



光电科学与工程专业规划教材

嵌入式系统设计与工程实践

——基于Cortex-M3内核处理器LPC17XX

Embedded System Design for Engineering and Practice
—Base on Cortex-M3 Core Processor LPC17××

漆 强 欧中华 刘子骥 杨先明 孙丽佳 蒋 泉 编著



国防工业出版社
National Defense Industry Press

光电科学与工程专业规划教材

嵌入式系统设计工程实践——基于 Cortex-M3 内核处理器 LPC17xx

漆 强 欧中华 刘子骥 杨先明 孙丽佳 蒋 泉 编著



国防工业出版社

· 北京 ·

内 容 简 介

本书主要以 LPC1700 系列微控制器为硬件平台,详细介绍 ARM Cortex-M3 内核微控制器的原理与应用开发技术,是读者学习 Cortex-M3 内核微控制器应用开发的入门教程。全书共 12 章,分成五个部分:第一部分为第 1 章,介绍嵌入式系统的基本概念。第二部分包括第 2 章和第 3 章,介绍 LPC1700 系列微控制器的基本结构,包括引脚配置、时钟和功率管理等,并设计了基于 LPC1768 微控制器的硬件实验平台。第三部分为第 4 章,介绍主流的 ARM 开发工具 Realview MDK 集成开发环境的使用。第四部分为第 5~10 章,介绍 LPC1700 系列微控制器的主要功能模块,并提供大量应用实例。第五部分为第 11 章和第 12 章,介绍嵌入式系统编程的模块化和层次化设计思想,以及一些基本的编程准则和编程经验,并介绍了 RL-RTX 嵌入式实时操作系统的应用。

为了拓展学生的自学能力,本书设计了基于 LPC1768 的最小系统实验平台,并提供了大量的应用代码,该平台可以和本书配合使用。

本书内容丰富实用,层次清晰,叙述详尽,并提供了硬件开发平台和应用程序,方便教学和自学。本书可作为高等院校电子科学与技术类专业的本科生及研究生“嵌入式系统”课程的入门教材,也可以供嵌入式爱好者或从事嵌入式系统应用的工程技术人员参考。

图书在版编目(CIP)数据

嵌入式系统设计工程实践:基于 Cortex-M3 内核处理器
LPC17XX/漆强等编著. --北京:国防工业出版社,2015.4
ISBN 978-7-118-09938-6

I. ①嵌… II. ①漆… III. ①微型计算机—系统设计
IV. ①TP360.21

中国版本图书馆 CIP 数据核字(2015)第 050489 号

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

北京奥鑫印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 17¼ 字数 405 千字

2015 年 4 月第 1 版第 1 次印刷 印数 1—3000 册 定价 48.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)88540777

发行邮购:(010)88540776

发行传真:(010)88540755

发行业务:(010)88540717

前 言

随着电子技术的迅猛发展,嵌入式系统的应用无处不在。从我们使用的家用电器,到随身携带的智能手机和平板电脑,再到目前发展如火如荼的智能穿戴设备和物联网应用,嵌入式系统和我们每个人的生活息息相关。

嵌入式系统包含了硬件和软件两个方面的内容,硬件主要是微控制器和相关外设的应用,软件主要包含了启动代码、嵌入式操作系统和应用程序。硬件系统的核心——微控制器,往往采用的是 ARM 公司设计的 Cortex 内核,目前全世界有 200 多家领先的半导体厂商获得了 ARM 授权,基于 Cortex 内核的微控制器占领了嵌入式系统的大部分份额。其中 Cortex-M0、Cortex-M3 和 Cortex-M4 内核一般用于中低端和低功耗的应用环境中,如智能穿戴设备、工业控制和物联网等领域。在这些应用中,一般采用嵌入式实时操作系统,如 FreeRTOS、RTX 和 $\mu\text{C}/\text{OS-II}$ 等。Cortex-A 系列内核则用于高端应用,如智能手机、平板电脑等消费类电子产品中。在这些应用中,一般采用移动平台操作系统,如苹果公司的 IOS 系统和谷歌公司的安卓系统等。

目前大部分高校在“单片机原理和应用”、“微机原理与接口技术”和“嵌入式系统”等课程中往往采用 MCS-51 单片机或 8086 处理器作为主要芯片来介绍(其中 8086 处理器芯片已经停产多年,无法获取对应的硬件开发平台)。随着电子技术的发展,应用环境的复杂化,对产品可靠性和稳定性的要求日益提高,MCS-51 单片机和 8086 处理器等芯片由于自身硬件结构的限制,越来越满足不了目前嵌入式系统的应用要求,这就需要我们摒弃这些传统芯片,采用新一代的 Cortex 内核微控制器作为讲授内容,以满足工程界对创新实践人才的需求,减少高校教育和工程实践的脱节现象。

NXP 公司的 LPC1700 系列微控制器,采用的是第二代 Cortex-M3 内核,时钟速度可达 100MHz,可以实现更低的功率损耗和维持更久的电池寿命,在调试和解决疑难问题的过程中,具有更好的灵活性,在电子仪表、照明、工业网络、报警系统、白色家电和电机控制等领域中得到了广泛的应用。本书主要以 LPC1700 系列微控制器为硬件平台,详细介绍 ARM Cortex-M3 微控制器的原理与应用开发技术。

第 1 章简要介绍嵌入式系统的概念、分类,以及常见的嵌入式处理器和嵌入式操作系统。

第 2 章介绍 LPC1700 系列微控制器的基本结构,包括引脚基本介绍、时钟、引脚配置和功率管理等。

第 3 章介绍基于 LPC1768 微控制器的最小系统电路的设计,并提供了一个本书使用的最小系统电路板,详细分析了电路的设计原理。

第 4 章介绍工程界主流的 ARM 开发工具 Realview MDK,详细介绍了它的用法。

第 5~10 章介绍 LPC1700 系列微控制器的通用输入输出接口(GPIO)、中断系统、定

时器/计数器、异步串口、模数转换器和系统节拍定时器等各个重要功能模块,给出相关寄存器的说明,并提供了大量应用实例和工程经验的总结。

第 11 章介绍嵌入式系统的编程设计,总结了作者多年的设计经验,提出了一些基本的设计思想和设计准则。

第 12 章介绍嵌入式操作系统基础及 MDK 开发工具提供的嵌入式操作系统 RL-RTX 的简单应用。

在教学实践中,本书设计了基于 LPC1768 的最小系统实验平台,并提供了大量的应用代码,该硬件平台包含了模数转换模块、指示灯模块、键盘模块、蜂鸣器模块、USB 转 TTL 串口模块和 TFT 液晶显示模块,学生在该平台上利用提供的应用代码可以理论结合实践,熟练掌握 Cortex-M3 内核微控制器的应用开发。同时该平台设计小巧,摒弃了传统硬件开发平台大而全的设计思路,总面积为 10cm×10cm,采用 USB 接口供电,USB 接口还可以直接和 PC 进行通信。这个硬件平台可以作为“口袋实验室”形式,在课堂上提供给学生使用,要求学生自带笔记本电脑,教师授课时可以讲完一个模块后,预留半小时时间给学生进行编程练习,立刻应用所学知识编写测试程序,达到活学活用的目的。学生也可以将该平台带回寝室使用,弥补课堂教学时间不足的问题。还可以基于该硬件平台和自身专业相结合,开发一些有创意的小制作。同时,该硬件平台也可作为参加电子设计竞赛的最小系统模块。

本书可作为高等院校电子科学与技术类专业本科生及研究生“单片机原理和应用”“微机原理与接口技术”和“嵌入式系统”等课程的教材,也可以作为嵌入式爱好者或从事嵌入式系统应用开发的工程技术人员参考。

本书在编写过程中参考和借鉴了大量的相关资料(见参考文献)及网络资源,并引用了其中的一些文字和代码,在此谨对这些作者表示衷心的感谢。

感谢我的家人在编写过程中提供的帮助,特别是我的爱人绘制了书中大量的插图和表格,协助我完成了文档的排版。

感谢刘爽老师和周鹰老师在教材编写中提供的帮助。

特别鸣谢广州周立功单片机公司提供了 LPC1700 系列微控制器的数据手册和应用笔记,该公司对国内 ARM 技术的推广和应用做出了不懈的努力。

鉴于 Cortex-M3 内核的强大功能和 LPC1700 系列微控制器的丰富外设,限于篇幅,本书只对主要内容进行了介绍,关于通用 DMA、USB、SSP、I²C、CAN 总线和以太网部分,请读者参考 LPC1700 系列微控制器的用户手册。由于编者水平有限,书中难免存在遗漏与不足之处,恳请广大读者批评指正。

漆 强

2014 年 11 月于电子科技大学沙河校区

目 录

第 1 章 嵌入式系统概论	1
1.1 嵌入式系统	1
1.1.1 嵌入式系统的应用	1
1.1.2 嵌入式系统的定义和特点	3
1.1.3 嵌入式系统的组成	4
1.2 嵌入式处理器	7
1.2.1 简介	7
1.2.2 分类	8
1.2.3 ARM 微处理器	8
1.3 嵌入式操作系统	9
1.3.1 基本概念	9
1.3.2 使用嵌入式操作系统的意义	10
1.3.3 常用的嵌入式操作系统	10
1.4 嵌入式系统学习方法	14
习题	14
第 2 章 LPC1700 系列微控制器硬件体系结构	15
2.1 LPC1700 系列微控制器简介	15
2.2 LPC1768 芯片引脚介绍	17
2.3 时钟管理	27
2.3.1 时钟源选择	27
2.3.2 锁相环 PLL0 和 PLL1	30
2.3.3 时钟分频设置	34
2.3.4 时钟初始化的编程实例	38
2.4 外设功率控制	45
2.5 引脚连接模块	46
习题	53
第 3 章 LPC1768 硬件电路的设计	54
3.1 硬件系统的组成	54
3.2 电源电路的设计	56
3.2.1 电源电路的设计	56
3.2.2 复位电路的设计	58
3.2.3 时钟电路的设计	59

3.2.4	调试电路的设计	60
3.2.5	处理器核心电路设计	60
3.3	外围电路设计	62
3.3.1	USB 转串口的设计	62
3.3.2	按键电路	64
3.3.3	指示灯电路	66
3.3.4	蜂鸣器电路	67
3.3.5	模数转换电路	68
3.3.6	液晶显示电路	68
习题		70
第 4 章	RealViewMDK 开发软件的使用	71
4.1	概述	71
4.2	JLINK 仿真器介绍	72
4.3	RealViewMDK 的安装	72
4.4	RealViewMDK 的使用	75
4.4.1	新建工程	76
4.4.2	配置工程编译环境	80
4.4.3	程序编译和下载	88
4.4.4	程序的仿真调试	90
4.5	RealViewMDK 的使用技巧	93
习题		94
第 5 章	通用输入/输出接口	95
5.1	概述	95
5.1.1	特性	95
5.1.2	引脚描述	95
5.2	相关寄存器描述	96
5.3	使用说明	107
5.3.1	使用步骤	107
5.3.2	使用注意事项	107
5.4	应用示例	108
5.4.1	GPIO 输出	109
5.4.2	GPIO 输入	109
5.5	GPIO 的库函数编写	112
习题		118
第 6 章	中断系统	119
6.1	嵌套向量中断控制器	119
6.1.1	NVIC 特性	119
6.1.2	中断源	119
6.1.3	NVIC 的编程	121

6.2	外部中断	123
6.2.1	概述	123
6.2.2	相关寄存器描述	124
6.2.3	中断设置	127
6.2.4	使用说明	129
6.2.5	使用注意事项	129
6.2.6	应用示例	130
6.3	GPIO 中断	134
6.3.1	概述	134
6.3.2	相关寄存器描述	134
6.3.3	中断设置	137
6.3.4	使用说明	138
6.3.5	GPIO 中断和外部中断的区别	138
6.3.6	应用示例	139
	习题	140
第 7 章	定时器/计数器	141
7.1	概述	141
7.1.1	特性	141
7.1.2	引脚描述	141
7.2	相关寄存器描述	142
7.3	使用说明	151
7.3.1	内部结构	151
7.3.2	使用步骤	154
7.3.3	中断设置	156
7.3.4	使用注意事项	158
7.4	使用示例	159
	习题	169
第 8 章	异步串口	171
8.1	概述	171
8.1.1	特性	171
8.1.2	引脚描述	171
8.1.3	典型应用电路	172
8.2	相关寄存器描述	174
8.3	串行通信基本概念	183
8.3.1	异步串行通信的特点	183
8.3.2	关于异步串行通信的准确性讨论	184
8.3.3	UART 初始化设置	185
8.4	使用说明	187
8.4.1	内部结构	187

8.4.2	工作原理	188
8.4.3	操作步骤	189
8.5	UART 中断	189
8.6	使用示例	193
8.7	串口通信的上位机程序开发	199
习题		201
第 9 章	模数转换器	202
9.1	概述	202
9.1.1	基本概念	202
9.1.2	特性	203
9.1.3	引脚描述	204
9.2	相关寄存器描述	204
9.3	使用说明	211
9.3.1	内部结构	211
9.3.2	使用步骤	212
9.3.3	中断配置	212
9.3.4	使用注意事项	214
9.4	应用示例	214
9.5	ADC 的模块化编程	221
习题		224
第 10 章	系统节拍定时器	225
10.1	概述	225
10.2	相关寄存器描述	225
10.3	工作原理和中断设置	228
10.3.1	工作原理	228
10.3.2	中断设置	228
10.4	应用示例	229
习题		234
第 11 章	嵌入式系统编程	235
11.1	程序编写风格	235
11.2	C 语言设计相关	236
11.2.1	数据类型	236
11.2.2	if 语句的用法	237
11.2.3	volatile 的用法	238
11.2.4	static 的用法	239
11.2.5	const 的用法	241
11.3	嵌入式系统命名规则	242
11.3.1	常量命名	242
11.3.2	变量命名	243

11.3.3 函数命名	243
11.4 模块化编程	243
11.5 分层设计思想	247
习题	249
第 12 章 嵌入式实时内核 RL-RTX 的应用	250
12.1 程序架构	250
12.2 RL-RTX 简介	252
12.3 RL-RTX 主要应用函数介绍	255
12.3.1 任务管理函数	255
12.3.2 时间管理函数	257
12.3.3 事件标志函数	258
12.4 RL-RTX 应用示例	260
12.4.1 时间管理函数应用:闪灯程序	262
12.4.2 事件标志函数应用:流水灯程序	266
12.5 RL-RTX 调试说明	270
习题	272
参考文献	273

第 1 章 嵌入式系统概论

1.1 嵌入式系统

随着科学技术的发展,嵌入式设备在我们的生产和生活中占据了越来越重要的地位,极大地改变了人们的生活、工作和娱乐方式。

嵌入式系统的发展离不开通用计算机技术的发展:在微型计算机时代,人们开始尝试将通用计算机系统用于航海、机械控制等领域,但是在使用过程中发现,由于通用计算机体积较大,将会占用大量的空间,同时在控制领域,往往只需要进行简单的输出输入,而不需要海量的存储和高分辨率的显示。正是由于在上述领域的要求,向计算机专家提出了新的设计需求:设计一种单芯片的微处理器,将存储器、输出输入接口、定时器和中断系统等都集成在上面,并减少芯片体积和功耗。于是各个半导体厂家以此为目的,不断进行研发,最终以 1978 年英特尔公司设计的 MCS-51 系列微处理器为代表,诞生了最初的嵌入式系统,和通用计算机系统形成了现代计算机的两大发展分支:通用计算机按照高速运算和海量存储的技术方向发展;嵌入式计算机系统则向低功耗、高可靠性和便携性等方向发展。

1.1.1 嵌入式系统的应用

嵌入式系统的应用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、鼠标、打印机、扫描仪、数字相机、USB 集线器等均是由嵌入式微处理器控制的。嵌入式系统在各行各业都得到了广泛的应用,其应用领域包括:

1. 工业控制

将嵌入式微处理器应用于工业控制,进行工业自动化的开发,取代传统的人工控制,大大提高生产效率和产品质量、减少人力成本,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。近年来,全球最大的代工企业鸿海科技也逐渐开始在生产线上采用机器人等控制设备取代人工作业。

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 模块、GSM 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。汽车、电动车、混合动力汽车越来越多地采用嵌入式系统来节能减排,其他汽车安全系统,包括防抱死制动系统、电子稳定控制系统、牵引力控制系统、自动四驱系统,都离不开嵌入式系统。

3. 环境工程和自然

在很多环境恶劣、地况复杂的地区,嵌入式系统将实现环境工程和自然的无人监测。

4. 国防和航天

第一款大批量生产的嵌入式系统正是美国在 1961 年发布的“民兵” I 型导弹内的 D-17 自动制导计算机。当“民兵” II 导弹在 1966 年开始生产的时候, D-17 升级成一种新型计算机, 其中首次大量使用了集成电路, 仅这个项目就将与非门模块的单价从 1000 美元降低至 3 美元, 大大降低了生产成本。我国“神舟”飞船和“长征”火箭中也采用了很多嵌入式系统。

5. 消费类电子和便携式医疗设备

这些设备对产品的体积和功耗都有严格的要求, 嵌入式微处理器体积微小, 单芯片上集成了大部分控制能力, 正好满足需求, 应用范围包括平板电脑、高清播放器、智能机顶盒、便携式血糖仪和心电监护仪等。

6. 智能家居

这个领域将是嵌入式系统今后最大的发展空间, 各种冰箱、空调等家电的网络化、智能化将引领人们的生活步入一个崭新的空间, 即使你不在家里, 也可以通过手机进行远程控制。2014 年 1 月, 谷歌公司以 32 亿美元的天价收购智能家居公司 Nest Labs, 该公司的主要产品是 Nest 智能恒温器, 售价高达 249 美元。该恒温器非常智能化, 它内置了多种类型的传感器, 可以不间断地监测室内的温度、湿度、光线以及恒温器周围的环境变化, 比如它可以判断房间中是否有人(是否有移动), 并以此决定是否开启温度调节设备。它还具有学习能力, 比如你每一次在某个时间设定了某个温度, 它都会记录一次, 然后经过一周的时间, 它就能学习和记住用户的日常作息习惯和温度喜好, 并且它会利用算法自动生成一个设置方案, 只要你的生活习惯没有发生变化, 你就不再需要手动设置 Nest 恒温器。同时支持联网, 可以通过手机进行控制。该设备外观非常漂亮, 其设计师是被称为 iPod 之父的托尼·法戴尔。Nest 智能恒温器如图 1-1 所示。



图 1-1 Nest 智能恒温器

7. 智能穿戴设备

这是目前发展最为热门的一个领域,它应用嵌入式技术对日常穿戴进行智能化设计,如眼镜、手表、服饰及鞋等,该领域目前的应用集中在两个方面:一个是运动健身;一个是医疗保健。如图 1-2 所示是国内研发的咕咚手环,其主要功能为“运动状况提醒”和“睡眠监测”,并可以和手机联网,将自己的运动状态发送到社交网络和爱好者分享。

在咕咚手环的电路设计中,采用了柔性电路板,将显示部分电路和控制部分电路对折在一起。显示部分使用了 5×19 个行列排布 LED 灯,用于显示运动时间和运动步数;控制部分采用加速度传感器完成运动状态的检测,采用蓝牙芯片完成和手机的信息传递,主控芯片则使用了意法半导体公司的 STM32 系列微处理器完成对整个系统的控制、显示和数据处理。

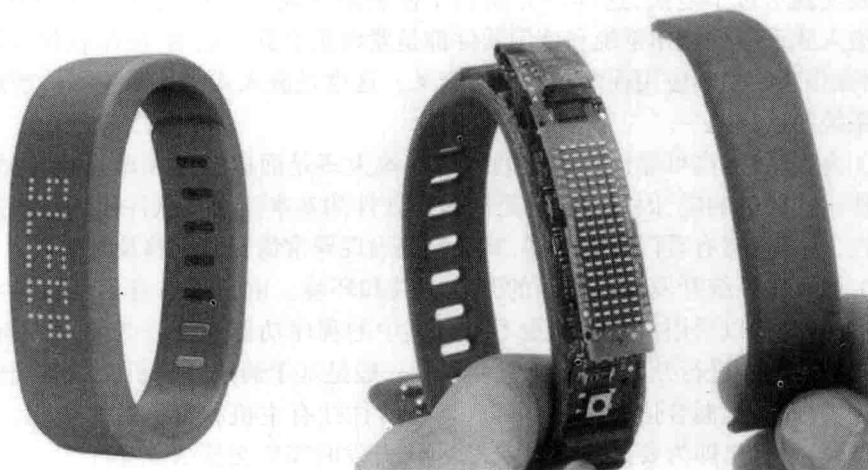


图 1-2 咕咚智能手环外观图和拆解图

1.1.2 嵌入式系统的定义和特点

嵌入式系统诞生于微型计算机,是嵌入到对象体系中,实现嵌入对象智能化的计算机。根据 IEEE(电气和电子工程师协会)的定义,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)。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备。

国内普遍认同的嵌入式系统定义为:以应用为中心,以计算机技术为基础,软硬件可裁剪,适应应用系统对功能、可靠性、成本、体积、功耗等严格要求的专用计算机系统。

广义上的嵌入式系统将凡是带有微处理器的专用系统都称为嵌入式系统,如单片机系统和带 OS 的 ARM 系统。一些小家电,如豆浆机,只需要加热,预打浆,打浆,煮浆,断电报警等五个流程即可完成操作,各个任务之间是线性的先后关系,没有并发的任务,采用一般的循环程序就可以了。这类单片机系统,在完成较为单一的专业功能时具有简洁高效的特点,但是由于它们没有采用操作系统,所以管理硬件和软件的能力有限,在实现复杂的多任务功能时往往困难重重,甚至无法实现。

狭义上的嵌入式系统是指那些使用嵌入式微处理器构成的独立系统,具有嵌入式操作系统,完成特定功能,用于特定场合。

与通用的计算机系统相比,嵌入式系统一般具备如下特点:

(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,系统资源有限,所以内核较之传统的操作系统要小得多。比如 $\mu\text{C}/\text{OS}-\text{II}$ 操作系统内核一般只有几十 KB,而传统计算机的操作系统动则几个 GB。

(2) 专用性强。由于嵌入式系统的软件和硬件结合非常紧密,一般要针对具体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进行修改。

(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。大部分嵌入式设备的操作系统和应用软件都是紧密集合到一起的:应用软件控制软件的运作,而操作系统控制应用程序与硬件的交叉。这也是嵌入式操作系统和一般通用计算机操作系统的区别之一。

(4) 高实时性,高可靠性。由于嵌入式系统大多是面向控制领域,要求对外界的各种信息作出实时的响应,因此实时性是嵌入式软件的基本要求。软件代码要求高质量和高可靠性,一般设置有看门狗定时器,确保系统出现异常情况时能够及时重启。

(5) 嵌入式系统开发需要专门的开发工具和环境。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、混合信号示波器等。开发时往往有主机和目标机的概念,主机用于程序的开发,目标机即为要开发的嵌入式设备,开发时需要交替结合进行。

1.1.3 嵌入式系统的组成

嵌入式系统主要由嵌入式硬件系统和嵌入式软件系统两部分组成,其结构层次如图 1-3 所示。

嵌入式硬件系统主要包括嵌入式微处理器、存储器、外围接口电路和电源等部分。嵌入式微处理器是嵌入式系统的核心,是控制、辅助系统运行的硬件单元。常用的嵌入式外围接口电路主要有串口、以太网接口、USB 接口、音频接口、液晶显示屏和摄像头等。针对不同的应用,具体的嵌入式设备所需的外围接口电路不同。

嵌入式软件系统包括启动代码、驱动程序、操作系统和应用程序等。启动代码主要用于完成硬件初始化,为操作系统建立一个良好的运行环境,驱动程序用于实现嵌入式系统硬件和软件之间的接口;操作系统实现系统的进程调度、任务处理;应用程序实现系统功能的应用,是根据具体的应用环境由设计人员自行编写的。

1. 嵌入式硬件系统

硬件部分包含嵌入式微处理器、存储器(RAM、ROM、Flash 等)、通用设备接口和 I/O 接口(A/D、D/A 和 I/O 等)。在一片嵌入式处理器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。

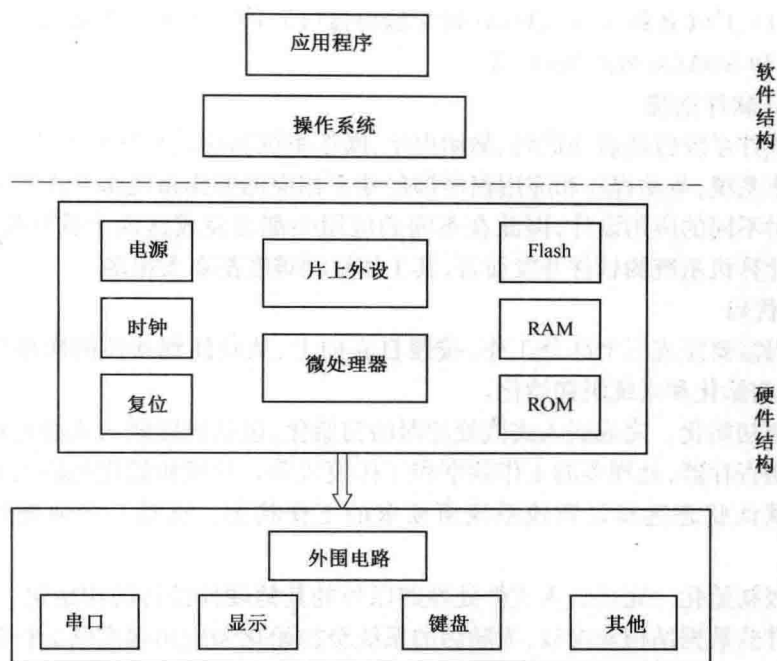


图 1-3 嵌入式系统层次结构图

1) 嵌入式微处理器

嵌入式系统硬件部分的核心是嵌入式微处理器。嵌入式微处理器与通用 CPU 最大的不同在于嵌入式微处理器大多工作在为特定用户群所专门设计的系统中,它将通用计算机系统中许多由板卡完成的模块集成在芯片内部,从而有利于嵌入式系统在设计时趋于小型化,同时还具有很高的效率和可靠性。

嵌入式微处理器的体系结构可以采用冯·诺依曼体系或哈佛体系结构;指令系统可以选用精简指令系统(Reduced Instruction Set Computer, RISC)或复杂指令系统 CISC(Complex Instruction Set Computer, CISC)。

嵌入式微处理器有各种不同的体系,即使在同一体系中也具有不同的时钟频率和数据总线宽度,或集成了不同的外设和接口。据不完全统计,目前全世界嵌入式微处理器已经超过 1000 多种,体系结构有 30 多个系列,其中主流的体系有 ARM、MIPS、PowerPC、X86 和 SH 等。但与全球 PC 市场不同的是,没有一种嵌入式微处理器可以主导市场,仅以 32 位的产品而言,就有 100 种以上的嵌入式微处理器。嵌入式微处理器的选择是根据具体的应用而决定的。

2) 通用设备接口和 I/O 接口

嵌入式系统和外界交互需要一定形式的通用设备接口,如 A/D、D/A、I/O 等,外设通过和片外其他设备或传感器的连接来实现微处理器的输入/输出功能。每个外设通常都只有单一的功能,它可以在芯片外也可以内置芯片中。外设的种类很多,可从一个简单的串行通信设备到非常复杂的 802.11 无线设备。

目前嵌入式系统中常用的通用设备接口有 A/D(模/数转换接口)和 D/A(数/模转换接口),I/O 接口有 RS-232 接口(串行通信接口)、Ethernet(以太网接口)、USB(通用

串行总线接口)、I²S(音频接口)、VGA(视频输出接口)、I²C(集成电路总线)、SPI(串行外围设备接口)和 IrDA(红外线接口)等。

2. 嵌入式软件系统

嵌入式软件系统包括启动代码、驱动程序、操作系统和应用程序四个大的模块,其中的核心为操作系统,驱动程序和应用程序都是基于相应的操作系统来开发。由于嵌入式硬件都是针对不同的应用设计,因此在不同的应用中都要完成这四个软件模块的移植,相对于通用计算机系统的软件开发而言,其工作量和难度都要大很多。

1) 启动代码

启动代码需要完成三个部分工作,按照自底向上、从硬件到软件的次序依次为片级初始化、板级初始化和系统级初始化。

(1) 片级初始化。完成嵌入式微处理器的初始化,包括设置嵌入式微处理器的核心寄存器和控制寄存器,处理器的工作频率和工作模式等。片级初始化把嵌入式微处理器从上电时的默认状态逐步设置成系统所要求的工作状态。这是一个纯硬件的初始化过程。

(2) 板级初始化。完成嵌入式微处理器以外的其他硬件设备的初始化。另外,还需设置某些软件的数据结构和参数,为随后的系统级初始化和应用程序的运行建立硬件和软件环境。这是一个同时包含软硬件两部分在内的初始化过程。

(3) 系统初始化。该初始化过程以软件初始化为主,主要进行操作系统的初始化。启动代码将嵌入式微处理器的控制权转交给嵌入式操作系统,由操作系统完成余下的初始化操作,包含加载设备驱动程序,建立系统内存区,加载并初始化其他系统软件模块,如网络系统、文件系统等。最后,操作系统创建应用程序环境,并将控制权交给应用程序的入口。

对于不使用操作系统的硬件系统而言,只需要完成前两个部分的工作,而第三个部分则是采用了嵌入式操作系统的硬件设备所需要的。

2) 驱动程序

驱动程序主要是编写和硬件相关的设备驱动,不同的操作系统对于驱动程序都有不同的接口标准,如 Windows CE 的驱动程序就要求按照它的标准编写 10 个接口函数。与硬件相关的驱动程序设计是嵌入式系统开发一个非常关键的环节。

3) 嵌入式操作系统

嵌入式操作系统一般由操作系统内核、文件系统、图形用户接口、网络系统及通用组件模块组成,其中操作系统内核是必备的,其他为可选模块。

操作系统内核是嵌入式应用软件的基础和开发平台。它负责嵌入式系统的全部软硬件资源的分配,任务调度,控制、协调并发活动。相对于一般操作系统而言,它除了具备一般操作系统最基本的功能,如任务调度、同步机制、中断处理、文件功能等外,还有以下特点:

- 强实时性。嵌入式操作系统实时性一般较强,可用于各种设备控制中。
- 统一的接口。提供各种设备驱动接口。
- 操作方便、简单,一般会提供友好的图形用户界面(GUI),追求易学易用。
- 提供网络功能,支持 TCP/IP 协议及其他协议,提供统一的 MAC 访问层接口,为

各种移动计算设备预留接口。

- 强稳定性,弱交互性。嵌入式系统一旦开始运行就不需要用户过多的干预,这就要负责系统管理的内核具有较强的稳定性。嵌入式操作系统的用户接口一般不提供操作命令,它通过系统调用命令向用户程序提供服务。

4) 应用程序

应用程序基于操作系统提供的各种接口函数进行开发,在中低端的嵌入式处理器中,应用程序往往和嵌入式操作系统结合在一起开发,如在 $\mu\text{C}/\text{OS} - \text{II}$ 操作系统的开发中,应用程序和操作系统都在一个工程中,应用程序利用 $\mu\text{C}/\text{OS} - \text{II}$ 提供的服务完成各个进程的通信、同步和定时等操作。而在高端的嵌入式处理器中,应用程序往往独立于操作系统来开发,如 Windows CE 的应用程序开发就可以直接利用 VS2010 等通用计算机上的集成开发环境进行编写,需要访问硬件时,调用硬件的驱动程序接口即可。应用程序一般要使用人机接口界面和网络协议栈等模块。

整体而言,嵌入式系统的组成和开发与传统的计算机系统有很大的不同,具体对比如表 1-1 所列。

表 1-1 嵌入式系统与通用计算机系统的对比

设备名称	嵌入式系统	通用计算机系统
处理器	嵌入式处理器 (ARM、MIPS、Power PC)	通用处理器 (Intel、AMD 等)
内存	SDRAM 芯片	SDRAM 或 DDR 内存
存储设备	Flash 芯片	硬盘
输入设备	触摸屏、按键等	鼠标、键盘、麦克风等
输出设备	小尺寸液晶屏、数码管等	显示器
其他设备	音频芯片、USB 芯片、网卡芯片等	主板集成或外接卡
启动代码	Bootloader 启动,针对不同电路进行移植	主板的 BIOS 启动,无须改动
操作系统	Linux、Windows CE、VxWorks、Andriod 等,需要移植	Windows、Linux 等,无须移植
驱动程序	每个设备驱动针对电路板进行开发或移植,一般不能直接下载使用	操作系统含有大多数驱动程序,或下载直接使用
开发环境	交叉编译	本机编译
协议栈	需要移植	操作系统或第三方提供

1.2 嵌入式处理器

1.2.1 简介

普通个人计算机 (PC) 中的处理器一般是通用处理器,它们的设计非常丰富。例如,现代处理器具有内置的内存管理单元 (MMU),提供内存保护和多任务能力的虚拟存储器等功能模块。在早期,嵌入式系统通常基于通用处理器设计。

近年来,随着处理器制造技术的发展,越来越多的嵌入式系统采用嵌入式处理器设