

普通高等院校嵌入式系统规划教材

Embedded

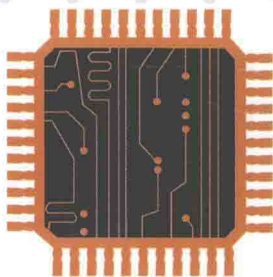
PRINCIPLES AND APPLICATIONS OF EMBEDDED SYSTEMS

嵌入式系统 原理与应用

常华 黄岚 张海燕 编著

Chang Hua Huang Lan Zhang Haiyan

深入解读ARM S3C2440的架构、指令和Linux操作系统下的驱动开发、实例开发
以及Qt/Embedded的移植与应用



CD-ROM

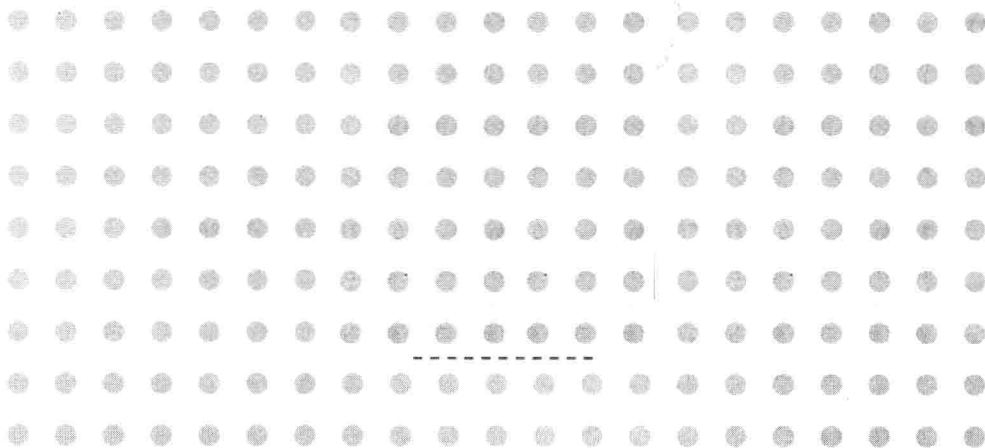
清华大学出版社



PRINCIPLES AND APPLICATIONS OF EMBEDDED SYSTEMS

嵌入式系统 原理与应用

常华 黄岚 张海燕 编著
Chang Hua Huang Lan Zhang Haiyan



清华大学出版社
北京

内 容 简 介

本教材使用通俗易懂的语言,由浅入深地介绍了嵌入式系统的组成、嵌入式微处理器的使用、Linux 开发环境的建立、简单驱动程序的编写、如何在 Linux 操作系统支持下开发嵌入式系统、嵌入式系统图形界面的使用以及如何构建自己的嵌入式系统等内容。书中包含大量短小、简单却可以对理解嵌入式系统工作原理起着重要作用的示例,通过对这些教学示例的演示,可以帮助初学者在较短的时间解除对嵌入式系统的神秘感,对嵌入式系统的组成、工作原理、开发流程有一个较全面的认识。

本教材可以作为具有计算机原理、微机接口技术、C 语言程序设计、单片机原理等课程基础的,初次接触嵌入式系统的本科生、研究生、教师以及对嵌入式系统感兴趣的科技工作者参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

嵌入式系统原理与应用/常华,黄岚,张海燕编著.--北京:清华大学出版社,2013.1

普通高等院校嵌入式系统规划教材

ISBN 978-7-302-29250-0

I. ①嵌… II. ①常… ②黄… ③张… III. ①微型计算机—系统设计—高等学校—教材
IV. ①TP360.21

中国版本图书馆 CIP 数据核字(2012)第 146363 号

责任编辑:盛东亮

封面设计:李召霞

责任校对:时翠兰

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:三河市君旺印装厂

装 订 者:三河市新茂装订有限公司

经 销:全国新华书店

开 本:186mm×240mm 印 张:27.75 字 数:624 千字

附光盘 1 张

版 次:2013 年 1 月第 1 版

印 次:2013 年 1 月第 1 次印刷

印 数:1~3000

定 价:49.50 元

随着嵌入式系统应用的普及,相关专业的学生迫切希望在较短的时间内较为全面地掌握相关知识。由于嵌入式系统所涉及的知识面非常宽泛,如何能在有限的时间内让学生掌握嵌入式系统的原理和开发过程中的关键点,并能在此基础上快速地将所学知识付诸工程实践,是编写本教材时考虑最多的问题。下面就教材的组织、使用方法等情况做些说明,权当本书的前言。

1. 教材的组织

嵌入式系统原理及应用这门课已经成为许多学校相关专业的必修课程。对于这门课如何讲,讲些什么,各个学校和任课老师取舍各有不同。

市面上可以看到各种版本的关于嵌入式系统原理及应用方面的著作。主要可以分为下面几类:

一类是针对嵌入式系统开发过程中某一环节的专著。如某款 CPU 的原理与开发、Linux 驱动程序、嵌入式系统的图形界面、实时操作系统、Linux 编程等。这类专著,对深入了解对应环节的内幕、深化对嵌入式系统的理解起着至关重要的作用。

一类是将上述的各个环节融会在一起的,类似手册性质的著作,如各类手册大全、开发详解等。这类著作详尽地介绍了各种相关知识、各类工具软件的安装使用方法、大量应用程序示例等。这类著作经常作为开发人员在开发过程中工作案头的伴侣。

还有一类是针对某一种应用,在介绍了相关基础知识之后,完整介绍一个较复杂的应用系统软硬件设计和调试过程的著作。诸如无线网络的开发、某种专用设备的开发与应用等。这类著作对相关行业开发人员而言大有裨益。

然而,笔者认为这几类著作是不宜作为教材的。特别是对相关专业的本科生、研究生等初次接触到嵌入式系统的学生而言,要么太专、要么教学时间不够。

对于初次接触嵌入式系统的学生而言,如何能在单位时间内向学生展示嵌入式系统原理及应用的全貌,这个“全貌”要能够涵盖嵌入式系统原理、嵌入式系统开发过程的各个关键点。如何选择教学内容,举什么样的例子才能达到上述目标,是我们组织教材时重点考虑的问题。

为此,我们选择了以下几个部分作为本教材的主要内容:将嵌入式微处理器 S3C2440 作为高档单片机的开发与应用,基于 Linux 操作系统的嵌入式系统开发环境的建立, Linux 操作系统下简单设备驱动程序的编写与应用, Linux 操作系统下关于信号、进程、线程的编程,图形界面系统 Qt-Embedded 的移植与应用,嵌入式 Linux 操作系统的移植过程。学生在了解了这些嵌入式系统的相关基础之后,就可以结合工程实践,借助各类相关专著和指导,在嵌入式系统的“海洋”里遨游了。

2. 教材面向的对象

本教材面向的对象是具有计算机原理、微机接口技术、C 语言程序设计、单片机原理等课程基础的,初次接触嵌入式系统的相关专业的本科生、研究生、教师,以及对嵌入式系统感兴趣的科技工作者参考。如果读者具有数据结构、软件工程、操作系统等课程的基础,入门会更快。

3. 教材特点

本教材以嵌入式微处理器 S3C2440 为蓝本,从嵌入式系统原理教学的角度出发,面向应用,用最简单的示例程序去揭示嵌入式系统的工作原理,尽可能地降低对某类具体开发板硬件系统的依赖,使学习过本教材的学生可以根据自己的硬件平台快速上手,针对自己的课题或项目展开研发工作。

本教材以授课的语言,用通俗易懂的方式描述嵌入式系统原理及开发过程中的主要问题,尽可能地避开生僻的专业名词。在嵌入式系统中,几乎涉及了与计算机相关的各个领域的内容,涉及众多相关的名词、术语,而学习这门课的学生未必学过所有这些课程,例如:数据结构、软件工程、操作系统等。因此,如何用通俗的语言、简单的示例去让学生理解基本概念,也是本教材在编写时特别注意的地方。

本教材中包含了嵌入式系统开发过程中所涉及各个关键点,对每个关键点都给出了最简单的、原理性的程序示例。通过这些示例,读者可以方便、快捷地理解各个关键点的关键所在。

本教材不过多地重复在计算机原理、微机接口技术、C 语言程序设计、单片机原理等基础课中已经讲过的内容。必须重复的东西,以列表等形式给予简要的说明。

本教材不过多地涉及属于其他课程的内容,例如,Qt 编程、Linux 操作系统等课程的内容,而是用通俗的语言(以避免没有学过这些课程的读者有过强的生疏感)去描述这些相关内容是如何在嵌入式系统中应用的。

为了节约篇幅,本教材不过多地描述那些只要面对具体装置、具体软件就可以了解的简单操作细节,具体操作细节在课堂演示中完成。

4. 教材的使用方法和课时安排

使用本教材教学时,建议采用 40 课时课堂教学和 8 学时实验教学的教学方案。

40 课时课堂教学课时分配如下:

第 1 章 嵌入式系统概述	——1 学时
第 2 章 将 ARM 微处理器 S3C2440 作为高档单片机的开发与应用	——14 学时
第 3 章 Linux 操作系统下嵌入式系统开发环境的建立	——3 学时
第 4 章 Linux 操作系统下设备驱动程序的编写与应用	——10 学时
第 5 章 Linux 操作系统下关于信号、进程、线程的编程	——6 学时
第 6 章 图形界面系统 Qt/Embedded 的移植与应用	——4 学时
第 7 章 嵌入式 Linux 操作系统的移植过程	——2 学时

课堂教学可通过讲解、演示(包括工具软件的使用、示例程序的演示)交叉进行。第 2 章的 2.20 一节可放在第 5 章之后讲解,此时学生已经具备了有关信号、进程及进程间通信的知识和相关示例程序的演示,对这些概念有了感性认识,此时再介绍 $\mu\text{C}/\text{OS-II}$ 的相关内容,学生接受起来会容易得多。

关于实验教学:

实验教学可根据所具备的实验条件灵活安排。例如,可以设计一个配乐电子相框的综合实验,该实验可以使用在 ADS 环境下实现,可在 Linux 操作系统下以单进程、多进程、多线程、利用图形界面等各种方式完成,也可以设计成在 $\mu\text{C}/\text{OS-II}$ 操作系统的支持下以多任务的工作方式完成,配乐与照片之间的配合可以采用固定式、随机式、可选式等多种方式。根据学生水平的差异选做。

5. 教材配套光盘的使用

配套光盘中包含以下内容:

- (1) 3 种开发板电路原理图——用于认识不同开发板之间的差异;
- (2) 各类软件使用手册——各类工具软件的使用说明;
- (3) 示例程序——与教材配用的示例程序(硬件平台: mini2440);
- (4) 相关硬件手册——开发板主要器件手册。

本教材由黄岚老师负责编写第 1 章和第 2 章的 2.20 节,由常华老师负责编写第 2 章、第 3 章、第 4 章、第 5 章,由张海燕老师编写第 6 章和第 7 章。最终由常华老师负责完成全书的文字审阅及例题的编制、搜集、注释及校核。

在本教材的编写过程中,我非常感谢我的团队,正是由于团队的通力合作才有了今天最终的教材文稿,团队成员认真负责的精神和一丝不苟的工作态度给我留下了深刻印象。在

本教材的编写过程中,还要感谢那些不知名的网友,是他们在专业论坛上的问答、想法拓宽了我们的思路、给予了我们启发,使我们对嵌入式系统的认知进一步深化。

鉴于嵌入式系统所涉及的知识面非常宽广,不可能在一本教材中全部囊括,加之编者相关知识水平的限制,教材中不免会出现这样那样的错误,还请广大读者给予批评指正,联系方式如下:

作者 changhua@cau. edu. cn

编辑 shengdl@tup. tsinghua. edu. cn

编著者

第 1 章 嵌入式系统概述	1
1.1 嵌入式系统的定义	1
1.2 嵌入式系统的发展	2
1.3 嵌入式系统的组成	3
1.4 嵌入式系统的分类	4
1.5 嵌入式系统的开发流程	8
1.6 如何学习嵌入式系统	10
第 2 章 将 ARM 微处理器 S3C2440 作为高档单片机的开发与应用	12
2.1 了解 ARM 微处理器 S3C2440	12
2.1.1 S3C2440 的硬件资源配置	12
2.1.2 S3C2410 与 S3C2440 的主要区别	14
2.1.3 认识一个由 S3C2440 微处理器组成的嵌入式系统	14
2.2 S3C2440 对存储器和寄存器的组织	15
2.2.1 S3C2440 对存储器的组织	16
2.2.2 S3C2440 对寄存器的组织	17
2.2.3 S3C2440 对特殊功能寄存器 SFR 的组织	18
2.3 S3C2440 的启动方式	19
2.3.1 Nand Flash 与 Nor Flash	19
2.3.2 S3C2440 的 Nor Flash 启动	20
2.3.3 S3C2440 的 Nand Flash 启动	20
2.4 ARM 指令系统	20
2.4.1 ARM 指令系统	21
2.4.2 ARM 指令系统的寻址方式	23
2.4.3 ARM 指令系统的分类	27
2.4.4 ARM 指令集和 Thumb 指令集的关系	29
2.5 ARM 伪指令系统	30

2.5.1	符号定义伪指令	31
2.5.2	数据定义伪指令	31
2.5.3	汇编控制伪指令	32
2.5.4	宏定义伪指令	33
2.5.5	段定义伪指令	33
2.5.6	杂项伪指令	34
2.5.7	ARM 伪指令	35
2.5.8	ARM 连接器的保留字	35
2.5.9	ARM 汇编器的内置变量	36
2.6	在 ADS 开发环境下开发 S3C2440	37
2.6.1	建立 ADS 开发环境	37
2.6.2	编写一个汇编语言框架程序	37
2.6.3	编写一个在 Boot RAM 中运行的 LED 驱动程序	43
2.6.4	制作一个可以自动运行的 LED 闪烁装置	45
2.7	S3C2440 对电源和时钟系统的管理	45
2.7.1	认识 S3C2440 时钟系统	45
2.7.2	系统时钟设置步骤	47
2.7.3	电源和时钟系统管理方案	48
2.8	S3C2440 对 SDRAM 的初始化	51
2.8.1	初始化 SDRAM 存储器所需的控制寄存器	52
2.8.2	代码搬移	55
2.9	用 C 语言对 S3C2440 编程	56
2.9.1	在 ADS 环境下直接使用 C 语言编程	57
2.9.2	由汇编语言转入 C 语言	58
2.9.3	汇编语言与 C 语言无参函数的相互调用	59
2.9.4	汇编语言与 C 语言带参函数的相互调用	61
2.9.5	在 C 语言中嵌入汇编语言	63
2.10	S3C2440 的 Nand Flash 控制器	64
2.10.1	Nand Flash 控制器的初始化	65
2.10.2	Nand Flash 读操作流程	70
2.10.3	Nand Flash 读操作编程示例	71
2.11	S3C2440 对中断的组织	74
2.11.1	S3C2440 的工作模式	74
2.11.2	S3C2440 对中断(异常)的响应	76
2.11.3	S3C2440 对中断源的组织	78
2.11.4	S3C2440 中断控制器的初始化	80

2.11.5	汇编语言的中断编程框架示例	83
2.11.6	C 语言的中断编程框架示例	87
2.12	S3C2440 TFT 型液晶显示器的使用	92
2.12.1	S3C2440 内置 LCD 控制器介绍	92
2.12.2	LCD 初始化编程示例	98
2.13	S3C2440 的 ADC 和电阻式触摸屏的使用	100
2.13.1	四线电阻式触摸屏的工作原理	101
2.13.2	ADC 和触摸屏接口使用方法	102
2.13.3	ADC 和触摸屏编程示例	104
2.14	S3C2440 中的各类定时器的使用	106
2.14.1	看门狗 Watch dog 的使用	106
2.14.2	实时时钟——RTC 的使用	108
2.14.3	S3C2440 定时器与 PWM 接口的使用	112
2.15	S3C2440 串行接口(UART)的使用	118
2.15.1	UART 控制器的结构	118
2.15.2	串行口的初始化	119
2.15.3	串行口应用举例	122
2.16	S3C2440 I ² C 接口的使用	124
2.16.1	AT24C08 EEPROM 操作	124
2.16.2	I ² C 接口的初始化	128
2.16.3	I ² C 接口的编程示例	128
2.17	S3C2440 DMA 控制器的使用	132
2.17.1	DMA 控制器的设置	132
2.17.2	DMA 控制器编程示例	135
2.18	S3C2440 的内存管理单元 MMU	137
2.18.1	MMU 的基本概念	137
2.18.2	MMU 的映射原理	140
2.18.3	S3C2440 的 MMU 设置	141
2.18.4	MMU 段映射编程示例	146
2.19	用 Scatter 文件组织程序	149
2.19.1	什么时候使用 Scatter 文件	150
2.19.2	Scatter 文件的编写和应用	150
2.20	在 S3C2440 上运行 μ C/OS- II 嵌入式实时操作系统	155
2.20.1	RTOS 的基本概念	155
2.20.2	RTOS 中的任务管理	156
2.20.3	RTOS 的任务状态与任务调度	159

2.20.4	RTOS 中的基本任务操作	160
2.20.5	RTOS 下的任务设计	161
2.20.6	嵌入式实时操作系统 $\mu\text{C}/\text{OS-II}$	164
2.20.7	将 $\mu\text{C}/\text{OS-II}$ 移植到 S3C2440 开发板	165
2.20.8	基于 $\mu\text{C}/\text{OS-II}$ 的多任务程序设计示例	169
第 3 章	Linux 操作系统下嵌入式系统开发环境的建立	174
3.1	嵌入式系统为什么要配用操作系统	174
3.2	认识一个配有 Linux 操作系统的 S3C2440 嵌入式系统开发板	175
3.3	建立嵌入式系统开发环境	177
3.3.1	建立嵌入式系统开发环境的步骤	177
3.3.2	嵌入式系统开发过程中经常使用的 Linux 操作系统命令	181
3.4	嵌入式系统的开发过程	182
3.4.1	vi 编辑器的使用	182
3.4.2	对源程序的编译、交叉编译及执行	185
3.4.3	环境变量的设置	191
3.4.4	gdb 调试工具的使用	193
3.5	shell 程序的编写	198
3.5.1	shell 是什么	198
3.5.2	一个简单的 shell 程序	199
3.5.3	shell 程序的用户变量定义与引用	200
3.5.4	shell 程序的常用系统变量	201
3.5.5	shell 程序的算术和逻辑运算	202
3.5.6	shell 程序的流程控制	204
3.5.7	shell 编程要点	208
3.5.8	shell 程序的调试	209
3.6	Makefile 文件的编写	211
3.6.1	什么是 Makefile 文件	211
3.6.2	Makefile 文件的结构	214
3.6.3	Makefile 文件变量的定义、赋值与引用	216
3.6.4	Makefile 文件中内嵌命令(函数)的使用	219
3.6.5	Makefile 文件中两种常用规则的使用	220
3.6.6	Makefile 文件的几种特殊的编程方法	222
3.6.7	make 命令的执行	223
3.6.8	Makefile 文件框架的自动生成	224
3.7	SourceInsight 程序编辑/浏览器使用方法	226

第 4 章 Linux 操作系统下设备驱动程序的编写与应用	228
4.1 什么是设备驱动程序	229
4.2 设备驱动程序初探	230
4.2.1 一个简单的设备驱动程序	231
4.2.2 设备驱动程序与应用程序的区别	233
4.2.3 带参数设备驱动程序的编写与加载	234
4.3 简单字符设备驱动程序框架	236
4.3.1 静态分配设备号的简单字符设备驱动程序框架	237
4.3.2 动态分配设备号的简单字符设备驱动程序框架	242
4.3.3 应用程序与简单字符设备驱动程序的数据交换	245
4.3.4 简单字符设备驱动程序编写和测试步骤	249
4.4 嵌入式系统简单字符设备驱动程序的编写	250
4.4.1 设备驱动程序交叉编译环境的建立	250
4.4.2 一个简单字符设备驱动程序	252
4.4.3 将设备驱动程序编写得更“专业”一些	255
4.4.4 带有中断功能的简单字符设备驱动程序	259
4.4.5 利用内存映射在应用层直接对外设编程	266
4.5 多个设备号驱动程序的编写	269
4.6 对简单字符设备驱动程序的基本印象	273
4.7 Linux 操作系统下字符设备驱动程序的应用	273
4.7.1 LCD 设备应用编程	274
4.7.2 音频设备应用编程	278
4.7.3 串口设备应用编程	285
4.7.4 USB 字符设备应用编程	287
4.7.5 触摸屏设备应用编程	292
4.7.6 触摸屏接口程序 tslib 的移植与应用	293
4.7.7 USB 摄像头设备应用编程	295
4.7.8 使用 camera 接口应用编程	297
4.7.9 实时时钟(RTC)应用编程	301
4.7.10 开发板上简单硬件装置的测试程序	304
4.8 块设备驱动程序的应用	306
4.8.1 SD 卡的使用	306
4.8.2 U 盘的使用	307
4.9 将设备驱动程序静态加载	307

第 5 章 Linux 操作系统下关于信号、进程、线程的编程	311
5.1 几个基本概念的解释	311
5.1.1 任务、进程、线程	311
5.1.2 进程间的通信	313
5.1.3 Linux 下关于进程与线程的基本操作命令	314
5.1.4 关于本章示例程序的说明	315
5.2 进程的创建及多进程编程	317
5.2.1 相关函数	317
5.2.2 创建进程编程示例	320
5.2.3 exec 函数族的使用	321
5.2.4 exec 函数族编程示例	322
5.3 信号的使用	324
5.3.1 什么是信号	324
5.3.2 进程对信号的响应	326
5.3.3 相关函数	327
5.4 进程间的通信	330
5.4.1 信号通信	330
5.4.2 管道通信	333
5.4.3 消息队列(报文)通信	339
5.4.4 内存(文件)映射与共享内存通信	343
5.4.5 信号量(信号灯)通信	351
5.4.6 套接字通信	356
5.5 线程的创建及多线程编程	368
5.5.1 相关函数	368
5.5.2 创建线程编程示例	370
5.5.3 互斥锁的使用	371
第 6 章 图形界面系统 Qt/Embedded 的移植与应用	375
6.1 Qt/Embedded 的移植	376
6.1.1 Qt/Embedded 移植前的准备	376
6.1.2 Qt/Embedded-4.6.3 的移植	377
6.2 Qt/Embedded-4.6.3 的测试	385
6.2.1 对 X86 版的测试	385
6.2.2 对 Qte 版的测试	386
6.2.3 对 ARM 版的测试	386
6.3 Qt/Embedded-4.6.3 应用程序的开发过程	387

6.3.1	编写一个 X86 版的应用程序	387
6.3.2	编写一个 Qte 版的应用程序	389
6.3.3	编写一个 ARM 版的应用程序	390
6.4	Qt/Embedded-4.6.3 与开发板外部设备之间的接口	391
第 7 章	嵌入式 Linux 操作系统的移植过程	395
7.1	嵌入式 Linux 操作系统的组成	396
7.2	BootLoader——引导加载程序	397
7.2.1	U-Boot 的基本概念	399
7.2.2	U-Boot 的目录结构	400
7.2.3	U-Boot 的启动流程	401
7.2.4	U-Boot 与 Linux 内核的关系	401
7.2.5	U-Boot 的常用命令	403
7.2.6	U-Boot 的移植过程	404
7.3	Linux 内核的配置(裁剪)与移植	407
7.3.1	Linux 操作系统内核特点及源码结构	407
7.3.2	Linux 内核配置(裁剪)过程	409
7.3.3	Linux 内核配置(裁剪)系统的组织	411
7.3.4	Linux 内核的 Makefile 文件	416
7.3.5	Linux 内核的编译	420
7.3.6	Linux 内核的移植过程	420
7.4	根文件系统的建立	421
7.4.1	嵌入式系统的文件系统	421
7.4.2	嵌入式系统的根文件系统	422
7.4.3	BusyBox 工具	422
7.4.4	使用 BusyBox 构建根文件系统过程	423
附录	配书光盘文件目录	428
参考文献		429

嵌入式系统概述

随着计算机技术的发展和网络的普及,嵌入式系统的应用几乎已经渗透到国民经济的各个领域——工业、农业、建筑业、运输业、电信业、商业、对外贸易、服务业、城市公用事业以及国防科技、社会管理领域等。嵌入式系统的应用实例不胜枚举,例如通信设备、智能仪器/仪表、消费类电子产品、检测/控制装置、监控装置、商品物流管理设备……。在日常生活中,人们几乎天天都在和嵌入式系统打交道,嵌入式系统在生产与生活中起到了举足轻重的作用。嵌入式系统的应用示例如图 1-1 所示。

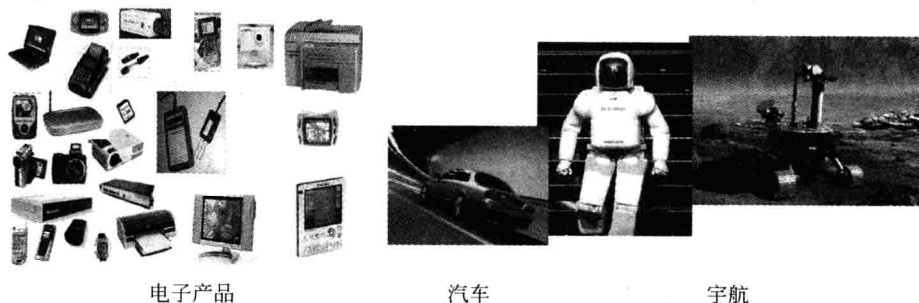


图 1-1 嵌入式系统的应用

据不完全统计,全球每年生产约 20 亿片微处理器芯片,其中有 95% 应用在各类嵌入式系统中,甚至一台通用计算机所配置的外部设备就包含了 5~10 个嵌入式微处理器,例如键盘、显示卡、网卡、打印机等。可以说“普适计算”(pervasive computing)的时代已经来临——人们能够在任何时间、任何地点、以任何方式进行信息的获取与处理。

1.1 嵌入式系统的定义

IEEE 对于嵌入式系统的定义如下:

“An Embedded system is the devices used to control, monitor, or assist the operation of equipment, machinery or plants.”——嵌入式系统是“用于控制、监视或者辅助操作机器和设备的装置”。

对于嵌入式系统,国内文献普遍这样概括: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。

换一种更通俗的说法:凡是与应用产品结合在一起的、具有专门用途的、带有特定接口的、配有专用软件的、满足特定要求的微处理器系统都可以称做嵌入式系统。

1.2 嵌入式系统的发展

自从微处理器问世,工程师们就开始了利用微处理器进行专用设备、仪器、家用电器控制装置等方面的研制工作。这些研制工作本身就是一个典型的嵌入式系统开发实例,只是当时嵌入式概念没有明确地提出来。

几十年来,随着计算机软、硬件技术的进步,嵌入式系统从理论到实践不断完善,嵌入式系统已经逐渐形成了相对独立的学科分支体系。

嵌入式系统的发展大致经历了3个阶段。

第一阶段,以单片机为核心的嵌入式系统。在这个阶段,系统的核心硬件为单片机,系统软件设计采用无操作系统的监控程序设计方法。这类系统只是为了实现某个控制功能,使用一个简单的循环控制或者采用前后台软件结构,完成根据外界请求进行控制、处理的功能。

第二阶段,以嵌入式微处理器为核心,将嵌入式微处理器与嵌入式操作系统相结合。当嵌入式系统的应用越来越复杂、使用的范围越来越广泛的时候,面对众多需要管理的任务,以简单单片机为核心的嵌入式系统就显得力不从心了。嵌入式微处理器的问世,特别是引入嵌入式操作系统后,用户可将大量系统管理任务交由嵌入式操作系统打理,从而可使嵌入式系统的开发者腾出大量时间去完善针对应用对象的软硬件设计,这就大大提高了开发效率,也大幅提高了系统的性能和可靠性。

第三阶段,以嵌入式片上系统为核心构成的嵌入式系统,系统体积更小、能耗更低、可靠性更高、开发周期更短。与此同时,工程师也在大力开发基于互联网的嵌入式应用系统,完善嵌入式操作系统对网络的支持,使用户可以在嵌入式系统上方便地实现基于互联网的远程数据采集、远程监视、远程数据文件传输等功能。

随着嵌入式服务器、嵌入式网关和嵌入式路由器的发展,嵌入式系统正在进入嵌入式Internet时代,我们目前由嵌入式系统构成的互联网“终端”将成为互联网中的一个节点、一个服务器。这些嵌入式服务器将各个嵌入式系统收集到的各种物理信息、生物信息通过Internet网自动、实时、方便、简单地提供给需要这些信息的对象。届时,世界会变得更加智能化和人性化。

更准确地说,这三个发展阶段已经演化成三个相互融合的发展方向,分别在各自不同的应用领域扮演着重要的角色。经过几十年的发展,嵌入式系统已经在很大程度改变了人们的生产、生活、工作、交流和娱乐方式,而且这种改变还在加速。

1.3 嵌入式系统的组成

与熟知的计算机系统一样,嵌入式系统作为一个“专用计算机系统”,也是由硬件系统和软件系统两大部分组成的。硬件系统由嵌入式微处理器和外围硬件设备组成,软件系统由嵌入式操作系统与用户应用程序共同组成。嵌入式系统结构如图 1-2 所示。

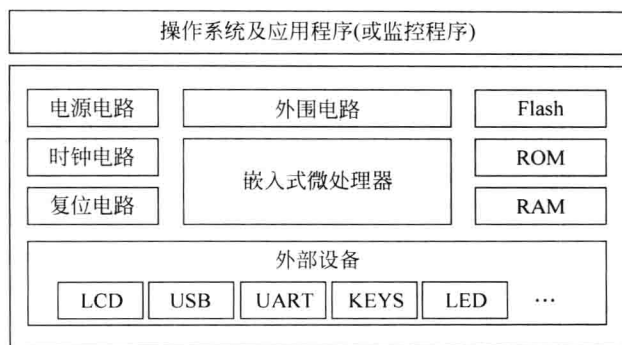


图 1-2 嵌入式系统结构

1. 嵌入式系统的硬件组成

嵌入式系统硬件组成部分的核心部件就是嵌入式微处理器。嵌入式微处理器与通用型微处理器的最大不同就是嵌入式微处理器大多工作在为特定用户群设计的系统中,通常都具有功耗低、体积小、集成度高等特点,能够把由通用型微处理器所组成的系统中许多由特定板卡完成的任务集成在芯片内部形成特定的外部设备接口。

目前,全世界嵌入式微处理器的品种已经超过 1000 多种,流行的体系结构多达 30 多个。嵌入式微处理器的寻址空间也从 64KB 到 2GB 不等,其运算速度可以从 0.1MIPS ~ 2000MIPS 甚至更高。

一般来说,可以把嵌入式微处理器分成 4 类:

1) EMPU(Embedded MicroProcessor Unit)嵌入式微处理器

嵌入式微处理器可以看成是通用计算机中微处理器的微缩版。相对于通用型微处理器而言,嵌入式微处理器具有体积小、功耗少、成本低的优点。

2) EMCU(Embedded MicroController Unit)嵌入式微控制器

在嵌入式微处理器的基础上将众多的外部设备接口集成到一块芯片中,可以以单片或以单片辅以外部大容量存储器的形式组成嵌入式系统,从而大幅度降低了系统组成成本,提高系统的可靠性。

3) EDSP(Embedded Digital Signal Processor)嵌入式信号处理器

嵌入式信号处理器是专门用于信号处理方面的处理器,在系统结构和指令算法方面进