

# ARM Cortex核 TI微控制器原理与应用

马忠梅 李 奇  
徐 琰 周红忠  
等编著



北京航空航天大学出版社  
BEIHANG UNIVERSITY PRESS

# ARM Cortex 核 TI 微控制器原理与应用

马忠梅 李 奇 等编著  
徐 琰 周红忠

北京航空航天大学出版社

## 内 容 简 介

基于 32 位 ARM Cortex 微控制器,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。全书共分 10 章,介绍嵌入式系统基础和 ARM Cortex-M 体系结构等理论内容, TI 公司的 Cortex 微控制器系统控制和片内资源、人机交互、总线接口、网络接口、电机控制应用程序设计等实践内容,以及物联网涉及的传感器和 RFID 采集程序、ZigBee 点对点通信和 32 位网络协调器节点设计内容。另外,附录中介绍  $\mu\text{C}/\text{OS}$  的移植方法。希望通过本书能普及高端 MCU 不要再面向寄存器编程,而要使用库函数,这得益于 Cortex MCU 很好的开发环境。

本书取材于最新资料,总结实际竞赛和科研经验,实例较多,且实用性强。

本书可作为本科生和研究生嵌入式系统类课程的教材,也适合于没有操作系统知识的单片机开发人员学习嵌入式系统。

### 图书在版编目(CIP)数据

ARM Cortex 核 TI 微控制器原理与应用 / 马忠梅等编著. —北京:北京航空航天大学出版社,2011.1  
ISBN 978-7-5124-0264-5

I. ①A… II. ①马… III. ①微控制器 IV. ①TP332.3

中国版本图书馆 CIP 数据核字(2010)第 222325 号

版权所有,侵权必究。

### ARM Cortex 核 TI 微控制器原理与应用

马忠梅 李 奇  
徐 琰 周红忠 等编著

责任编辑 王 实

\*

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(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱: [bhpress@263.net](mailto:bhpress@263.net) 邮购电话:(010)82316936

北京时代华都印刷有限公司印装 各地书店经销

\*

开本:787×1092 1/16 印张:22.5 字数:576 千字

2011 年 1 月第 1 版 2011 年 1 月第 1 次印刷 印数:4 000 册

ISBN 978-7-5124-0264-5 定价:38.00 元

# 前 言

笔者从1985年初接触单片机到现在已有25年,亲历我国嵌入式系统的发展,也从事着嵌入式教学的探索,并于1997年出版了《单片机的C语言应用程序设计》。该书是国内第一本单片机C语言教材,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。为了使程序的可移植性更强,上操作系统是进一步发展的需要,但8位/16位单片机上操作系统还是有局限性的。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来看,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由于手机的火爆,已成为32位RISC处理器的事实标准。

自2002年出版了国内第一本ARM图书——《ARM嵌入式处理器结构与应用基础》后,笔者一直致力于ARM处理器的应用推广工作,所编写的《ARM & Linux嵌入式系统教程》已在本科教学中使用。Linux的教学难度很大,一般要求学过操作系统。最早我们在嵌入式系统教学中耗费了很大精力,教学生怎么用Linux,直到计算机操作系统课程有了Linux上机实验,情况才有所好转。但有限的理论和实验课时,使得软硬件很难兼顾。笔者在研究生教学中一直没有放弃单片机,从研究生教学调查情况来看,还是微控制器MCU(俗称单片机)用得较多。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,也可以创新一个新产品。MCU对于大学生和研究生容易上手,更重要的是其集成度高,使学生能够方便地学做电路板,学习硬件接口技术和直接面向硬件的软件编程技术,从而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,并收购德国的Keil公司,推出中国版的RealView MDK开发工具。这样,原来8051单片机的用户可使用熟悉的开发环境过渡到ARM MCU。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”,区别的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 TI、ST、NXP 和 Atmel 等半导体公司都得到了 Cortex 的授权,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 位/16 位单片机速度快,适合运算量大的应用,用其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,方便软件移植。

本书是 32 位微控制器教材,包括 TI 公司的 Cortex 微控制器系统控制和片内资源、人机交互、总线接口、网络接口、电机控制和物联网等应用程序设计内容,力图多做些样例,以方便读者学习 32 位微控制器的编程方式。希望通过本书能普及高端 MCU 不要再面向寄存器编程,而要使用库函数,这得益于 Cortex MCU 很好的开发环境。另外,书中尽量给出寄存器名和指令等的英文原义,我们发现这是国内翻译资料没有注意到的。

我们曾指导过大学生的电子设计竞赛,通过参加竞赛可以看到,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,不涉及汇编语言。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,再没有必要记住寄存器名、寄存器位定义和使用汇编语言了。这也是学生原来学习单片机感觉难度最大的地方。现在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,许多样例程序可直接使用。

从应用 8 位/16 位单片机到采用 32 位嵌入式处理器,一方面是互联网的普及和高端应用的需求,另一方面是 8 位单片机对半导体公司来说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,推动了高等院校电子信息类专业的教学改革、课程体系及实验室建设,各高等院校纷纷开设嵌入式系统课程。本书前 3 章即为笔者开课后的讲稿整理而成。

全书共分 10 章及一个附录,内容安排如下:

第 1 章介绍嵌入式系统的基础知识,包括嵌入式系统概念、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 公司的 Stellaris 微控制器分类、特性、结构和系统控制,介绍最小系统设计、Stellaris 外设驱动库和开发工具。

第 5 章给出 TI 公司的 Stellaris 微控制器 GPIO、中断、定时器、脉宽调制器、串行口和模/数转换器等各种片内资源的基本编程。

第 6 章具体介绍行列式键盘、键盘显示接口芯片、字符型和点阵型 LCD 显示模块以及触摸屏的人机交互程序设计。

第 7 章涉及串行总线接口 SPI、I<sup>2</sup>C 接口、I<sup>2</sup>S 接口和 USB 接口的编程实例。

第 8 章介绍 CAN 总线和以太网的网络接口编程方法。

第 9 章介绍步进电机和直流电机控制模块,以及 TI 公司的电机控制方案。

第 10 章针对物联网技术涉及的温湿度、光照度、加速度传感器和 RFID,给出采集程序,还包括 ZigBee 点对点通信和 32 位网络协调器节点等设计内容。

附录 A 介绍将  $\mu$ C/OS-II 移植到 Cortex 处理器上需要修改的内容。

本书可作为本科生和研究生嵌入式系统类课程的教材,也适合没有操作系统知识的单片机开发人员学习嵌入式系统。

书中许多内容取材于学生的例程开发和竞赛设计。李奇、徐琰、叶青林、杨郑和、孙娟、王飞、卢彦超、金伟、陈航、宋亚斐、施度、谢华勇以及西藏自治区广播电视厅访问学者周红忠参与了开发设计实践。希望通过本书所列举的大量的程序样例培养学生对嵌入式系统的兴趣。只有共同把 Cortex 微控制器的应用推动起来,国内高端嵌入式应用才能进一步发展。希望不久的将来,国内能看到大量基于 Cortex 微控制器的产品,为学生的就业提供更多更好的机会。

下决心写此书是 TI 公司大学计划总监沈洁的诚意感动了我。她专程来京,因北京下大雨机场关闭又飞回上海,后又再次来京给我们带来 LM3S9B96 开发板。感谢 TI 公司赵宗睿、鲍震、王继涛提供技术资料和支持;感谢精仪达盛公司毕才术、胡芳提供 EXP-LM3S3749 实验箱和开发板,芮德华、陈佳提供资料和技术支持;感谢周立功提供 EasyARM1138 开发板,周立功公司李佰华提供翻译资料和程序样例;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 CodeSourcery 公司齐尧编写了 Sourcery G++ 开发工具简介。在此谨借此书向帮助过我们的人深表谢意。

马忠梅  
2010 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 .....	4
1.1.5 实时系统 .....	5
1.2 嵌入式处理器 .....	6
1.2.1 嵌入式处理器分类 .....	6
1.2.2 典型的嵌入式处理器 .....	8
1.3 嵌入式操作系统 .....	10
1.3.1 操作系统概念和分类 .....	10
1.3.2 实时操作系统 .....	13
1.3.3 常见的嵌入式操作系统 .....	15
1.4 实时操作系统的内核 .....	19
1.4.1 任务管理 .....	19
1.4.2 任务间的通信和同步 .....	24
1.4.3 存储器管理 .....	26
1.4.4 定时器和中断管理 .....	26
习 题 .....	27

## 第 2 章 嵌入式系统开发过程

2.1 嵌入式软件开发的特点 .....	28
2.2 嵌入式软件的开发流程 .....	29
2.3 嵌入式系统调试 .....	34
2.4 板级支持包 .....	37
习 题 .....	38

## 第3章 Cortex-M 体系结构

3.1	ARM 体系结构概述 .....	39
3.1.1	ARM 体系结构的特点 .....	39
3.1.2	流水线 .....	41
3.1.3	ARM 处理器核 .....	42
3.1.4	结构框图 .....	47
3.1.5	典型的连接方式 .....	48
3.1.6	ARM JTAG 调试接口 .....	49
3.2	编程模型 .....	50
3.2.1	Thumb-2 指令集体系结构(ISA) .....	50
3.2.2	寄存器 .....	51
3.2.3	工作模式和特权级别 .....	54
3.2.4	Cortex-M3 堆栈 .....	55
3.2.5	数据类型 .....	55
3.2.6	存储器和存储器映射 I/O .....	56
3.3	ARM 基本寻址方式 .....	57
3.4	Thumb-2 指令集说明 .....	62
3.4.1	条件执行 .....	62
3.4.2	指令分类说明 .....	63
3.4.3	Thumb-2 指令集的特点 .....	77
3.4.4	ARM 汇编语言程序设计 .....	78
3.5	存储器映射 .....	81
3.5.1	存储系统简介 .....	81
3.5.2	存储器映射空间 .....	81
3.5.3	位绑定操作 .....	83
3.6	中断和异常 .....	86
3.6.1	异常类型 .....	87
3.6.2	优先级的定义 .....	88
3.6.3	向量表 .....	90
3.6.4	中断输入及挂起行为 .....	91
3.6.5	NVIC 与中断控制 .....	92
3.6.6	中断/异常的响应序列 .....	94
3.6.7	尾链中断 .....	95
3.6.8	迟到异常处理 .....	96
3.6.9	异常返回值 .....	96
3.6.10	中断延迟 .....	97
3.7	ARM Cortex-M 的优势 .....	97
	习 题 .....	99



**第 4 章 ARM Cortex - M 微控制器**

4.1	Stellaris 微控制器分类 .....	100
4.2	Stellaris 微控制器特性 .....	101
4.3	Stellaris 微控制器结构 .....	102
4.3.1	内部资源概述 .....	104
4.3.2	芯片引脚描述 .....	108
4.3.3	存储器组织 .....	112
4.4	系统控制器单元 .....	113
4.4.1	系统控制 .....	113
4.4.2	复位控制 .....	114
4.4.3	时钟控制 .....	116
4.4.4	功耗控制 .....	118
4.5	Stellaris 微控制器最小系统 .....	119
4.6	Stellaris 外设驱动库 .....	128
4.7	微控制器开发工具 .....	135
4.7.1	CCSv4 集成开发环境简介 .....	136
4.7.2	RealView MDK 开发工具简介 .....	138
4.7.3	IAR EWARM 集成开发环境简介 .....	140
4.7.4	CodeSourcery Sourcery G++ 开发工具简介 .....	142
	习 题 .....	143

**第 5 章 片内资源的编程技术**

5.1	通用输入/输出端口 GPIO .....	145
5.1.1	GPIO 概述 .....	145
5.1.2	功能描述 .....	145
5.1.3	与 GPIO 有关的库函数 .....	147
5.1.4	LED 显示例程 .....	150
5.2	按键中断 .....	152
5.2.1	中断基础知识 .....	152
5.2.2	中断控制 .....	152
5.2.3	与中断有关的库函数 .....	153
5.2.4	按键中断例程 .....	154
5.3	定时器 .....	155
5.3.1	定时器概述 .....	155
5.3.2	功能描述 .....	156
5.3.3	与定时器有关的库函数 .....	161
5.3.4	通用定时器示例程序 .....	164
5.3.5	看门狗定时器 .....	165

5.4 脉宽调制器 PWM .....	167
5.4.1 脉宽调制基本原理 .....	167
5.4.2 与 PWM 有关的库函数 .....	171
5.4.3 PWM 示例程序 .....	172
5.5 串行口 UART .....	173
5.6 模/数转换器 ADC .....	177
习 题 .....	180

## 第 6 章 人机交互的接口及编程

6.1 行列式键盘 .....	182
6.1.1 键盘硬件接口设计 .....	182
6.1.2 键盘扫描程序设计 .....	183
6.2 可编程键盘显示接口芯片 HD7279 .....	187
6.2.1 HD7279 简介及接口 .....	187
6.2.2 数码管和键盘示例 .....	190
6.3 字符型 LCD 显示模块 .....	193
6.3.1 字符型 LCD 模块和指令 .....	193
6.3.2 LCD 硬件接口及编程 .....	195
6.4 点阵式 LCD 显示模块 .....	199
6.5 16×16 LED 点阵显示汉字 .....	202
6.6 带触摸屏的 QVQA LCD 模块 .....	205
习 题 .....	209

## 第 7 章 总线接口及编程

7.1 串行总线接口 SPI .....	210
7.1.1 串行总线接口 SPI 简介 .....	210
7.1.2 Stellaris 的同步串行接口 (SSI) .....	210
7.1.3 与串行总线接口 SPI 有关的函数 .....	211
7.1.4 A/D 转换器 TLC2543 .....	213
7.1.5 D/A 转换器 TLV5616 .....	216
7.1.6 Flash 存储器扩展 .....	219
7.1.7 SD/MMC 卡 .....	221
7.2 I <sup>2</sup> C 接口 .....	223
7.2.1 I <sup>2</sup> C 总线简介 .....	223
7.2.2 I <sup>2</sup> C 总线的通信规约 .....	224
7.2.3 Stellaris 的 I <sup>2</sup> C 接口 .....	225
7.2.4 与 I <sup>2</sup> C 有关的库函数 .....	227
7.2.5 I <sup>2</sup> C 的 IC 卡接口及编程 .....	228
7.2.6 I <sup>2</sup> C 的数据采集模块 .....	231

7.3 I <sup>2</sup> S 接口 .....	234
7.3.1 I <sup>2</sup> S 总线简介 .....	234
7.3.2 Stellaris 的 I <sup>2</sup> S 接口 .....	236
7.3.3 与 I <sup>2</sup> S 有关的库函数 .....	239
7.3.4 I <sup>2</sup> S 的音频芯片及编程 .....	240
7.4 USB 接口 .....	243
7.4.1 USB 接口概述 .....	243
7.4.2 Stellaris 的 USB 接口 .....	247
7.4.3 与 USB 有关的库函数 .....	257
7.4.4 基于 USB 设备的 Device 实验 .....	261
7.4.5 基于 USB 设备的 Host 实验 .....	261
习 题 .....	263
<b>第 8 章 网络接口及编程</b>	
8.1 CAN 总线通信接口 .....	264
8.1.1 CAN 总线简介 .....	264
8.1.2 Stellaris 内部集成的 CAN 控制器 .....	265
8.1.3 与 CAN 有关的库函数 .....	271
8.1.4 双机数据传输通信 .....	272
8.2 以太网接口 .....	276
8.2.1 以太网接口简介 .....	276
8.2.2 Stellaris 内部集成的以太网控制器 .....	276
8.2.3 与以太网有关的库函数 .....	281
8.2.4 基于以太网的 Web 服务器 .....	282
习 题 .....	284
<b>第 9 章 电机控制的接口及编程</b>	
9.1 步进电机控制模块 .....	285
9.2 直流电机控制模块 .....	290
9.3 带以太网和 CAN 的无刷直流电机控制 .....	294
习 题 .....	297
<b>第 10 章 物联网数据采集与传输</b>	
10.1 物联网简介 .....	298
10.2 温湿度采集 .....	300
10.2.1 温度和湿度传感器 SHT11 .....	300
10.2.2 温度传感器 DS18B20 .....	305
10.3 光照度传感器采集 .....	308
10.4 加速度传感器采集 .....	311

# 目 录

10.5	RFID 采集 .....	314
10.6	ZigBee 点对点通信 .....	317
10.7	无线传感器网络协调器节点设计 .....	325
	习 题 .....	329

## 附录 A $\mu$ C/OS - II 移植

A.1	移植代码说明 .....	330
A.1.1	includes.h 文件 .....	331
A.1.2	OS_CPU.H 文件 .....	331
A.1.3	OS_CPU_C.C 文件 .....	332
A.1.4	OS_CPU_A.ASM 文件 .....	333
A.2	异常/中断 .....	339
A.3	启动文件和目标板配置文件 .....	339
A.3.1	startup_ewarm.c 文件 .....	339
A.3.2	Target.H 文件 .....	341
A.3.3	Target.C 文件 .....	344
A.4	用户文件 .....	346

参考文献 .....	348
------------	-----

# 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。在有些系统里面也有计算机,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”到更大的、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,如天天必用的手机、戴在手腕上的电子表、烹调用的微波炉、办公室里的打印机、汽车里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)、数码相机和数码摄像机等,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,到办公室里的远程会议系统等,这些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对功能、可靠性、成本、体积和功耗等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、网卡、声卡、打印机、扫描仪和数码相机等均由嵌入式处理器进行控制的。在制造业、过程控制、通信、仪器、仪表、汽车、船舶、航空、航天、军事装备和消费类产品等方面,嵌入式系统都有用武之地。

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 1999 年 1 月访华时曾预言“4~5 年以后嵌入式智能电脑将是继 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 1 000 多种嵌入式处理器芯片,其中使用最广泛的有 8051、PIC、AVR、ARM、MIPS、PowerPC 和 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的完成存储、通信、调试和显示等辅助功能的其他部件。根据外围设备的功能可分为以下 3 类:

- 存储器 静态易失型存储器(RAM 和 SRAM)、动态存储器(DRAM)、非易失型存储器(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 通用串行总线接口和 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的,基于相应的嵌入式硬件平台的,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,因此为降低系统的成本,除了精简每个硬件单元的成本外,应尽量减少应用软件的资源消耗,尽可能优化。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:

① 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。

② 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,以缩短程序二进制代码长度,提高执行速度。

③ 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。

④ 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,实际应用环境越来越复杂,从而使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。

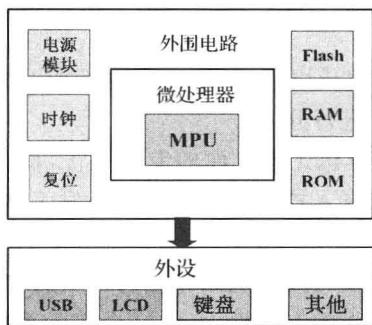


图 1-1 典型嵌入式系统基本组成——硬件

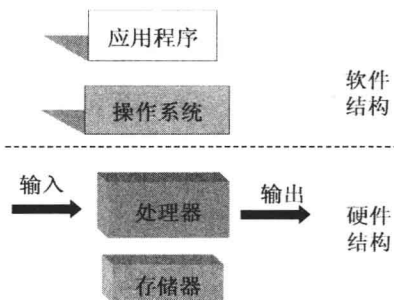


图 1-2 典型嵌入式系统基本组成——软件

### 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、面向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特定用户群所设计的系统中。它通常具有低功耗、体积小、集成度高和成本低等特点,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、高度分散和不断创新的知识集成系统。

③ 嵌入式系统与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,一般都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

## 第1章 嵌入式系统基础

相同,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,所以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。一般不会用嵌入式系统作为开发应用软件的环境,也不会运行一些大型的软件。因此,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,并因此成为操作系统的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、可靠和高效。

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

### 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。从海湾战争到伊拉克战争都广泛使用。我国嵌入式计算机最早用于导弹控制。

② 民用。我国各种信息家电产品,如数字电视机、机顶盒,数码相机,VCD 和 DVD 音响设备,可视电话,家庭网络设备,洗衣机,电冰箱及智能玩具等,都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及汽车电子设备等,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,以及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,如 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、辅助诊断系统和专家系统等。

### 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足够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做出响应的系统。

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日期,任务必须在这个截止日期之内完成,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,这就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。所谓可靠性,一方面是指系统的正确性,即系统所产生的结果在返回值和运行费时上都是正确的;另一方面是指系统的健壮性。也就是说,虽然系统出现了错误或外部环境预先假定的外部环境不符合,但系统仍然可以处于可预测状态,它仍可以安全地带错运行和平缓地降级。

实时系统中主要通过以下 3 个指标来衡量系统的实时性:

- 响应时间(response time)是计算机从识别一个外部事件到做出响应的时间。
- 生存时间(survival time)是数据的有效等待时间,在这段时间里数据是有效的。
- 吞吐量(throughput)指在一给定时间内,系统可以处理的事件总数。吞吐量通常比平均响应时间的倒数小一点。

实时系统根据响应时间可分为以下 3 种类型:

① 强实时系统。在强实时系统中,各任务不仅要保证执行过程和结果的正确性,同时还要保证在系统能够允许的时间内完成任务。它的响应时间在毫秒或微妙的数量级上。这对于关系到安全、军事领域的软硬件系统是至关重要的。