言, 除了开发出支持(传输控制协议/网际协议)的嵌入式系统之外, 家电产品控制协议也需要制定和统一, 这需要家电生产厂家共同完成。同样的道理, 所有基于网络的远程控制器件都需要与嵌入式系统之间实现接口, 然后再由嵌入式系统来控制并通过网络实现控制。

1.2 嵌入式系统组成结构

嵌入式系统是专用的计算机应用系统, 图 1.2 所示完整地描述了嵌入式系统软硬件各部分的组成结构。嵌入式系统的硬件是嵌入式系统软件环境运行的基础, 它提供了嵌入式系统软件运行的物理平台和通信接口; 嵌入式操作系统和嵌入式应用软件则是整个系统的控制核心, 控制整个系统的运行, 提供人机交互的信息等。

整个嵌入式系统从大的方面可分为嵌入式系统硬件平台和嵌入式系统软件结构两大部分, 其中系统软件结构部分具体又分为若干层次。下面将围绕嵌入式系统的这两大组成部分进行详细介绍。

1.2.1 嵌入式系统硬件平台

硬件平台包含嵌入式处理器、存储器(SDRAM、ROM、Flash 等)、通用设备接口和 I/O 接口(A/D 转换器、D/A 转换器、I/O 接口控制卡等)。在一片嵌入式处理器的基础上添加电源电路、时钟电路和存储器电路, 就构成了一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。

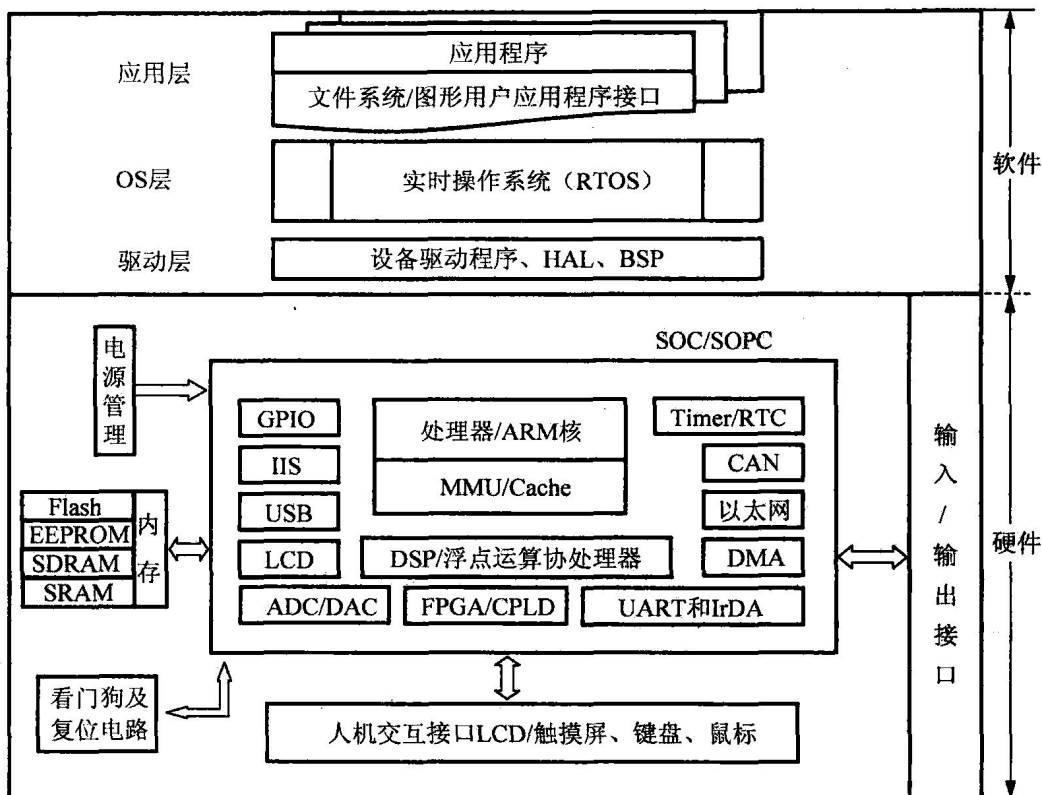


图 1.2 典型嵌入式系统组成

1. 嵌入式处理器

嵌入式系统核心是嵌入式处理器，它可分为以下几大类：

1) 嵌入式微处理器

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的体积和功耗。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是一样的，但在工作温度、抗电磁干扰、可靠性等方面一般都做了各种增强。

2) 嵌入式微控制器

嵌入式微控制器又称单片机，它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/计数器、Watchdog、I/O、串行口、脉宽调制输出、A/D、D/A、Flash、EEPROM 等各种必要功能和外设。为适应不同的应用需求，一般一个系列的单片机具有多种衍生产品，每种衍生产品的处理器内核都是一样的，不同的是存储器和外设的配置及封装。这样可以使单片机最大限度地和应用需求相匹配，功能不多不少，从而减少功耗和成本。

和嵌入式微处理器相比，嵌入式微控制器的最大特点是单片化，体积大大减小，从而使功耗和成本下降、可靠性提高。微控制器是目前嵌入式系统工业的主流。微控制器的片上外设资源一般比较丰富，适合于控制，因此称为微控制器。

3) 嵌入式 DSP 处理器

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,编译效率较高,指令执行速度也较高。在数字滤波、FFT(快速傅里叶变换)、谱分析等方面 DSP 算法正在大量进入嵌入式领域,DSP 应用正从在通用单片机中以普通指令实现 DSP 功能,过渡到采用嵌入式 DSP 处理器。

4) 嵌入式片上系统 SoC

随着电子设计自动化(Electronic Design Automation, EDA)的推广和超大规模集成电路(VLSI)设计的普及化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经来临,这就是片上系统(System on Chip, SoC)。各种通用处理器内核将作为 SoC 设计公司的标准库,和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,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去,应用系统电路板将变得很简洁,对于减小体积和功耗、提高可靠性非常有利。

2. 存储器

嵌入式系统需要存储器来存放和执行代码。嵌入式系统的存储器包含 Cache(高速缓冲存储器)、主存和辅助存储器等。

1) Cache

Cache 是一种容量小、速度快的存储器阵列,它位于主存和嵌入式微处理器内核之间,存放的是最近一段时间内微处理器使用最多的程序代码和数据。在需要进行数据读取操作时,微处理器尽可能地从 Cache 中读取数据,而不是从主存中读取,这样就大大改善了系统的性能,提高了微处理器和主存之间的数据传输速率。Cache 的主要目标就是:减小存储器(如主存和辅助存储器)给微处理器内核造成的存储器访问瓶颈,使处理速度更快,实时性更强。

在嵌入式系统中 Cache 全部集成在嵌入式微处理器内,可分为数据 Cache、指令 Cache 或混合 Cache,Cache 的大小依据不同处理器而定。一般中高档的嵌入式微处理器才会把 Cache 集成进去。

2) 主存

主存是嵌入式微处理器能直接访问的寄存器,用来存放系统和用户的程序及数据。主存可以位于微处理器的内部或外部,其容量为 256KB~1GB,根据具体的应用而定,一般片内存储器容量小、速度快,片外存储器容量大。常作为主存的存储器有:

ROM 类: NOR Flash、EPROM 和 PROM 等。

RAM 类: SRAM、DRAM 和 SDRAM 等。



知识要点提醒

NOR Flash 凭借其可擦写次数多、存储速度快、存储容量大、价格便宜等优点,在嵌入式领域内得到了广泛应用。



3) 辅助存储器

辅助存储器用来存放大数据量的程序代码或信息，它的容量大，但读取速度与主存相比就慢得很多，用来长期保存用户的信息。嵌入式系统中常用的辅助存储器有：硬盘、CF卡、NAND Flash、MMC 和 SD 卡等。

3. 嵌入式处理器的选择

嵌入式系统的功能软件集成于硬件系统之中，硬件与系统的应用软件一体化。在嵌入式系统的硬件设备中，嵌入式处理器是整个系统的核心部件，其性能的好坏决定着整个系统的运行效果，所以选择好处理器对于整个工作是至关重要的。

选择处理器有两个前提条件，分别说明如下：

(1) 深入研究具体的嵌入式系统应用需求，分析后获取应用系统的性能指标。具体的应用需求决定着嵌入式处理器的性能选型。应用需求分析后，需要定义产品具备的基本功能和性能指标，如系统处理的数据量大小和处理实时性要求、系统正常运行时的工作环境、系统运行过程中可能遇到的突发事件、系统的尺寸大小和功耗指标等。

(2) 分析研究市场上各大厂商提供的各款嵌入式处理器的性能指标，如功耗、体积、成本、可靠性、速度、处理能力、电磁兼容性等。这些参数也是各个半导体芯片厂商之间竞争的热点。

开发人员通过应用需求分析获取了产品的功能性和非功能性指标，研究了市场上嵌入式处理器的性能参数后，能够对嵌入式处理器做出较好的选择。嵌入式处理器选择的基本原则是满足具体功能性和非功能性指标需求的、市场应用反应良好的、硬件配置最少的嵌入式处理器。如简单的智能仪器仪表设计考虑使用 8 位单片机 8051；和数字信号处理相关的选用 DSP 芯片；如果产品侧重于通信功能，可考虑选用 Motorola 公司的嵌入式处理器 68K 系列；如果产品功能比较齐全，可选用 ARM 芯片等。

除了产品性能需求因素之外，选择嵌入式处理器时也需要考虑开发人员对此系列处理器的熟悉程度。因为不同公司开发的嵌入式处理器差别较大，从一类处理器转移到另一类处理器的难度比同一系列不同处理器间的转移难度大得多，风险高得多。

1.2.2 嵌入式系统软件结构

对于包含有操作系统的嵌入式系统来讲，嵌入式系统软件结构包含 4 个层次：设备驱动层、实时操作系统 RTOS 层、应用程序接口 API 层、应用程序层。

1. 设备驱动层

设备驱动层为上层软件提供了设备的操作接口。上层软件不用理会设备的具体内部操作，只需调用设备驱动层程序提供的接口即可。驱动层程序一般包括硬件抽象层 HAL、板级支持包 BSP 和设备驱动程序。

1) 硬件抽象层

硬件抽象层 HAL(Hardware Abstraction Layer)是位于操作系统内核与硬件电路之间的接口层，其目的在于将硬件抽象化。这样就使系统的设备驱动程序与硬件设备无关，从而

大大地提高了系统的可移植性。抽象层一般应包含相关硬件的初始化、数据的输入/输出操作、硬件设备的配置操作等功能。

2) 板级支持包

板级支持包 BSP(Board Support Package)是介于主板硬件和操作系统中驱动层程序之间的一层,一般认为它属于操作系统的一部分,主要是实现对操作系统的支持,为上层的驱动程序提供访问硬件设备寄存器的函数包,使之能够更好地运行于硬件主板。

板级支持包实现的功能有以下两个方面:

- ① 系统启动时,完成对硬件的初始化;
- ② 为驱动程序提供访问硬件的手段。

3) 设备驱动程序

设备驱动程序为上层软件提供设备的操作接口,上层软件只需调用设备驱动程序提供的接口,而不用理会设备的具体内部操作。设备驱动程序的好坏直接影响着系统的性能。设备驱动程序不仅要实现设备的基本功能函数,如初始化、中断响应、发送、接收等,使设备的基本功能能够实现,而且因为设备在使用过程中还会出现各种各样的差错,所以好的设备驱动程序还应该具备完善的错误处理函数。

2. 实时操作系统 RTOS 层

对于使用操作系统的嵌入式系统而言,其操作系统一般以内核映像的形式下载到目标系统中。RTOS 是一段在嵌入式系统启动后首先执行的背景程序,用户的应用程序是运行于 RTOS 之上的各个任务,RTOS 根据各个任务的要求,进行资源(包括存储器、外设等)管理、消息管理、任务调度及异常处理等工作。在 RTOS 支持的系统中,每个任务均有一个优先级,RTOS 根据各个任务的优先级,动态地切换各个任务,保证对实时性的要求。

3. 应用程序接口 API 层

应用程序接口(Application Programming Interface, API)是一系列复杂的函数、消息和结构的集合体。在操作系统中提供标准的 API 函数,可加快用户应用程序的开发,统一应用程序的开发标准,也为操作系统版本的升级带来了方便。在 API 函数中,提供了大量的常用模块,可大大简化用户应用程序的编写。

4. 应用程序层

实际的嵌入式系统应用软件建立在系统的主任务(Main Task)基础之上。用户应用程序主要通过调用系统的 API 函数对系统进行操作,完成用户应用程序开发。在用户的应用程序中,也可以创建用户自己的任务。任务之间的协调主要依赖于系统的消息队列。

1.3 嵌入式系统开发基本流程

当前,嵌入式开发已经逐步规范化,在遵循一般工程开发流程的基础上,嵌入式开发有其自身的一些特点,图 1.3 所示为嵌入式系统开发的一般流程。主要包括:系统定义与