

“十一五”高等院校规划教材

# ARM Cortex

# 微控制器教程

马忠梅 徐 琰 叶青林 编著



北京航空航天大学出版社

TP332.3  
M245

“十一五”高等

# ARM Cortex 微控制器教程

马忠梅 徐 琰 叶青林 编著

TP332.3

M245

## 内 容 简 介

本书是 32 位微控制器教材,基于 ARM Cortex 讲述嵌入式系统概念、软硬件组成、开发过程以及 Cortex 体系结构和应用程序开发设计方法。全书共 8 章,有嵌入式系统基础和 ARM Cortex - M 体系结构等理论内容,也有 TI、ST、NXP 和 Atmel 四家公司的 Cortex 微控制器时钟控制和应用程序设计等实践内容,另外还包括  $\mu\text{C}/\text{OS}$  移植和 UML 设计方法。本书有两个主要目的,一是普及高端 MCU,不要再面向寄存器编程,而要使用库函数;二是体现 Cortex MCU 很好的“生态环境”,有多家厂商支持。

本教材的特点是取材于最新资料,总结实际竞赛经验,实例较多,实用性较强。本书适用于没有操作系统知识的单片机开发人员学习嵌入式系统,可作为研究生和本科生嵌入式系统课程的教材使用。

### 图书在版编目(CIP)数据

ARM Cortex 微控制器教程/马忠梅,徐琰,叶青林编  
著. —北京:北京航空航天大学出版社,2010.1

ISBN 978 - 7 - 81124 - 945 - 3

I. A… II. ①马…②徐…③叶… III. 微处理器—教材  
IV. TP332.3

中国版本图书馆 CIP 数据核字(2009)第 193083 号

### ARM Cortex 微控制器教程

马忠梅 徐 琰 叶青林 编著

责任编辑 董云凤 张金伟 张 淳

\*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(100191) 发行部电话:(010)82317024 传真:(010)82328026

<http://www.buaapress.com.cn> E-mail:bhpress@263.net

北京市媛明印刷厂印装 各地书店经销

\*

开本:787×960 1/16 印张:26 字数:582 千字

2010 年 1 月第 1 版 2010 年 1 月第 1 次印刷 印数:4 000 册

ISBN 978 - 7 - 81124 - 945 - 3 定价:38.00 元



# 前 言

笔者从1985年初接触单片机到现在已25年,亲历了我国嵌入式系统的发展,同时也进行了嵌入式系统教学的探索。笔者1997年出版的《单片机的C语言应用程序设计》是国内第1本单片机C语言图书。该书推广单片机编程,以C语言为主,汇编为辅,采用模块化的编程方法。用C语言编程,程序可读性强、可重用性高,提高了编程的效率。为了程序的可移植性更强,上操作系统是进一步发展的必然趋势,但8/16位单片机上操作系统有其局限性。正是由于想上操作系统,才关注到了ARM。从计算机体系结构发展看,精简指令集计算机(RISC)在低功耗、流水线执行方面更具优势。ARM公司由于手机的火爆,已成为32位RISC处理器的事实标准。

笔者自2002年出版国内第1本ARM图书《ARM嵌入式处理器结构与应用基础》后,一直致力于ARM处理器的应用推广工作,编写了《ARM&Linux嵌入式系统教程》,在本科教学中使用。Linux的教学难度很大,一般要求学生学过操作系统。最早的嵌入式系统课在教学生怎么用Linux上耗费了很大精力,直到计算机操作系统课加了Linux上机实验,情况才好转。但有限的理论课时和实验学时,软硬件很难兼顾。研究生教学一直就没有放弃单片机。从研究生教学调查看,还是微控制器(俗称单片机,简称MCU)用得更多。嵌入式系统应用以微控制器为主。微控制器用量大,应用面广,已有很好的应用基础。一个MCU设计可以改造一个旧产品,也可以创造一个新产品。MCU对于本科生和研究生容易上手,更重要的是它集成度高,能够便于学生学做电路板、学习硬件接口技术和学习直接面向硬件的软件编程技术,这样才能对嵌入式软硬件有深入的理解。

ARM公司推出的Cortex-M核专门针对微控制器市场,并收购了德国的Keil公司,推出中国版的RealView MDK开发工具。这样原用8051单片机的用户可以将熟悉的开发环境过渡到ARM MCU的应用。2008年,ARM公司“微控制器市场创新研讨会”的新理念是就支离破碎的微控制器市场,以ARM来统一微控制器市场开发环境,以后微控制器的选型不再以8位、16位和32位来划分,而

是以高、中、低档芯片来划分。因为这些微控制器都有统一的机芯,区别只是它们的外围部件不同和厂商的技术支持不同。这样,工具和软件都可以重复使用,不用购置各种开发工具。国际著名的生产单片机的半导体公司 TI、ST、NXP、Atmel 等都得到 Cortex 授权,有望在 32 位嵌入式处理器市场统一机芯,构成很好的“生态环境”。ARM 微控制器比 8/16 位单片机速度快,适合运算量大的应用,用此进行产品设计更有竞争优势。原 ARM 微控制器和 Cortex-M 微控制器差别较大,后者有 CMSIS 编程标准,有一致的存储器映射,便于软件移植。

本书是 32 位微控制器教材,包括四家 MCU,主要涉及 GPIO、中断和定时器的片上资源基本编程,力图做类似的样例,比较各家的编程方式。主要目的有两个,一是普及高端 MCU,不要再面向寄存器编程,而要使用库函数;二是体现 Cortex MCU 很好的“生态环境”,有多家厂商支持。还有一点是尽量给出寄存器名、指令等的英文原义,我们发现这是国内翻译资料没有注意到的。对于 LM3S,我们带过竞赛,接口以它为主。通过竞赛看到,优秀的大三学生就能上手使用 Cortex 微控制器。这种 32 位 MCU 完全可用 C 语言编程,不涉及汇编。片上外设的寄存器配置更为复杂,支持的工作方式更多。厂家都纷纷提供了库函数,只要用户依据参数配置外设,使用库函数就能方便地操作外设,再没必要记住寄存器名、寄存器位定义和使用汇编了。这也是学生原来感觉学习单片机难度最大的地方。现编程只需关注外设的工作方式和可有哪些操作即可,大大降低了学习门槛。只要会 C 语言,许多样例程序就可直接使用。

由采用 8/16 位单片机到采用 32 位嵌入式处理器的嵌入式系统,一方面是由于互联网的普及和高端应用的需求,另一方面是由于 8 位单片机在半导体公司的利润空间已经很小了。由全国大学生电子设计竞赛组委会主办,Intel 公司协办的“全国大学生电子设计竞赛——嵌入式系统专题竞赛”进一步丰富了全国大学生电子设计竞赛的形式和内容,推动了高校信息电子类专业的教学改革、课程体系及实验室建设,各高校纷纷开设了嵌入式系统课程。本书前 3 章即为开课后的讲稿整理而成。全书共 8 章,各章节内容安排如下:

第 1 章为嵌入式系统的基础知识,讲述嵌入式系统概念、嵌入式系统应用、嵌入式处理器、嵌入式操作系统和实时操作系统内核。

第 2 章介绍嵌入式系统开发的特点、开发流程、调试方法和板级支持包的功能。

第 3 章详细介绍 ARM Cortex 体系结构和编程模型,分类给出 ARM 指令集的说明及汇编语言程序设计,特别讲述存储器映射及中断和异常的处理方法,最

后介绍 Cortex 的优势。

第 4 章给出 TI、ST、NXP 和 Atmel 四家公司的 Cortex 微控制器分类、特性、结构和时钟控制等。

第 5 章给出 TI、ST、NXP 和 Atmel 四家公司的 Cortex 微控制器应用程序的编写方法,侧重于 GPIO、中断和定时器的片上资源基本编程。

第 6 章详细给出了 LCD 显示模块、行列式键盘、电机控制、串行总线接口 SPI 和 I<sup>2</sup>C 接口及编程实例。

第 7 章为将  $\mu\text{C}/\text{OS} - \text{II}$  移植到 Cortex 处理器上需要修改的内容。

第 8 章简要介绍 UML 设计方法中常见的模型元素、图和视图。

本书是由北京理工大学教师带领学生共同完成的。笔者负责全书内容的统稿,许多内容取材于学生的例程开发和竞赛设计。徐琰、叶青林、卢彦超、金伟、陈航、宋亚斐、施度、谢华勇同学参与了开发设计实践。下决心写此书是由于国内缺少合适的高端 32 位微控制器教材,全校研究生的课程又不好局限于某一家公司的芯片。我们只希望培养学生对嵌入式系统的兴趣,更多地让学生比较与选型,只有共同把 Cortex 微控制器应用推动起来,国内高端嵌入式应用才能健康发展。希望不久的将来,在国内能看到大量基于 Cortex 微控制器的产品,学生们能更好找到工作。

感谢 ARM 中国公司前总裁谭军,是他的鼓励和支持才使我们下决心写此书。感谢周立功提供 EasyARM 开发板,周立功公司李佰华、周小明提供 NXP 公司的 LPC1700 系列和 TI 公司 LM3S 系列翻译资料;感谢 ST 公司梁平、张军辉提供 STM32 开发板并允许使用翻译资料;感谢 NXP 公司王朋朋提供 LPC1700 系列开发板,并对样例程序提出宝贵建议;感谢 IAR 公司叶涛提供 J-LINK 仿真器;感谢英蓓特公司廖武提供 U-LINK 仿真器;感谢 Atmel 公司庞长富抽奖出的 SAM3U 开发板,应了笔者和 Atmel 公司的缘分,编写的第一本 ARM 芯片的书就是《AT91 ARM 核微控制器结构与开发》;感谢武汉理工大学李宁提供了一部分 SAM3U 翻译资料。希望借此书感谢帮助过我们的人,在此谨向他们深表谢意。

马忠梅

2009 年 9 月

# 目 录

## 第 1 章 嵌入式系统基础

1.1 嵌入式系统概念 .....	1
1.1.1 嵌入式系统定义 .....	1
1.1.2 嵌入式系统组成 .....	2
1.1.3 嵌入式系统特点 .....	3
1.1.4 嵌入式系统应用 .....	5
1.1.5 实时系统 .....	6
1.2 嵌入式处理器 .....	7
1.2.1 嵌入式处理器分类 .....	7
1.2.2 微控制器 .....	8
1.2.3 嵌入式微处理器 .....	9
1.2.4 DSP 处理器 .....	9
1.2.5 片上系统 .....	10
1.2.6 典型的嵌入式处理器 .....	11
1.3 嵌入式操作系统 .....	12
1.3.1 操作系统概念和分类 .....	12
1.3.2 实时操作系统 .....	16
1.3.3 常见的嵌入式操作系统 .....	19
1.4 实时操作系统的内核 .....	23
1.4.1 任务管理 .....	23
1.4.2 任务间的通信和同步 .....	30
1.4.3 存储器管理 .....	32
1.4.4 定时器和中断管理 .....	33
习 题 .....	34

**第 2 章 嵌入式系统开发过程**

2.1 嵌入式软件开发的特点 .....	35
2.2 嵌入式软件的开发流程 .....	36
2.3 嵌入式系统调试 .....	42
2.4 板级支持包 .....	46
习 题 .....	48

**第 3 章 Cortex - M 体系结构**

3.1 ARM 体系结构概述 .....	49
3.1.1 ARM 体系结构的特点 .....	49
3.1.2 流水线 .....	52
3.1.3 ARM 处理器核 .....	54
3.1.4 结构框图 .....	64
3.1.5 典型的连接方式 .....	65
3.1.6 ARM JTAG 调试接口 .....	67
3.2 编程模型 .....	68
3.2.1 Thumb - 2 指令集体系结构 (ISA) .....	68
3.2.2 寄存器 .....	69
3.2.3 工作模式和特权级别 .....	72
3.2.4 Cortex - M3 堆栈 .....	73
3.2.5 数据类型 .....	74
3.2.6 存储器和存储器映射 I/O .....	74
3.3 ARM 基本寻址方式 .....	77
3.4 Thumb - 2 指令集说明 .....	82
3.4.1 条件执行 .....	82
3.4.2 指令分类说明 .....	83
3.4.3 Thumb - 2 指令集的特点 .....	103
3.4.4 ARM 汇编语言程序设计 .....	108
3.5 存储器映射 .....	111
3.5.1 存储系统简介 .....	111
3.5.2 存储器映射空间 .....	111
3.5.3 位绑定操作 .....	114
3.6 中断和异常 .....	117
3.6.1 异常类型 .....	118



3.6.2	优先级的定义 .....	119
3.6.3	向量表 .....	121
3.6.4	中断输入及挂起行为 .....	122
3.6.5	NVIC 与中断控制 .....	124
3.6.6	中断/异常的响应序列 .....	126
3.6.7	尾链中断 .....	127
3.6.8	迟到异常处理 .....	128
3.6.9	异常返回值 .....	128
3.6.10	中断延迟 .....	130
3.7	ARM Cortex - M 的优势 .....	130
	习 题 .....	132
 <b>第 4 章 ARM Cortex - M 微控制器</b>		
4.1	LM3S 微控制器 .....	134
4.1.1	LM3S 微控制器分类 .....	134
4.1.2	LM3S 微控制器特性和结构 .....	135
4.1.3	系统控制和时钟 .....	144
4.1.4	Stellaris 外设驱动库 .....	150
4.2	STM32 微控制器 .....	155
4.2.1	STM32 微控制器分类 .....	155
4.2.2	STM32F103 微控制器特性和结构 .....	156
4.2.3	STM32 复位与时钟控制 .....	163
4.2.4	STM32 启动模式 .....	168
4.2.5	STM32 固件库 .....	169
4.3	LPC1000 系列微控制器 .....	174
4.3.1	LPC1000 系列微控制器分类 .....	174
4.3.2	LPC1700 微控制器特性和结构 .....	175
4.3.3	LPC1700 系统控制 .....	179
4.3.4	LPC1700 时钟和功耗控制 .....	181
4.4	SAM3U 微控制器 .....	187
4.4.1	SAM3U 微控制器系列 .....	187
4.4.2	SAM3U 微控制器特性和结构 .....	188
4.4.3	SAM3U 复位与时钟控制 .....	191
	习 题 .....	198

## 第 5 章 片上资源的编程技术

5.1	LM3S 微控制器片上资源 .....	199
5.1.1	LM3S 通用输入/输出端口 GPIO .....	199
5.1.2	LM3S 按键中断 .....	208
5.1.3	LM3S 通用定时器 .....	210
5.1.4	LM3S 串行口 UART .....	221
5.1.5	LM3S 模/数转换器(ADC) .....	226
5.2	STM32 微控制器片上资源 .....	228
5.2.1	STM32 通用和复用功能 I/O(GPIO 和 AFIO) .....	228
5.2.2	STM32 按键中断 .....	234
5.2.3	STM32 通用定时器 .....	240
5.3	LPC1700 微控制器片上资源 .....	267
5.3.1	LPC1700 通用输入输出 GPIO .....	267
5.3.2	LPC1700 按键中断 .....	276
5.3.3	LPC1700 通用定时器 .....	284
5.4	SAM3U 微控制器片上资源 .....	298
5.4.1	SAM3U 通用输入输出 GPIO .....	298
5.4.2	SAM3U 按键中断 .....	304
5.4.3	SAM3U 通用定时器 .....	307
	习 题 .....	321

## 第 6 章 嵌入式系统接口及编程

6.1	字符型 LCD 显示模块 .....	322
6.1.1	LCD 硬件接口设计 .....	322
6.1.2	LCD 显示编程 .....	323
6.2	行列式键盘 .....	326
6.2.1	键盘硬件接口设计 .....	326
6.2.2	键盘扫描程序设计 .....	327
6.3	电机控制 .....	332
6.3.1	步进电机控制模块 .....	332
6.3.2	直流电机控制模块 .....	339
6.4	串行总线接口 SPI .....	343
6.4.1	串行总线接口 SPI 简介 .....	343

6.4.2	Stellaris 的同步串行接口 (SSI) .....	343
6.4.3	与串行总线接口 SPI 有关的函数 .....	346
6.4.4	ZigBee 通信实例 .....	347
6.5	I <sup>2</sup> C 接口 .....	360
6.5.1	I <sup>2</sup> C 总线简介 .....	360
6.5.2	I <sup>2</sup> C 总线的通信规约 .....	360
6.5.3	Stellaris 的 I <sup>2</sup> C 接口 .....	362
6.5.4	与 I <sup>2</sup> C 有关的库函数 .....	364
6.5.5	I <sup>2</sup> C 接口应用实例 .....	365
	习 题 .....	370
<b>第 7 章 <math>\mu</math>C/OS - II 移植</b>		
7.1	移植代码说明 .....	371
7.1.1	includes. h .....	372
7.1.2	OS_CPU. H 文件 .....	372
7.1.3	OS_CPU_C. C 文件 .....	374
7.1.4	OS_CPU_A. ASM .....	375
7.2	异常/中断 .....	381
7.3	启动文件和目标板配置文件 .....	381
7.3.1	startup_ewarm. c 文件 .....	382
7.3.2	Target. H 文件 .....	384
7.3.3	Target. C 文件 .....	387
7.4	用户文件 .....	389
<b>第 8 章 UML 设计方法</b>		
8.1	常见的模型元素 .....	391
8.1.1	结构要素 .....	391
8.1.2	行为要素 .....	393
8.1.3	分组要素 .....	393
8.1.4	注释要素 .....	394
8.1.5	关 系 .....	394
8.2	图 .....	395
8.3	视 图 .....	400
	参考文献 .....	402

# 第 1 章

## 嵌入式系统基础

### 1.1 嵌入式系统概念

#### 1.1.1 嵌入式系统定义

所谓嵌入式系统(Embedded Systems),实际上是“嵌入式计算机系统”的简称,它是相对于通用计算机系统而言的。有些系统也有计算机,但是计算机是作为某个专用系统中的一个部件而存在的。像这样“嵌入”到更大、更专用的系统中的计算机系统,就称为“嵌入式计算机”、“嵌入式计算机系统”或“嵌入式系统”。

在日常生活中,早已存在许多嵌入式系统的应用,如人们天天必用的移动电话、戴在手腕上的电子表、烹调用的微波炉、办公室中的打印机、汽车中的供油喷射控制系统、防锁死刹车系统(ABS),以及现在流行的个人数字助理(PDA)、数码相机、数码摄像机等,它们内部都有一个中央处理器 CPU。

嵌入式系统无处不在,从家庭的洗衣机、电冰箱、小汽车,到办公室中的远程会议系统等,都属于可以使用嵌入式技术进行开发和改造的产品。嵌入式系统本身是一个相对模糊的定义。一个手持的 MP3 和一个 PC104 的微型工业控制计算机都可以认为是嵌入式系统。

根据电气工程师协会(IEE)的定义:嵌入式系统是用来控制或监视机器、装置或工厂等的大规模系统的设备(devices used to control, monitor, or assist the operation of equipment, machinery or plants)。可以看出此定义是从应用上考虑的,嵌入式系统是软件和硬件的综合体,还可以涵盖机电等附属装置。

国内一般定义如下: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了对功能、可靠性、成本、体积、功耗等有严格要求的专用计算机系统。

嵌入式系统的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就包含了 5~10 个嵌入式微处理器,键盘、硬盘、显示器、Modem、网卡、声卡、打印机、扫描仪、数码相机、集线器等均是由嵌入式处理器进行控制的。在制造工业、过程控制、通信、仪器、仪表、汽车、船舶、航空、航天、军事装备、消费类产品等方面,嵌入式系统都有用武之地。

## 第1章 嵌入式系统基础

美国汽车大王福特公司的高级经理曾宣称：“福特出售的‘计算能力’已超过了IBM。”由此可以想象嵌入式计算机工业的规模和广度。美国著名未来学家尼葛洛庞帝在1999年1月访华时曾预言“4~5年以后嵌入式智能计算机将是继PC和因特网之后最伟大的发明”。

### 1.1.2 嵌入式系统组成

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。

#### 1. 嵌入式处理器

嵌入式处理器是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,其大多工作在为特定用户群设计的系统中。它通常把通用计算机中许多由板卡完成的任务集成在芯片内部,从而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。

大的硬件厂商会推出自己的嵌入式处理器,因而现今市面上有1000多种嵌入式处理器芯片,其中使用最为广泛的有8051、PIC、AVR、ARM、MIPS、PowerPC、MC68000等。

#### 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用来完成存储、通信、调试、显示等辅助功能的其他部件。它根据外围设备的功能可分为以下3类。

- ▶ 存储器:静态易失性存储器(RAM、SRAM)、动态存储器(DRAM)、非易失性存储器(Flash)。其中,Flash以可擦写次数多、存储速度快、容量大及价格便宜等优点,在嵌入式领域得到广泛的应用。
- ▶ 接口:应用最为广泛的包括并口、RS-232串口、IrDA红外接口、SPI串行外围设备接口、I<sup>2</sup>C(Inter IC)总线接口、USB通用串行总线接口、Ethernet网口等。
- ▶ 人机交互:LED、LCD、键盘和触摸屏等人机交互设备。

#### 3. 嵌入式操作系统

在大型嵌入式应用系统中,为了使嵌入式开发更方便、快捷,就需要具备相应的管理存储器分配、中断处理、任务间通信和定时器响应,以及提供多任务处理等功能的稳定的、安全的软件模块集合,即嵌入式操作系统。嵌入式操作系统的引入大大地提高了嵌入式系统的功能,方便了应用软件的设计,但同时也占用了宝贵的嵌入式系统资源。一般在比较大型或需要多任务的应用场合才考虑使用嵌入式操作系统。

#### 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,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。用户的任务可能有时间和精度的要求,有些应用软件需要嵌入式操作系统的支持,但在简单的应用场合下不需要专门的操作系统。

由于嵌入式应用对成本十分敏感,因此为减少系统的成本,除了精简每个硬件单元的成本外,应尽可能地减少应用软件的资源消耗,尽可能地实现优化。应用软件是实现嵌入式系统功能的关键,对嵌入式系统软件和应用软件的要求也与通用计算机有所不同。嵌入式软件的特点如下:

- ▶ 软件要求固态化存储。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中。
- ▶ 软件代码要求高质量、高可靠性。半导体技术的发展使处理器速度不断提高,也使存储器容量不断增加。但在大多数应用中,存储空间仍然是宝贵的,还存在实时性的要求。为此,程序编写和编译工具的质量要高,以减少程序二进制代码长度,提高执行速度。
- ▶ 系统软件的高实时性是基本要求。在多任务嵌入式系统中,对重要性各不相同的任务进行统筹兼顾的合理调度是保证每个任务及时执行的关键,单纯通过提高处理器速度是无法完成和没有效率的。这种任务调度只能由优化编写的系统软件来完成,因此,系统软件的高实时性是基本要求。
- ▶ 多任务实时操作系统成为嵌入式应用软件的必需。随着嵌入式应用的深入和普及,接触到的实际应用环境也越来越复杂,嵌入式软件也越来越复杂。支持多任务的实时操作系统成为嵌入式软件必需的系统软件。

典型嵌入式系统的硬件和软件基本组成如图 1-1 和图 1-2 所示。

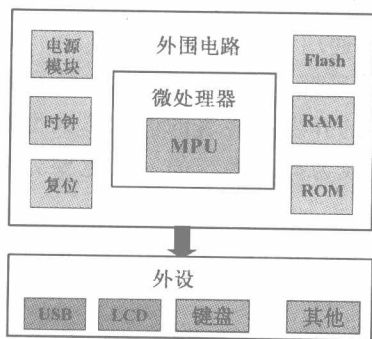


图 1-1 典型嵌入式系统基本组成——硬件

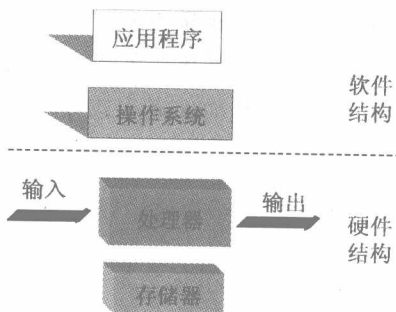


图 1-2 典型嵌入式系统基本组成——软件

### 1.1.3 嵌入式系统特点

由于嵌入式系统是应用于特定环境下、面对专业领域的应用系统,所以不同于通用计算机系统的多样化和适用性。它与通用计算机系统相比具有以下特点:

- ① 嵌入式系统通常是面向特定应用的,一般都有实时要求。嵌入式处理器大多工作在为

## 第1章 嵌入式系统基础

特定用户群所设计的系统中。它通常具有功耗低、体积小、集成度高、成本低等特点,从而使嵌入式系统的设计趋于小型化、专业化,也能使移动能力大大增强,与网络的耦合也越来越紧密。

② 嵌入式系统是将先进的计算机技术、半导体工艺、电子技术和通信网络技术与各领域的具体应用相结合的产物。这一特点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 嵌入式系统和具体应用有机地结合在一起,它的升级换代也和具体产品同步进行。因此,嵌入式系统产品一旦进入市场,一般具有较长的生命周期。

④ 嵌入式系统的硬件和软件都必须高效率地设计,在保证稳定、安全、可靠的基础上量体裁衣,去除冗余,力争在同样的硅片面积上实现更高的性能。这样,才能最大限度地降低应用成本。在具体应用中,对处理器的选择决定了市场竞争力。

⑤ 嵌入式系统常常还有降低功耗的要求。这一方面是为了省电,因为嵌入式系统往往以电池供电;另一方面是要减少发热量,因为嵌入式系统中常常没有风扇等排热设施。

⑥ 可靠性与稳定性对于嵌入式系统有着特别重要的意义,因此即使逻辑上的系统结构相同,其在物理组成上也会有所不同。同时,对使用的元器件包括接插件、电源等的质量和可靠性要求都比较高,因此元器件的平均无故障时间 MTBF(Mean Time Between Failure)成为关键性的参数。此外,允许的环境温度也是个需要重点考虑的问题。

⑦ 嵌入式系统提供的功能以及面对的应用和过程都是预知的、相对固定的,而不像通用计算机那样有很大的随意性。既然是专用的系统,在可编程方面就不需要那么灵活。一般也不会用嵌入式系统作为开发应用环境,在嵌入式系统上通常也不会运行一些大型的软件。一般而言,嵌入式系统对 CPU 计算能力的要求并不像通用计算机那么高。

⑧ 许多嵌入式系统都有实时要求,需要有对外部事件迅速作出反应的能力,特别是在操作系统中有所反映,从而使嵌入式软件的开发与常规软件的开发有显著的区别。典型的嵌入式实时操作系统与常规的操作系统的区别,并因而成为操作系统的一个重要分支和一个独特的研究方向。

⑨ 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,必须有一套交叉开发工具和环境才能进行开发。

⑩ 通用计算机的开发人员通常是计算机科学或者计算机工程方面的专业人士,而嵌入式系统开发人员却往往是各个应用领域中的专家,这就要求嵌入式系统所支持的开发工具易学、易用、可靠、高效。

嵌入式系统是面向用户、面向产品、面向应用的,必须与具体应用相结合才会具有生命力,才更具有优势。嵌入式系统是与应用紧密结合的,具有很强的专用性,必须结合实际系统需求进行合理的裁减利用。嵌入式系统必须根据应用需求对软硬件进行裁减,满足应用系统的功能、可靠性、成本、体积等要求。

### 1.1.4 嵌入式系统应用

嵌入式系统以应用为中心,强调体积和功能的可裁减性,是以完成控制、监视等功能为目标的专用系统。在嵌入式应用系统中,系统执行任务的软硬件都嵌入在实际的设备环境中,通过专门的 I/O 接口和外界交换信息。一般,它们执行的任务程序不由用户编制。

嵌入式系统主要用于各种信号处理与控制,目前已在国防、国民经济及社会生活各领域普及应用,用于企业、军队、办公室、实验室以及个人、家庭等各种场所。

① 军用。各种武器控制(火炮控制、导弹控制、智能炸弹制导引爆装置)、坦克、舰艇、轰炸机等陆海空各种军用电子装备,雷达、电子对抗军事通信装备,野战指挥作战用各种专用设备,在海湾战争到最近的伊拉克战争中广泛使用。我国嵌入式计算机最早用于导弹控制。

② 家用。我国各种信息家电产品,如数字电视机、机顶盒、数码相机、VCD、DVD 音响设备、可视电话、家庭网络设备、洗衣机、电冰箱、智能玩具等,广泛采用微控制器及嵌入式软件。EMIT(嵌入式 Internet 技术)已用于社区对家用电、水、煤气表的远程抄表。

③ 工业用。各种智能测量仪表、数控装置、可编程控制器、分布式控制系统、现场总线仪表及控制系统、工业机器人、机电一体化机械设备、汽车电子设备等,广泛采用微处理器、微控制器芯片级、标准总线的模板级及系统嵌入式计算机。

④ 商用。各类收款机、POS 系统、电子秤、条形码阅读机、商用终端、银行点钞机、IC 卡输入设备、取款机、自动柜员机、自动服务终端、防盗系统、各种银行专业外围设备。

⑤ 办公用。复印机、打印机、传真机、扫描仪、激光照排系统、安全监控设备、手机、寻呼机、个人数字助理(PDA)、变频空调设备、通信终端、程控交换机、网络设备、录音录像及电视会议设备、数字音频广播系统等。

⑥ 医用电子设备。各种医疗电子仪器,X 光机、超声诊断仪、计算机断层成像系统、心脏起搏器、监护仪、辅助诊断系统、专家系统等。

当今最热门的嵌入式系统有以下几种。

#### 1. 个人数字助理 PDA

PDA 具有网络、多媒体等强大的功能。康柏公司 iPAQ 掌上电脑一般都预装 Pocket PC 操作系统。现 PDA 手机,即智能手机成为新的热点。

#### 2. 机顶盒 STB

所谓的机顶盒 STB(Set Top Box),表面上理解只是放在电视机上的盒子,能提供通过电视机直接上 Internet 的功能。但它更吸引人的地方在于简单易用,专为那些不很了解计算机的人而设计。现今用户端机顶盒的趋势是朝微型电脑发展,即逐渐集成电视和电脑的功能,成为一个多功能服务的工作平台。用户透过此设备即能够实现交互式数字电视、数字电视广播、Internet 访问、远程教学、会议电视、电子商务等多媒体信息服务。



## 第1章 嵌入式系统基础

### 3. IP 电话

IP 电话(IP Phone)把电话网和 Internet 结合成一个功能强大的通信网络。IP 电话在 IP 网络上实时传输被压缩的语音信息,并达到实时传输的目的。IP 电话以数字形式作为传输媒体,占用资源少,因此成本很低,价格便宜,用户能够从中得到实惠。

嵌入式系统的应用正在从狭窄应用范围、单一应用对象以及简单功能,向着未来社会的应用需求进行转变。社会对嵌入式系统的需求正在慢慢扩大,特别是最近几年来国际互联网的发展,从 PC 时代步入到后 PC 时代,对信息家电的需求越来越明显。嵌入式系统在信息家电中的应用,则是对嵌入式系统概念和应用范围的一个变革,从而打破了过去 PC 时代被单一微处理器厂家和单一操作系统厂家垄断的旧局面,出现了一个由多芯片、多处理器占领市场领域的新局面。

### 1.1.5 实时系统

实时系统(real time system)是指产生系统输出的时间对系统是至关重要的系统。从输入到输出的滞后时间必须小到一个可以接受的时限内。因此,实时逻辑正确性不仅依赖于计算结果的正确性,还取决于输出结果的时间。

实时系统是一个能够在指定或者确定的时间内完成系统功能及对外部或内部事件在同步或异步时间内作出响应的系统。

实时系统是在逻辑和时序控制中,如果出现偏差将会引起严重后果的系统。对于实时系统来说,它应具备以下几个重要的特性。

① 实时性。实时系统所产生的结果在时间上有着严格的要求,只有符合时间要求的结果才认为是正确的。在实时系统中,每个任务都有一个截止期限,任务必须在这个截止期限之内完成,以此保证系统所产生的结果在时间上的正确性。

② 并行性。一般来说,一个实时系统常常有多个外部输入端口。因此,要求系统具有并行处理的能力,以便能同时响应来自不同端口的输入信号。

③ 多路性。实时系统的多路性表现在对多个不同的现场信息进行采集以及对多个对象和多个执行机构实行控制。

④ 独立性。每个用户向实时系统提出服务请求,相互间是独立的。在实时控制系统中,对信息的采集和对象控制也是相互独立的。

⑤ 可预测性。实时系统的实际行为必须处在一定的限度内,而这个限度可以由系统的定义而获得。这意味着系统对来自外部输入的反应必须全部是可预测的,即使在最坏的条件下,系统也要严格遵守时间的约束。因此,在出现过载时,系统必须能以一种可预测的方式来降级它的性能。

⑥ 可靠性。可靠性一方面指系统的正确性,即系统所产生的结果在返回值和运行费时上