

“十二五”高等院校规划教材

ARM 9

嵌入式系统设计

——基于S3C2410与Linux
(第2版)



徐英慧 马忠梅 王磊 王琳 编著

“十二五”高等院校规划教材

ARM9 嵌入式系统设计

——基于 S3C2410 与 Linux

(第2版)

徐英慧 马忠梅 编著
王 磊 王 琳

北京航空航天大学出版社

内 容 简 介

针对在嵌入式市场上颇具竞争力的 ARM9 处理器——S3C2410 和开放源码的 Linux 操作系统,讲述嵌入式系统的概念、软硬件的开发和调试手段、嵌入式 Linux 驱动程序和应用程序的开发以及图形用户界面 MiniGUI 的移植和应用。第 2 版将 ADS1.2 内容替换为 RealView MDK 和 IAR EWARM 的内容,同时开发平台也升级为 EL-ARM-860。

本书的特点是集嵌入式系统开发的理论知识和实验教学于一体,并结合北京精仪达盛科技有限公司的开发板,给出了大量实例。

本书可作为高等院校嵌入式系统课程的教材,也可作为对嵌入式系统开发感兴趣的读者的入门教材,同时还可以作为从事 ARM 嵌入式系统应用开发工程师的参考书。

图书在版编目(CIP)数据

ARM9 嵌入式系统设计:基于 S3C2410 与 Linux / 徐英慧等编著. —2 版. —北京:北京航空航天大学出版社, 2010.9

ISBN 978-7-5124-0194-5

I. ①A… II. ①徐… III. ①微处理器 x 系统设计—软件开发 IV. ①TP332

中国版本图书馆 CIP 数据核字(2010)第 161978 号

版权所有,侵权必究。

ARM9 嵌入式系统设计 ——基于 S3C2410 与 Linux(第 2 版)

徐英慧 马忠梅 编著
王磊 王琳
责任编辑 张楠 王松

*

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:emsbook@gmail.com 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 mm×1 092 mm 1/16 印张:23.5 字数:602 千字

2010 年 9 月第 2 版 2010 年 9 月第 5 次印刷 印数:20 001~25 000 册

ISBN 978-7-5124-0194-5 定价:39.00 元

第2版前言

本书是2007年9月出版的《ARM9 嵌入式系统设计——基于 S3C2410 与 Linux》的第2版。第2版对第1版的内容进行了全面的整理、优化和修改。第1版中介绍的 ARM 开发环境主要是 ADS1.2,第2版则主要介绍了目前流行的 RealView MDK 和 IAR EWARM。第1版配套的开发平台是北京精仪达盛有限公司提供的 EL-ARM-830,第2版则升级为 EL-ARM-860,对相应的示例也进行了修整。

当前,嵌入式技术的应用越来越广泛,从航天科技到民用产品,嵌入式产品的身影无处不在,而这些嵌入式产品的核心——处理器决定了产品的市场和性能。在32位嵌入式处理器市场中,ARM 处理器占有很大的份额。ARM 不仅是一个公司、一种技术,也是一种经营理念,即由 ARM 公司提供核心技术,只出售芯片中的 IP 授权,采取了别具一格的“Chipless 模式”(无芯片的芯片企业),不参与生产,而是由合作厂商去生产具体的芯片和产品。

现在由于存储空间等原因,在嵌入式芯片上编程有较大的困难,选取合适的平台就显得很重要。Linux 自出现以来,得到了迅速的发展。Linux 是开放源码的操作系统,吸引着全世界的程序员参与到发展和完善的工作中来,所以 Linux 保持了稳定而且卓越的性能。Linux 在服务器领域已经占有很大的份额,在图形界面方面也不输于 Windows。由于源码可以修改、移植,Linux 在嵌入式领域中的应用也越来越广。选用 Linux 作为平台,可以根据具体需要自由地裁减源码,打造适合目标平台的环境,编写最有效率的应用程序。

可以预见,ARM 与 Linux 在未来会更加壮大,在嵌入式产品市场上会占有越来越大的份额。在这种形势下,学习研究 ARM 与 Linux 非常有必要,这也是本书编写的目的。在新的科技浪潮来临之际,我国正全力迎接机遇和挑战,嵌入式领域方兴未艾,Linux 也越来越成熟,我们需要掌握更新的知识,实现自我价值,为祖国贡献力量!

S3C2410 是 ARM9 系列中非常优秀的一款处理器,应用广泛。本书主要介绍 ARM (S3C2410)和嵌入式 Linux 的基本知识及具体应用。本书共分8章,第1章和第2章介绍嵌入式系统的基本概念和开发方法,第3章到第5章介绍 ARM 的基本知识及系统硬件的设计,第6章和第7章介绍 Linux 的移植和基本应用,第8章介绍图形用户界面 MiniGUI 的移植和应用。

第1章:介绍嵌入式系统的基本概念。包括嵌入式系统的概念、特点及应用,读者可以由本章了解嵌入式系统的基础知识,掌握嵌入式的发展动向。

第2章:介绍嵌入式软件的开发过程和调试手段。读者需要注意嵌入式软件与普通软件

在开发和调试上的区别。

第3章:介绍 ARM 的体系结构。主要包括 ARM 体系结构的特点,ARM 处理器的结构、分类,ARM 寄存器及存储器说明。

第4章:介绍 ARM 系统硬件设计的基础知识。包括 ADS 集成开发环境、对 ARM 的汇编程序设计和混合编程以及 ARM 硬件启动程序设计。

第5章:介绍 S3C2410 系统的硬件设计。分别介绍了 I/O 口、中断、DMA、UART、A/D 接口、键盘、LCD、触摸屏、音频及 USB 设备的硬件设计要点。

第6章:介绍 Linux 的基础知识。包括内核结构(进程调度、内存管理、虚拟文件、进程通信、网络接口)、设备管理以及 Linux 基本命令。

第7章:介绍 Linux 的软件设计。包括 BootLoader 引导程序、Linux 的移植、驱动程序及应用程序的开发。

第8章:介绍 MiniGUI 在 Linux 和 S3C2410 上的使用。包括 MiniGUI 在 Linux 下运行环境的建立过程和 MiniGUI 移植到 S3C2410 上的过程。此外还对 QVFB 和 FrameBuffer 做了简单介绍。

本书可作为高等院校嵌入式系统课程的教材*,也可以作为对嵌入式系统开发感兴趣的读者的入门教材,同时还可以作为从事 ARM 嵌入式系统应用开发工程师的参考书。

本书由北京机械工业学院和北京理工大学的老师及学生合作完成。北京理工大学的马忠梅老师负责从第1章到第3章第5节的编写;北京机械工业学院的徐英慧老师负责全书的组织工作,并负责从第3章第6节到第5章,以及第8章的编写;北京机械工业学院研究生王磊同学负责第6章的编写;北京机械工业学院研究生王琳同学负责第7章的编写,并做了大量实践工作。

本书在编写过程中得到了许多人的支持和帮助,在此表示特别感谢。感谢北京精仪达盛科技有限公司的毕才术、阳海涛、肖奕之、胡芳、石燕智,他们提供了 ARM9 开发平台,为本书的完成提供了良好的测试环境。感谢北京机械工业学院的杨根兴老师,感谢他一直以来在嵌入式领域对我们的指导,并给我们提供了嵌入式系统开发的实战机会。感谢北京机械工业学院计算中心的周长胜、刘梅彦、林乐荣和黄宏博等各位老师,感谢他们在本书的书写过程中给予的支持和帮助以及一直以来对我们的关心和爱护。感谢王晖、崔建武、郝晓、王山平、侯莉茹和吴德天,感谢他们协助我们对书中的实例进行了验证并对本书做了大量的校对工作。在此,对所有帮助过我们的人真诚地说声“谢谢”。

由于作者的水平有限,书中的错误在所难免,恳请广大读者批评指正。

作者

2010年8月

* 本教材配有教学课件,有需要的教师,请联系 emsbook@gmail.com 索取。



第1章 嵌入式系统基础..... 1

- 1.1 嵌入式系统概念 1
 - 1.1.1 嵌入式系统的定义 1
 - 1.1.2 嵌入式系统的组成 1
 - 1.1.3 嵌入式系统的特点 4
 - 1.1.4 嵌入式系统的应用 5
 - 1.1.5 实时系统 6
- 1.2 嵌入式处理器 8
 - 1.2.1 嵌入式处理器分类 8
 - 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 操作系统的概念和分类 12
 - 1.3.2 实时操作系统 16
 - 1.3.3 常见的嵌入式操作系统 18
- 1.4 实时操作系统的内核 22
 - 1.4.1 任务管理 22
 - 1.4.2 任务间的通信和同步 28
 - 1.4.3 存储器管理 30
 - 1.4.4 定时器和中断管理 30
- 1.5 嵌入式技术发展现状及趋势 31
- 1.6 习 题 32

第2章 嵌入式系统开发过程 33

- 2.1 嵌入式软件开发的特点 33

- 2.2 嵌入式软件的开发流程 35
- 2.3 嵌入式系统的调试 40
- 2.4 板级支持包 43
- 2.5 习 题 45

第3章 ARM 体系结构 46

- 3.1 ARM 体系结构概述 46
 - 3.1.1 ARM 体系结构的特点 46
 - 3.1.2 ARM 处理器结构 49
 - 3.1.3 ARM 处理器内核 60
 - 3.1.4 ARM 处理器核 65
- 3.2 编程模型 67
 - 3.2.1 数据类型 67
 - 3.2.2 处理器模式 67
 - 3.2.3 处理器工作状态 68
 - 3.2.4 寄存器组织 68
 - 3.2.5 异 常 72
 - 3.2.6 存储器和存储器映射 I/O 74
- 3.3 ARM 基本寻址方式 76
- 3.4 ARM 指令集 80
 - 3.4.1 条件执行 80
 - 3.4.2 指令分类说明 80
 - 3.4.3 Thumb 指令集与 ARM 指令集的区别 97
 - 3.4.4 Thumb 指令分类说明 99
- 3.5 ARM9 与 ARM7 的比较 103
- 3.6 ARM9TDMI 内核 106
 - 3.6.1 ARM9TDMI 编程模型 ... 106

3.6.2 ARM9TDMI 存储器接口	108	4.4 基于 ARM 的 C 语言与汇编语言混合编程	185
3.7 ARM920T 核	114	4.4.1 C 语言与汇编语言混合编程应遵守的规则	185
3.7.1 ARM920T 编程模型	115	4.4.2 汇编程序调用 C 程序的方法	186
3.7.2 存储器管理单元(MMU)	126	4.4.3 C 程序调用汇编程序的方法	187
3.7.3 Cache、写缓冲和物理地址 TAG RAM	139	4.4.4 C 程序中内嵌汇编语句	187
3.7.4 时钟模式	150	4.4.5 基于 ARM 的 C 语言与汇编语言混合编程举例	188
3.7.5 总线接口单元	152	4.5 印制电路板制作简介	188
3.8 习 题	159	4.5.1 印制电路板设计软件——Protel	189
第 4 章 ARM 系统硬件设计基础	161	4.5.2 单面板与多层板	189
4.1 ARM 开发环境简介	161	4.5.3 印制电路板设计的注意事项	190
4.1.1 RealView MDK 开发工具简介	161	4.6 习 题	191
4.1.2 IAR EWARM 集成开发环境简介	170	第 5 章 基于 S3C2410 的系统硬件设计	
4.2 基于 ARM 的汇编语言程序设计	177	5.1 S3C2410 简介	192
4.2.1 ARM 汇编器支持的伪指令	177	5.1.1 S3C2410A 的特点	193
4.2.2 基于 ARM 的汇编语言语句格式	179	5.1.2 存储器控制器	197
4.2.3 ARM 汇编语言程序的基本结构	179	5.1.3 NAND Flash 控制器	198
4.2.4 基于 ARM 的汇编语言程序举例	180	5.1.4 时钟和电源管理	201
4.3 基于 ARM 的硬件启动程序设计	182	5.2 I/O 口	202
4.3.1 分配中断向量表	182	5.2.1 S3C2410A 的 I/O 口工作原理	202
4.3.2 初始化存储系统	183	5.2.2 I/O 口编程实例	208
4.3.3 初始化堆栈	183	5.3 中 断	209
4.3.4 初始化有特殊要求的硬件模块	184	5.3.1 ARM 的中断原理	209
4.3.5 初始化应用程序执行环境	184	5.3.2 S3C2410A 的中断控制器	210
4.3.6 改变处理器模式	185	5.3.3 中断编程实例	213
4.3.7 调用主应用程序	185	5.4 DMA	214
		5.4.1 DMA 工作原理	214
		5.4.2 S3C2410A 的 DMA 控制器	215
		5.4.3 DMA 编程实例	219

目 录

5.5	UART	220
5.5.1	UART 的工作原理	220
5.5.2	S3C2410A 的 UART	222
5.5.3	UART 编程实例	227
5.6	A/D 接口	230
5.6.1	A/D 接口原理	230
5.6.2	S3C2410A 的 A/D 转换器	231
5.6.3	A/D 接口编程实例	233
5.7	键盘和 LED 控制	235
5.7.1	键盘和 LED 的接口原理	235
5.7.2	键盘和 LED 控制的编程实例	236
5.8	LCD	238
5.8.1	LCD 显示原理	238
5.8.2	S3C2410A 的 LCD 控制器	239
5.8.3	LCD 显示的编程实例	245
5.9	触摸屏	247
5.9.1	触摸屏工作原理	247
5.9.2	S3C2410A 的触摸屏接口	248
5.9.3	触摸屏编程实例	250
5.10	音频录放	253
5.10.1	音频录放的实现原理	253
5.10.2	S3C2410A 的 I ² S 总线接口	254
5.10.3	音频录放的编程实例	257
5.11	USB 设备的数据收发	259
5.11.1	USB 接口及编程简介	259
5.11.2	S3C2410A 的 USB 设备控制器	260
5.11.3	USB 设备收发数据编程实例	260

第 6 章 Linux 操作系统基础

6.1	Linux 操作系统概述	264
6.1.1	Linux 操作系统的产生及	

发展	264
6.1.2 Linux 操作系统的特点	265
6.2 Linux 内核的结构	267
6.2.1 进程调度	268
6.2.2 内存管理	270
6.2.3 虚拟文件系统	272
6.2.4 进程间通信	273
6.2.5 网络接口	279
6.2.6 各个子系统之间的依赖关系	281
6.3 Linux 设备管理	281
6.3.1 设备文件	282
6.3.2 设备驱动	282
6.3.3 控制方式	282
6.4 Linux 的使用	283
6.4.1 Linux 常用命令	283
6.4.2 vi 编辑器的使用	297
6.4.3 make 工具和 gcc 编译器	299

第 7 章 嵌入式 Linux 软件设计

7.1	Bootloader 引导程序	308
7.1.1	BootLoader 的启动过程	308
7.1.2	BootLoader——vivi	309
7.2	Linux 的移植	312
7.2.1	Linux 内核的目录结构	312
7.2.2	Linux 内核源码	313
7.2.3	交叉编译环境的建立	313
7.2.4	Linux 内核文件的修改	314
7.2.5	Linux 内核及文件系统的编译	317
7.2.6	Linux 内核及文件系统的下载	317
7.3	驱动程序开发	318
7.3.1	驱动程序的开发步骤	321
7.3.2	键盘驱动程序的开发	322
7.3.3	LCD 驱动程序的开发	326
7.4	应用程序开发	328

7.4.1	应用程序的开发步骤	329	8.5	MiniGUI 在 S3C2410 上的移植	348
7.4.2	键盘应用程序的开发	330	8.5.1	安装 MiniGUI 资源文件	348
7.4.3	基本绘图应用程序的开发	331	8.5.2	配置安装 MiniGUI 库文件	349
7.4.4	跑马灯应用程序的开发	333	8.5.3	MiniGUI 的移植步骤	350
第 8 章 图形用户接口 MiniGUI			8.6	MiniGUI 输入引擎 IAL 的开发	351
8.1	MiniGUI 简介	336	8.6.1	IAL 引擎简介	351
8.1.1	MiniGUI 的功能特色	336	8.6.2	IAL 引擎的开发	351
8.1.2	MiniGUI 的技术优势	336	8.7	在 S3C2410 上运行简单的绘图程序	355
8.2	MiniGUI 在 Linux 下运行环境的建立	337	8.7.1	MiniGUI 的基本绘图函数	355
8.2.1	MiniGUI 在 Linux 下的运行环境	337	8.7.2	绘图程序举例	355
8.2.2	安装资源文件	340	附录 A EL-ARM-860 型嵌入式实验开发系统简介	358	
8.2.3	配置安装 MiniGUI 库文件	340	一、系统结构简介	358	
8.3	MiniGUI 在 QVFB 上的仿真应用	342	二、可开设的实验项目	360	
8.4	在 Linux 下 S3C2410 Frame Buffer 的启动	344	三、产品特点	363	
8.4.1	FrameBuffer 简介	344	附录 B 达盛科技 Techv-DM355 介绍	364	
8.4.2	FrameBuffer 驱动的添加	344	参考文献	367	
8.4.3	FrameBuffer 设备文件的添加	345			
8.4.4	FrameBuffer 测试程序的编写	346			

第 1 章

嵌入式系统基础

1.1 嵌入式系统概念

1.1.1 嵌入式系统的定义

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、专用的系统中的计算机系统,称之为“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,如天天必用的移动电话、带在手腕上的电子表、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和防抱死刹车系统(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,从家庭中的洗衣机、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。

根据英国电气工程师协会(IEE)的定义:嵌入式系统是用来控制或监视机器、装置或工厂等大规模系统的设备。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。

国内对嵌入式系统的一般定义是:以应用为中心,以计算机技术为基础,软硬件可裁剪,从而能够适应实际应用中对功能、可靠性、成本、体积、功耗等严格要求的专用计算机系统。

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。

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 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 月访华时预言,四五年以后,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 嵌入式系统的组成

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,其中使用最为广泛的有 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,嵌入式处理器以外的完成存储、通信、调试、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:包括静态易失型存储器(RAM,SRAM)、动态存储器(DRAM)和非易失型存储器(Flash)。其中,Flash 以可擦写次数多、存储速度快、容量大及价格低等优点,在嵌入式领域得到了广泛的应用。
- 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、I²C(Inter IC)总线接口、USB 通用串行总线接口、Ethernet 网口等。
- 人机交互:包括 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,需要具备一种稳定、安全的软件模块集合,用来管理存储器分配、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,这样的软件模块集合就是嵌入式操作系统。嵌入式操作系统的引入大大扩展了嵌入式系统的功能,方便了应用软件的设计,但同时也占用了嵌入式系统的宝贵资源。一般在比较大型或多任务的应用场合,才考虑使用嵌入式操作系统。

早期的嵌入式系统几乎都用于控制,或多或少都有些实时要求,所以从前“嵌入式操作系统”实际上是“实时操作系统”的代名词。近年来,由于手持式计算机和掌上电脑等设备的出现,也有了许多不带实时要求的嵌入式系统。另一方面,由于 CPU 速度的提高,一些原先被认为是“实时”的反应速度现在已经很普遍了,以前需要在“实时操作系统”上才能实现的应用,现在已不难在常规的操作系统上实现。在这样的背景下,“嵌入式操作系统”和“实时操作系统”就成了不同的概念和名词。

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,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,因此,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。

嵌入式应用对成本十分敏感,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软件有所不同。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,以减少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低效甚至无法完成的。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,接触到的实际应用环境越来越复杂,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。

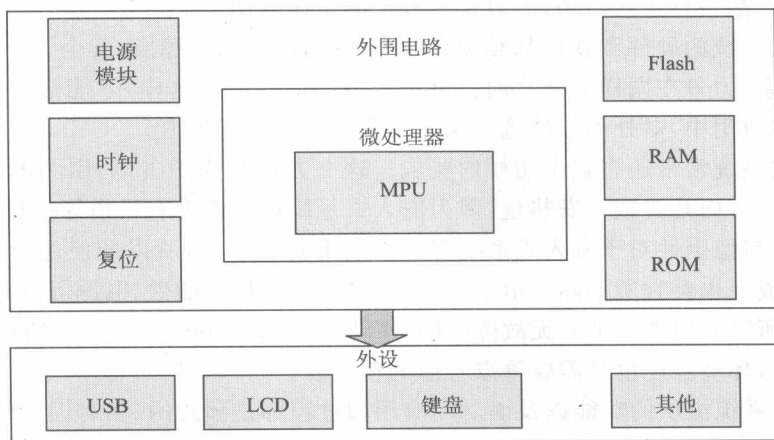


图 1.1 典型嵌入式系统的硬件基本组成

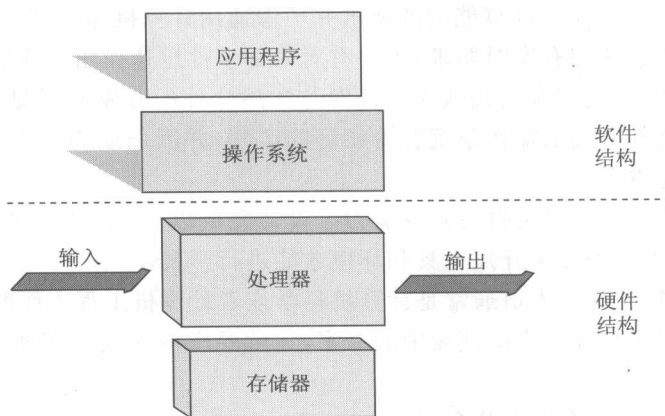


图 1.2 典型嵌入式系统的软件基本组成

1.1.3 嵌入式系统的特点

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,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,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,同时移动能力大大增强,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) 嵌入式系统与具体应用有机地结合在一起,其升级换代也与具体产品同步进行。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产品的市场竞争力。

(5) 嵌入式系统常常还有减小功耗的要求。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手段。

(6) 可靠性与稳定性对于嵌入式系统有着特别重要的意义,所以即使逻辑上的系统结构相同,在物理组成上也会有所不同。由于对所用元器件(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,环境温度也是需要重点考虑的参数。

(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,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了显著的区别。典型的嵌入式实时操作系统与常规的操作系统有着显著的区别,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主开发能力。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,必须具备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。

归纳起来,嵌入式系统有以下几个特点:

- 软硬件一体化,集计算机技术、微电子技术、行业技术于一体;
- 需要操作系统支持,代码小,执行速度快;
- 专用紧凑,用途固定,成本敏感;
- 可靠性要求高;

- 多样性,应用广泛,种类繁多。

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会具有生命力,才会更有优势。嵌入式系统具有很强的专用性,必须根据应用需求对软硬件进行裁剪,满足应用系统的功能、可靠性、成本、体积等要求。

同时还应该看到,嵌入式系统本身是一个外延极广的名词。凡是与产品结合在一起的具有微处理器的系统都可以叫做嵌入式系统,有时很难给它下一个准确的定义。现在人们讲嵌入式系统时,某种程度上指近些年比较热门、具有操作系统的嵌入式系统。

1.1.4 嵌入式系统的应用

嵌入式系统以应用为中心,强调体积和功能的可裁剪性,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,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。它们执行的任务程序一般不由用户编制。

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遍应用。

- 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、轰炸机等陆海空军用电子装备,雷达、电子对抗军事通信装备,野战指挥作战专用设备等。从海湾战争到伊拉克战争中嵌入式系统都有广泛使用。我国嵌入式计算机最早用于导弹控制。
- 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 和 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)广泛采用微处理器、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、煤气表远程抄表以及洗衣机遥控。
- 工业用:各种智能测量仪表、数控装置、可编程控制器、控制机、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,广泛采用微处理器和控制器芯片级、标准总线的模板级及嵌入式计算机系统级。
- 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、防盗系统和各种银行专业外围设备等。
- 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。女娲 Hopen 嵌入式软件已用于机顶盒、网络电视、电话、手机和 PDA 等。
- 医用电子设备:各种医疗电子仪器,如 X 光机、超声波诊断仪、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。

目前,嵌入式系统最热门的应用有以下几种:

1. 个人数字助理(PDA)

目前市面上已经出现了基于 Linux 的 PDA,它具有网络、多媒体等强大功能。康柏公司 iPAQ 掌上电脑一般都预装 Pocket PC 操作系统。iPAQ 采用英特尔的 StrongARM 处理器,这种处理器尽管支持 Linux 系统,但它刚问世的时候却是使用微软公司的 Pocket PC 操作系统。现在,PDA 手机已成为新的热点。

第1章 嵌入式系统基础

2. 机顶盒(STB)

所谓机顶盒 STB(Set Top Box),从表面上理解是指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,是专为那些不很了解计算机的人设计的。现今用户端机顶盒正在朝微型计算机的方向发展,即逐渐集成电视和计算机的功能,成为一个多功能服务的工作平台。用户通过此设备即可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。

3. IP 电话

IP 电话(IP Phone)把电话网与 Internet 结合成一个功能强大的通信网络,它在 IP 网络上实时传输被压缩的语音信息。IP 电话使得语音通信方便与网络价格低廉的特性很好地结合起来,它以数字形式作为传输媒体,占用资源小,成本很低,价格便宜,具有良好的应用前景。

嵌入式系统的应用正在从应用范围狭窄、应用对象单一以及功能简单,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,特别是最近几年随着国际互联网的发展,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,就是对嵌入式系统概念和应用范围的一次变革,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。因此实时逻辑的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以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;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,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前完成,以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,要求系统必须具有并行处理能力,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,信息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,这个限度可以由系统的定义获得。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都是正确的;另一方面指系统的健壮性,也就是说,即使系统出现了错误,或外部环境与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,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 3 个指标来衡量系统的实时性,即:

- 响应时间(Response Time):指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):指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):指在一段给定时间内,系统可以处理事件的总数。吞吐量通常比平均响应时间的倒数小一点。

根据响应时间的不同,实时系统可分为 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,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秒数量级上,这对于关系到安全、军事领域的软硬件系统来说是至关重要的。

2. 弱实时系统

在弱实时系统中,各任务运行得越快越好,但并没有严格限定某一任务必须在多长时间内完成。弱实时系统更多地关注软件运行的结果正确与否,而对任务执行时间的要求相对较宽松。它的响应时间一般是数十秒或更长,可能随着系统的负载轻重而有所变化。

3. 一般实时系统

一般实时系统是弱实时系统和强实时系统的折中。它的响应时间可以在秒的数量级上,可广泛应用于消费电子设备中,如 PDA、手机等。

根据确定性的不同,实时系统可分为以下 2 类。

1. 硬实时

硬实时指系统对响应时间有严格的要求。如果系统响应时间不能满足要求,就会引起系统崩溃或出现致命的错误。

2. 软实时

软实时指系统对响应时间有要求,但是如果系统响应时间不能满足要求,并不会导致系统出现致命的错误或崩溃。

1.2 嵌入式处理器

1.2.1 嵌入式处理器分类

嵌入式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。从硬件方面看,目前世界上具有嵌入式功能特点的处理器已经超过1 000种,流行的体系结构包括MCU、MPU等30多个系列,速度越来越快,性能越来越强,价格也越来越低。

嵌入式处理器可分为:

- 低端的微控制器(MicroController Unit,MCU);
- 中高端的嵌入式微处理器(Embedded MicroProcessor Unit,EMPU);
- 通信领域的DSP处理器(Digital Signal Processor,DSP);
- 高度集成的片上系统(System on Chip,SoC)。

1.2.2 微控制器

微控制器(MicroController Unit,MCU),俗称单片机,它将整个计算机系统集成到一块芯片中。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Flash、RAM、总线逻辑、定时器/计数器、WatchDog、I/O、串行口、脉宽调制输出、A/D及D/A等各种必要的功能模块和外围部件。

最早的单片机是Intel公司的8048,出现在1976年。与此同时,Freescale(原Motorola公司半导体部)公司推出了68HC05,Zilog公司推出了Z80系列,这些早期的单片机均含有256字节的RAM、4KB的ROM、4个8位并口、1个全双工串行口、2个16位定时器。之后在20世纪80年代初,Intel公司又进一步完善了8048,在它的基础上研制成功了8051。20世纪80年代中期,Intel公司将8051内核使用权以专利互换或出售形式转让给世界许多著名IC制造厂商,这样8051就变成有众多制造厂商支持的、发展出上百个品种的大家族。现在,8051是单片机教学的首选机型。

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相同的,不同的是存储器和外设的配置及封装,这样可以使不同的单片机适合不同的应用。与嵌入式微处理器相比,微控制器的最大特点是单片化,体积小,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中的主流产品,其片内资源一般比较丰富,适合于控制。

与微处理器相比,微控制器显而易见的好处是成本上的优势。微控制器将一些接口电路和功能模块集成在CPU芯片上,价格虽然比相应的微处理器高,但如果算上本来就需要的I/O接口芯片和一些独立的功能模块芯片,微控制器往往更加实惠。当然实际的好处远不止于此。

首先,采用微控制器可以在相当程度上缩短产品的设计、开发以及调试时间,从而节约了这些方面的开支。

其次,由于系统中芯片的数量减少了,整个系统的故障率就会降低。研究和统计结果表明,单个芯片的故障率与其集成规模和复杂性的关系并不很大,而整个系统的可靠率则为所有元器件的可靠率的乘积。因此,如果系统中芯片的数量减少了,系统发生故障的概率就会降