

嵌入式系统应用开发

—— 基于C*Core处理器

与嵌入式Linux实时操作系统

陈双龙 朱兴泉 孙学明 编著

- * 细述开发流程和方法
- * 呈现实际案例和源码
- * 提供技术支持和指导



清华大学出版社

91

2007

嵌入式系统应用开发

——基于 C*Core 处理器与嵌入式 Linux 实时操作系统

陈双龙 朱兴泉 孙学明 编著

清华大学出版社

北 京

内 容 简 介

本书以国产高性能 32 位嵌入式 CPU 和开源操作系统 Linux 为核心,结合作者多年从事嵌入式操作系统、开发工具、嵌入式应用的研发的经验体会,系统地介绍嵌入式系统的基本概念和设计原理,嵌入式 Linux 和集成开发环境的设计方法、基于国产高性能 32 位嵌入式 CPU 和嵌入式 Linux 的应用系统设计方法等。

全书分为 8 章,第 1 章和第 3 章介绍嵌入式系统和嵌入式操作系统的基本概念。第 2 章系统地介绍国产高性能 32 位嵌入式 CPU(C*Core)的体系结构、指令系统、中断及接口等。第 4 章介绍 Linux 的体系结构及设计嵌入式 Linux 的步骤及方法。第 5 章介绍目前流行的一些接口,如串行接口、并行接口、USB、网络以及模拟输入输出接口等。第 6 章介绍嵌入式系统集成开发工具的设计方法,如何利用开源代码和第三方工具来提高集成开发工具的开发效率和扩展功能。第 7 章以大量的实例介绍嵌入式系统驱动程序和管理程序的设计方法和具体实现。第 8 章以嵌入式零售信息终端为例介绍基于 C*Core 的控制板及管理软件的设计方案。

本书既可以作为嵌入式信息终端、智能家电、医疗保健、工业控制、智能化社区、移动计算平台、电子商务平台、计算机外设、智能玩具等产品研发的参考书,也可以供高等院校、技术培训班等作为有关嵌入式系统教学的教材。

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 13501256678 13801310933

图书在版编目(CIP)数据

嵌入式系统应用开发——基于 C*Core 处理器与嵌入式 Linux 实时操作系统/陈双龙,朱兴泉,孙学明编著.
—北京:清华大学出版社,2007.5

ISBN 978-7-302-14878-4

I. 嵌… II. ①陈… ②朱… ③孙… III. ①微处理器, ARM——系统设计 ②Linux 操作系统—系统设计 IV. TP332 TP316.89

中国版本图书馆 CIP 数据核字(2007)第 037466 号

责任编辑:文开棋

封面设计:杨玉兰

版式设计:北京东方人华科技有限公司

责任校对:马素伟

责任印制:孟凡玉

出版发行:清华大学出版社 地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn> 邮 编:100084

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印 刷 者:北京市清华园胶印厂

装 订 者:三河市金元印装有限公司

经 销:全国新华书店

开 本:185×260 印 张:23 字 数:542 千字

版 次:2007 年 5 月第 1 版 印 次:2007 年 5 月第 1 次印刷

印 数:1~4000

定 价:39.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系
调换。联系电话:(010)62770177 转 3103 产品编号:023655-01

前 言

嵌入式系统是先进的计算机技术、半导体技术和各个行业具体应用结合而成的产物。嵌入式系统的应用十分广泛，其体积小，功能强，可实现多种功能，因而越来越被市场所认同，其无处不在的应用前景和良好的发展趋势已引起了全球 IT 业众多厂商及广大消费者的关注。随着现代经济的不断进步和信息技术产业的迅猛发展，智能化、低功耗和小型化电子产品的需求日益旺盛，应用领域日益拓宽。

后 PC 时代的到来，使得网络接入终端设备多样化的趋势加快，PC 机将不再是信息处理的惟一选择，嵌入式设备将成为网络接入终端设备的主流。在 21 世纪，嵌入式系统必将得到更加广泛的应用，将有更多、更好、更便宜的嵌入式产品进入千家万户，为人们的生活带来便利。人们也将会更多地依赖手持设备，通过无线网络关注家庭和各種信息，远在千里之外就可以观察到家中的家电设备工作状态及收费情况。到那时，因特网上主要传播的不再是电子邮件和其他信息，而是嵌入式设备与人们交流的信息。

进入 21 世纪以来，随着国家信息产业政策的调整 and 市场的迫切需求，促成国内的一些研究机构和企业自主进行微处理器和嵌入式操作系统的研发。苏州国芯科技有限公司在 Motorola M*Core 基础上（由 Motorola 公司授权）研发的 C*Core 就是一款高性能 32 位嵌入式 CPU，其总体性能与 ARM 相当，片内集成了 IC、LCD、SPI、PWM、RS232 等接口，可以满足很多嵌入式系统的使用要求。最近几年，随着嵌入式系统的快速发展，不少有关嵌入式技术的科技书籍先后出版，但绝大多数都是面向 ARM 处理器，涉及国产嵌入式处理器的书籍稀缺，使得有志从事国产芯片应用开发的读者难觅需要的参考书，多少阻碍了国产 CPU 的普及进程。

本书涉及的内容从国产处理器、嵌入式 Linux、集成开发环境到嵌入式应用系统，基本覆盖了嵌入式系统所有知识点，既可以作为高等院校培养嵌入式系统专业技术人员的教材，也可以作为嵌入操作系统、集成开发环境、嵌入式产品开发专业人员的实用参考手册，更是从事基于 C*Core 处理器和嵌入式 Linux 的嵌入式产品应用开发的技术人员必备参考书籍。

本书共分 8 章。第 1 章介绍嵌入式系统基本特点及其与桌面系统的主要差异，同时还介绍实时系统和嵌入式系统的设计要点。第 2 章介绍嵌入式处理器的特点及分类，详细介绍 C*Core 系列嵌入式 CPU CCM3118 的体系结构、数据类型、指令系统、输入输出接口等，并对 CCM3118 的寄存器和寻址模式进行详细说明。第 3 章阐述嵌入式实时操作系统的技术特点及与通用操作系统的主要差异，并介绍嵌入式操作系统的发展趋势等。第 4 章介绍 Linux 的发展历史、技术特点及体系结构，重点介绍嵌入 Linux 设计的步骤及方法。第 5 章介绍目前流行的一些接口，如串行接口、并行接口、USB、网络以及模拟输入输出接口等。第 6 章介绍嵌入式系统集成开发工具的设计方法，如何利用开源代码和第三方工具来提高集成开发工具的开发效率和扩展功能。第 7 章以大量的实例介绍嵌入式系统驱动

程序和管理程序的设计方法和具体实现。第 8 章以嵌入式零售信息终端为例介绍基于 C*Core 的控制板及管理软件的设计方案。

本书主要由陈双龙、朱兴泉和孙学明编写。参与本书编写的还有柏桂枝、张继果、何炜、郑美红等。全书由陈双龙负责统稿。

感谢连云港杰瑞深软科技有限公司的支持和帮助，感谢清华大学出版社编辑人员为本书出版所做的努力，正是由于他们的辛勤劳动，才使得本书与读者见面。

由于作者工作的局限性，书中难免存在疏漏之处，敬请读者谅解，并恳请各位指正。本书源程序可从 <http://www.deepsoft.net.cn/> 下载。读者朋友也可以从该网站进一步了解嵌入式实时操作系统和开发工具。

陈双龙 朱兴泉 孙学明
2007 年于江苏连云港

目 录

第 1 章 嵌入式系统基本概念 1	
1.1 嵌入式系统的基本特点..... 1	
1.2 实时系统..... 3	
1.2.1 基本概念..... 4	
1.2.2 实时系统的特征..... 6	
1.2.3 实时系统的分类..... 7	
1.3 嵌入式系统设计要点..... 8	
1.3.1 嵌入式处理器的选择..... 8	
1.3.2 操作系统的选择..... 10	
1.3.3 开发工具的选择..... 11	
1.3.4 任务的划分及优先级安排..... 11	
1.3.5 任务间通信方式的选择..... 12	
1.3.6 最小时间片的确定..... 12	
1.3.7 编程语言的选择..... 12	
1.4 嵌入式系统的发展趋势..... 13	
1.4.1 联网成为必然趋势..... 15	
1.4.2 支持小型设备实现小尺寸、 低功耗和低成本..... 15	
1.4.3 提供精巧的多媒体界面..... 16	
第 2 章 嵌入式处理器 17	
2.1 概述..... 17	
2.2 CCM3118 简介..... 18	
2.3 操作模式..... 19	
2.3.1 芯片工作模式..... 19	
2.3.2 低功耗操作模式..... 20	
2.4 编程模式与寄存器..... 20	
2.4.1 编程模式..... 20	
2.4.2 寄存器..... 21	
2.5 数据格式..... 26	
2.6 指令系统..... 26	
2.6.1 指令系统综述..... 26	
2.6.2 指令类型和寻址方式..... 30	
2.7 异常和中断处理机制..... 34	
2.7.1 异常处理概述..... 34	
2.7.2 异常处理过程..... 34	
2.7.3 异常类型..... 37	
2.7.4 异常优先级..... 40	
2.7.5 在异常悬停时的调试请求..... 41	
2.7.6 异常处理程序的返回..... 41	
2.8 处理器的中断控制器..... 41	
2.8.1 寄存器地址映射..... 42	
2.8.2 寄存器功能描述..... 43	
2.8.3 中断处理的实现和中断 服务程序编写..... 48	
2.9 SPI 模块..... 49	
2.9.1 引脚信号说明..... 49	
2.9.2 寄存器地址映射..... 49	
2.9.3 寄存器功能描述..... 50	
2.10 SCI 模块..... 54	
2.10.1 引脚信号说明..... 55	
2.10.2 寄存器地址映射..... 55	
2.10.3 寄存器功能描述..... 55	
2.11 PS/2 主机控制器模块..... 62	
2.11.1 引脚信号说明..... 62	
2.11.2 寄存器地址映射..... 62	
2.11.3 寄存器功能描述..... 63	
2.12 通用读卡器模块..... 70	
2.12.1 引脚信号说明..... 71	
2.12.2 寄存器地址映射..... 71	
2.12.3 寄存器功能描述..... 71	
2.13 液晶显示控制器模块..... 79	
2.13.1 引脚信号说明..... 79	
2.13.2 寄存器地址映射..... 79	
2.13.3 寄存器功能描述..... 80	
第 3 章 嵌入式操作系统 91	
3.1 嵌入式实时操作系统与通用操作 系统的差异..... 91	
3.2 嵌入式操作系统的技术特点..... 95	

3.2.1	微内核结构.....	96	4.3.4	系统调用.....	125
3.2.2	任务调度.....	96	4.3.5	进程调度.....	126
3.2.3	硬实时和软实时.....	96	4.3.6	内存管理.....	128
3.2.4	内存管理.....	96	4.4	Linux 的基本命令.....	130
3.2.5	实时系统的通信.....	97	4.4.1	进入与退出系统.....	130
3.2.6	内核加载方式.....	98	4.4.2	文件的复制、删除和 移动命令.....	131
3.3	嵌入式实时操作系统的体系结构.....	98	4.4.3	目录的创建与删除命令.....	132
3.3.1	层次模型.....	99	4.4.4	Linux 文本处理命令.....	135
3.3.2	单内核.....	99	4.4.5	Linux 备份与压缩命令.....	136
3.3.3	微内核.....	99	4.4.6	Linux 系统管理命令.....	138
3.3.4	虚拟机.....	100	4.4.7	Linux 磁盘管理.....	139
3.3.5	客户机/服务器.....	101	4.5	嵌入式 Linux.....	141
3.4	嵌入式实时操作系统的核心技术 指标.....	102	4.5.1	嵌入式 Linux 设计过程.....	142
3.4.1	任务调度算法.....	102	4.5.2	实时内核设计.....	143
3.4.2	上下文切换时间.....	102	4.5.3	流行的嵌入式 Linux 版本.....	147
3.4.3	中断响应时间.....	103	4.6	基于 C*Core 的嵌入式 JARI-Linux.....	148
3.4.4	确定性.....	103	4.6.1	C*Core 与 Intel 80386 处理器 体系结构的差别.....	148
3.4.5	最大中断禁止时间.....	103	4.6.2	JARI-Linux 内核设计.....	149
3.4.6	系统开销.....	103	4.6.3	JARI-Linux 在 JARI-EVB3118 开发板上的应用.....	155
3.5	嵌入式实时操作系统的设计原则.....	103	第 5 章 嵌入式系统接口技术.....		
3.5.1	实时进程调度算法.....	105	5.1	接口概述.....	159
3.5.2	实时进程调度算法的 正确性.....	107	5.1.1	接口的基本概念.....	159
3.5.3	实时系统的内存.....	107	5.1.2	接口的功能.....	160
3.5.4	外围设备管理.....	108	5.1.3	接口的组成.....	161
3.6	嵌入式实时操作系统现状.....	108	5.1.4	接口的数据传输方式.....	162
3.7	嵌入式实时操作系统发展趋势.....	111	5.2	串行接口技术.....	165
3.7.1	特定应用嵌入式实时操作 系统.....	111	5.2.1	基本概念.....	165
3.7.2	嵌入式安全实时操作系统.....	114	5.2.2	通用异步收发器.....	166
第 4 章 嵌入式 Linux.....			5.2.3	串行接口 RS-232C 标准.....	172
4.1	Linux 发展简史.....	117	5.2.4	串行接口 RS-422 标准.....	174
4.2	Linux 内核原理.....	119	5.3	并行接口技术.....	175
4.3	Linux 内核析析.....	122	5.3.1	并行接口概述.....	175
4.3.1	Linux 源码文件组织结构.....	122	5.3.2	可编程并行接口 芯片 8255A.....	176
4.3.2	Linux 内核引导与初始化.....	123			
4.3.3	中断处理.....	125			

5.4 网络接口技术.....	179	6.4.3 在 IDE 中双击编译信息实现 警告或错误定位	219
5.4.1 网络概述.....	179	6.5 工程管理器的运用.....	222
5.4.2 OSI 体系结构.....	180	6.5.1 功能与界面实现	222
5.4.3 以太网技术简介.....	181	6.5.2 工程管理与 Makefile	224
5.4.4 介质访问控制协议 CSMA/CD	182	6.5.3 工程管理器的技术实现	230
5.4.5 以太网地址.....	183	6.6 联机与程序下载.....	234
5.4.6 以太网的帧格式.....	183	6.6.1 联机	234
5.4.7 网络互连设备.....	184	6.6.2 程序下载	235
5.4.8 网络体系结构.....	189	6.7 交叉调试器的实现.....	236
5.4.9 解决通信时间不确定性的 几种方法.....	190	6.7.1 如何使下载的程序开始 运行	237
5.4.10 几种容错网络方案.....	192	6.7.2 怎么在断点处停下来	238
5.5 数字/模拟、模拟/数字转换接口 技术.....	194	6.7.3 下一步该停在哪儿	240
5.5.1 数模(D/A)转换器.....	194	6.7.4 寄存器、内存和变量的 查看与修改是如何实现的	240
5.5.2 模数(A/D)转换器.....	195	6.7.5 让调试器变得更体贴	243
5.6 串行总线扩展技术.....	196	第 7 章 基于 Linux 的编程实例.....	246
5.6.1 USB(通用串行总线).....	196	7.1 C*Core 多功能开发板简介	246
5.6.2 IEEE-1394.....	197	7.1.1 存储器设计	247
5.6.3 I ² C 总线接口	198	7.1.2 接口设计和引脚说明	248
第 6 章 嵌入式软件开发工具的设计	200	7.2 JARI-IDE 简介	258
6.1 嵌入式软件开发过程及其 基本知识.....	200	7.3 怎样使用仿真器.....	268
6.1.1 什么是宿主机与目标机.....	200	7.3.1 仿真器硬件及配置	268
6.1.2 早期的嵌入式软件开发 方式.....	201	7.3.2 仿真器的使用方法	268
6.1.3 现代的嵌入式软件开发方式 ——交叉开发.....	201	7.3.3 仿真器调试程序的 具体步骤	269
6.1.4 交叉开发基本模型.....	203	7.4 中断处理.....	273
6.2 嵌入式 IDE 开发方案的选择.....	204	7.4.1 C*Core 中断源	273
6.3 功能与界面设计要点.....	206	7.4.2 Linux 中断处理.....	275
6.4 基于 Windows 的 IDE 如何集成 GCC 工具链	209	7.4.3 中断处理程序示例	276
6.4.1 生成 GCC 工具链	209	7.5 看门狗定时器.....	280
6.4.2 在 IDE 的输出窗口中实时 显示编译信息.....	212	7.5.1 看门狗定时器的操作模式及 寄存器	280
		7.5.2 看门狗示例程序	283
		7.6 音频输出.....	286
		7.6.1 音频原理和说明	286
		7.6.2 音频输出示例程序实验	287

7.7 文件系统应用.....	288	7.14.3 USB 设备在 Linux 内核中的配置.....	320
7.7.1 Linux 文件系统特点.....	288	7.14.4 USB 接口示例程序.....	321
7.7.2 Yaffs 文件系统的基本原理.....	289	7.15 Linux 驱动框架.....	322
7.7.3 Linux 下的文件系统应用.....	290	7.15.1 Linux 系统下的设备驱动程序.....	323
7.8 异步串行端口.....	290	7.15.2 设备驱动程序设计示例.....	324
7.8.1 异步串行 I/O 原理.....	290	7.16 以太网卡驱动程序.....	326
7.8.2 串口终端函数.....	291	7.16.1 8390 网卡工作原理.....	327
7.8.3 串行端口应用.....	292	7.16.2 网卡驱动程序示例.....	327
7.9 SPI 接口.....	294	7.17 图形界面设计示例.....	329
7.9.1 C*Core SPI 接口说明.....	295	7.17.1 图形界面简介.....	329
7.9.2 SPI 接口处理程序示例.....	296	7.17.2 Nano-X 的图形显示应用.....	330
7.10 I ² C 串行接口实验.....	297	7.17.3 图形界面设计示例.....	331
7.10.1 I ² C 接口电路说明.....	297	第 8 章 应用案例.....	339
7.10.2 示例程序.....	301	8.1 零售网络管理信息系统简介.....	339
7.11 PS/2 键盘实验.....	309	8.2 零售信息终端的工作原理及主要功能.....	340
7.11.1 键盘按键的工作原理.....	309	8.2.1 主要功能.....	340
7.11.2 PS/2 键盘接口.....	309	8.2.2 设备组成原理.....	340
7.11.3 键盘驱动程序流程.....	310	8.3 零售信息终端设计方案.....	342
7.11.4 PS/2 键盘应用程序示例.....	310	8.3.1 硬件设计方案.....	342
7.12 UCI 接口实验.....	311	8.3.2 零售信息终端软件方案.....	347
7.12.1 UCI 简介.....	311	8.4 应用软件设计.....	350
7.12.2 C*Core 的 UCI 接口.....	312	8.4.1 初始化代码.....	350
7.12.3 UCI 接口处理示例.....	312	8.4.2 输入事件处理模块.....	350
7.13 矩阵键盘实验.....	315	8.4.3 显示处理模块.....	351
7.13.1 矩阵键盘简介.....	315	8.4.4 数据处理模块.....	352
7.13.2 键盘按键值的获取.....	315	8.4.5 输入处理模块.....	352
7.13.3 矩阵键盘处理示例程序.....	316	8.4.6 远程数据传输模块的设计.....	353
7.14 USB 接口实验.....	318		
7.14.1 USB 接口的特点.....	318		
7.14.2 USB 协议简介.....	319		

第 1 章 嵌入式系统基本概念

本章将从嵌入式系统的基本特点、实时系统、嵌入式系统设计要点等几个方面来阐述嵌入式系统的基本概念。

1.1 嵌入式系统的基本特点

嵌入式系统是以应用为中心，以计算机技术为基础，软硬件可裁剪，适应应用系统对功能、性能、可靠性、成本、体积、功耗等严格要求的专用计算机系统。

计算机应用可分成两大类，主机式应用和嵌入式应用。主机式应用是指以计算机系统为工具，完成如科学计算、办公自动化管理、教学等任务。其主要特点是人机交互、任务多变，人们可以直接感受到计算机的存在。而嵌入式应用则把计算机作为一个部件或模块嵌入到应用系统的内部，人们从设备的外部根本感觉不到计算机的存在。前面提到，嵌入式系统是以应用为中心，以计算机技术为基础的专用计算机系统，它一般由嵌入式微处理器、外围硬件设备、嵌入式操作系统以及用户的应用程序四个部分组成，用于实现对其他设备的控制、监视或管理等功能。嵌入式应用是计算机应用最为活跃的领域，也是最有前途的领域。小如收音机、电视、洗衣机等，大如武器控制系统、航天飞机等，计算机都作为核心控制部件嵌入其中，按人们的旨意自动地完成相应的任务。嵌入式应用和主机式应用有着显著的差异，其主要的差别见表 1.1。

表 1.1 嵌入式应用与主机式应用的比较

比较内容	主机式应用	嵌入式应用
运行方式	装入式	嵌入式(固化)
任务性质	多变	固定
操作方式	人机交互	自动或很少的人工干预
实时性	一般	很高
操作系统	大型通用操作系统	嵌入式操作系统
可靠性	一般	很高
修改程度	容易	很难
响应速度	慢	快

现在进一步讨论这些差异。

(1) 运行方式

主机式应用采用装入式方式运行，系统加电或复位时，从大容量的辅助存储介质装入操作系统和应用软件。嵌入式应用通过把控制软件固化在 EPROM 中，直接在 EPROM 上运行，以提高启动速度和可靠性。

(2) 任务性质

主机式应用的任务通常是不定和多变的，可以随时根据情况进行调整。而嵌入式应用

的任务是固定的，改变任务就需重新设计系统。

(3) 操作方式

主机式应用采用人机交互的方式。人们可通过键盘或者其他输入设备向计算机下达各种指令，控制计算机完成指定的工作，而计算机则通过输出设备，如打印机、显示器等，输出计算机的状态和命令执行的结果。而嵌入式应用大多采用自动运行的方式，加电复位后自动投入运行，很少需要人工干预，即使需要人机交互，也有一套严格的操作规范和有限的操控命令。

(4) 实时要求

主机式应用一般对实时性无要求，或者要求不高，只要计算机系统能满足人的视觉感受就可以被人们所接受。而嵌入式应用一般都是实时系统，对时间有很高的要求，通常在 ms 级，有些可达到 μs 级。对实时系统来说，超时可能引发严重后果。

(5) 操作系统要求

主机式应用采用通用操作系统或者分时操作系统。而嵌入式应用则采用嵌入式实时操作系统，有些规模小的系统或装置就根本不需要操作系统。

(6) 可靠性

主机式应用对可靠性有一定的要求，而且有些系统要求也很高，如银行管理系统。但主机式应用的工作环境一般都很好，防止计算机故障的手段也很多。而嵌入式应用需要很高的可靠性，设备的故障所引起的后果可能很严重，而且嵌入式系统的工作环境比主机式应用要差，有些可能在恶劣的环境中运行。

(7) 修改和调整

主机式应用修改、调整、扩充都较为方便，还可以通过购置更高档的计算机和系统来进一步提高功能和性能，这些修改和调整往往可以由非专业设计人员来完成。而嵌入式应用一旦投入使用，很少进行修改和调整，即使需要修改也需由原来的设计人员来实施，因为扩充意味着重新设计。

(8) 响应速度

由于主机式应用采用装入式操作系统，从开机到投入正常运行需要几分钟的时间，人们也已经习惯了这段等待的时间。而嵌入式系统需要快速的响应，尤其是武器控制系统，对于这类控制系统来说，快速响应决定着成败。比如，在敌人突然来袭时，需要紧急开机，如果不能立即投入运行，几分钟后战斗可能已经结束了。

嵌入式系统是围绕着外部世界来工作的，当外部世界需要输入数据时，嵌入式计算机就自动地从外部世界获取数据，并按照原先设计好的工作程序进行处理，处理的结果输出给外部世界。而且，从数据输入到结果输出应该在规定的时间内完成。这就是说，系统工作的正确性不仅依赖于计算结果的逻辑正确性，而且还依赖于得出结果的时间。因此，多数嵌入式系统是实时系统，实时系统的特点及要求均适用于嵌入式系统，而且有些要求比一般的实时系统更苛刻。

嵌入式系统具有以下特点。

- 嵌入式系统通常是面向特定应用的嵌入式 CPU。与通用型 CPU 最大的不同就是嵌入式 CPU 大多工作在为特定用户群设计的系统中，它通常都具有低功耗、体积小、集成度高等特点，能够把通用 CPU 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，移动能力大大增强，跟网络的耦合也越来越紧密。

- 嵌入式系统是将先进的计算机技术、半导体技术和电子技术与各个行业的具体应用相结合后的产物。这一点就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。
- 嵌入式系统的硬件和软件都必须高效率地设计，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。
- 嵌入式系统和具体应用有机地结合在一起，它的升级换代也是和具体产品同步进行，因此嵌入式系统产品一旦进入市场，就会具有较长的生命周期。
- 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芯片或单片机本身中，而不是存储在磁盘等载体中。
- 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，必须有一套开发工具和环境才能进行开发。

1.2 实时系统

实时系统是计算机应用领域里的一个重要分支，大多数实时系统都是嵌入式系统，嵌入在一个更大的控制系统中。实时系统的应用领域很广，从最简单的仪器、仪表控制到最复杂的太空系统，都能看见它的“踪影”。图 1.1 是一个典型的实时过程控制系统。

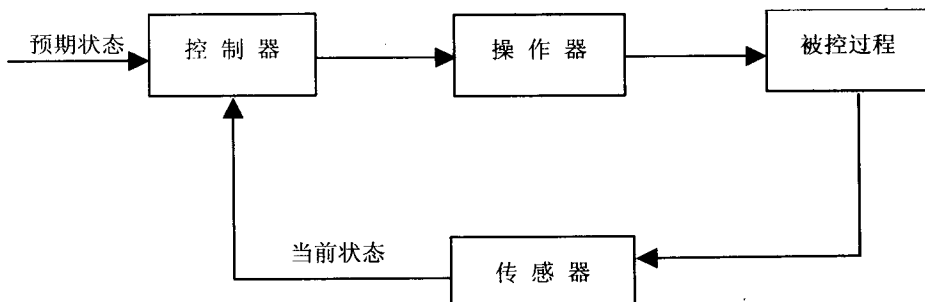


图 1.1 实时过程控制系统

实时系统是这样的系统，即系统工作的正确性不仅依赖于计算结果的逻辑正确性，还依赖于得出结果的时间。实时系统的一个重要特点就是实时系统中的任务具有时间约束。任务的时间约束多种多样，最普遍的时间约束是周期性(periodic)和非周期性(aperiodic)的。周期性任务意味着该任务必须在每个周期内完成一次，可以在周期开始时启动，也可以在周期开始以后启动，但必须在周期结束前完成。非周期性任务则没有规则的时间要求，但必须有一个启动时间或完成时间，或二者皆有。

实时系统的另一个重要的时间约束是时限(deadline)。实时系统与非实时系统的不同之处就在于实时系统的任务都有时限或其他明确的时间约束，实时系统就是围绕着如何满足任务的时间约束来开展工作的。

1.2.1 基本概念

在这里，我们进一步介绍实时系统的几个基本概念。

(1) 响应时间

嵌入式系统响应于外部事件，并与外部世界的时间息息相关，“事件”和对事件的“响应”之间的延迟时间叫做系统的响应时间。响应时间包括“事件”接收后，对数据的获取、数据的处理以及处理结果的输出，直至驱动执行机构的所有时间的总和。响应时间是一个嵌入式系统的重要特征。响应时间一般从几毫秒到几秒，有些需要快速反应的系统甚至要求低于1毫秒。

响应时间是设计嵌入式系统的关键之一，不同的响应时间要求，对系统中数据获取的速度、数据处理的速度、输出数据的速度都有不同的要求，特别对系统中使用的计算机的处理速度、容量以及相应的软件都有不同的时间要求。

(2) 吞吐率

所谓吞吐率，就是一定时间内的信息流量。对于大多数嵌入式系统而言，信息的流量是不均匀的。在有的情况下系统没有信息流量，而在另一些情况下系统具有突发的峰值信息流量。比如，情报处理，搜索雷达以每分钟60转的速度监视它周围的空域，把接收到的点迹数据送给情报处理机，由于目标分布的不均匀，有时目标信息很多，有时根本就没有目标。因此，送给情报处理机的信息是随机的，有时信息量很大，有时没有信息。因此，要求系统必须满足一定的吞吐率要求，即必须满足一定的信息流量要求，最大吞吐率必须满足该系统在峰值负荷情况下信息流量的要求。

(3) 暂存时间

有时，为了使数据不致丢失或得到足够的精度，在一个特定的时间内，必须把这些数据快速采集并保存下来，在规定的时间内处理完并输出计算的结果。如若不然，在获取一个或若干个数据后，不立即对这些数据进行处理，再去获取下一个或下一批数据，就可能在时间上来不及，从而导致数据的丢失。数据被保存下来以资利用的生存时间就是所谓的暂存时间。这种情况在控制设备中屡见不鲜，多个传感器的信息陆续传给控制设备，先把它们暂存起来，在一个确定的时刻才对它们进行处理。暂存的数据往往被保存在缓冲寄存器中。缓冲器的大小取决于暂存数据的信息量以及信息传入的方式。

(4) 多任务计算

一个嵌入式系统往往要处理许许多多的外部事件，而且并不是处理完一个事件后，下一个事件才发生。经常是在处理一个事件的同时，又发生另一个或多个其他事件，甚至同时发生多个事件需要处理。这意味着要在一个时间内响应多个外部事件，处理多个任务，否则就不能及时响应各个外部事件。

(5) 优先级

在一个嵌入式系统中，可能要同时响应多个外部事件，或者说，有多个外部事件同时触发一个系统。在这种情况下，这些外部事件就必须依紧急情况按先后次序进行排队，然后依次进行处理。最紧急的事件先处理，次紧急的情况后处理，并以此类推。这就赋予每个外部事件优先级的概念。优先级高的先处理，优先级低的后处理。外部事件的触发叫做“中断”。因此，合理安排好外部事件的中断优先级是嵌入式系统设计的一个重要任务。

(6) 时限

一个时间界限。它要求一个任务(或一个事件的处理)在该时刻之前完成。它表示一个特定时刻或特定的时间间隔。

(7) 任务执行时间

一个任务从启动到完成所花的时间。它分为平均任务执行时间和最坏任务执行时间。嵌入式系统中一般指最坏执行时间,考虑了可能的等待、阻塞等最不利的情况。

(8) 任务余量

时限减去任务执行时间所得的值。它的大小反映了系统工作的从容程度。

(9) 紧时间约束

紧时间约束指在满足任务时限的同时,留给系统的时间比较紧张。它主要是由任务的余量小而引起的。

(10) 松时间约束

松时间约束与紧时间约束相反,指在满足任务时限的同时,留给系统的时间比较宽松。

(11) 时限粒度

任务从启动时刻到时限之间的时间。当这个值较小时,称为时限粒度小(或小粒度时限);反之,则称为时限粒度大(或大粒度时限)。

(12) 强实时任务

强实时任务也称为强时限任务。要求该任务在时限之前完成,否则其结果将失去可用性。

(13) 弱实时任务

弱实时任务也称为弱时限任务。若该任务不能在时限到来之前完成,其结果的可用性将逐渐下降。

(14) 关键任务

关键任务强调任务的重要程度。若一个任务在时限之前不能完成,则会产生灾难性的后果。

(15) 周期任务

按一定的时间间隔,反复运行的任务称为周期任务。周期任务的运行频率通常与外部事件的采样间隔相关。图 1.2 表示了有关概念之间的关系。

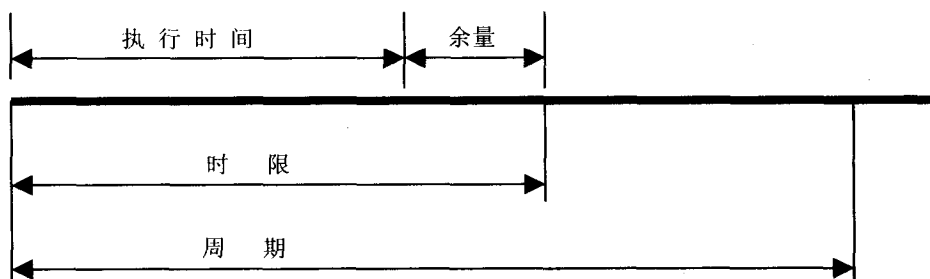


图 1.2 任务时间关系图

1.2.2 实时系统的特征

嵌入式系统必须同时满足逻辑正确性和时间约束两个条件，它在不同应用中表现出的特征各不相同，但其主要的内涵大致相同。

(1) 时限粒度和任务余量

如果一个任务从启动到必须完成之间的时间很短(即时限粒度小)，那么这个时限就紧。但大时限粒度也可能引起紧时限。这是由于这类大时限粒度的任务计算量大、余量小，也会造成紧时限。在嵌入式系统中，紧时限占有绝对的支配地位。设计者应把主要精力集中在这方面，采用有效且简单的技术，以响应这类任务的启动。

(2) 时限的严格性

时限的严格性是指当一个任务的完成超过时限，所得到的结果可用程度。对于强实时任务，在时限过后得到的结果对该任务已失去意义，没有可用性；对于弱实时任务，在时限过后得到的值还有可用性，只是该结果的可用性和精确性逐渐减小。比如在火力控制系统中，火力解算任务是一个强实时任务，超过时限得到的计算结果是无效的，而显示处理任务则是弱实时任务，超过时限的值虽然不太精确，但仍然可以使用。图 1.3 给出了两种任务的时限要求。

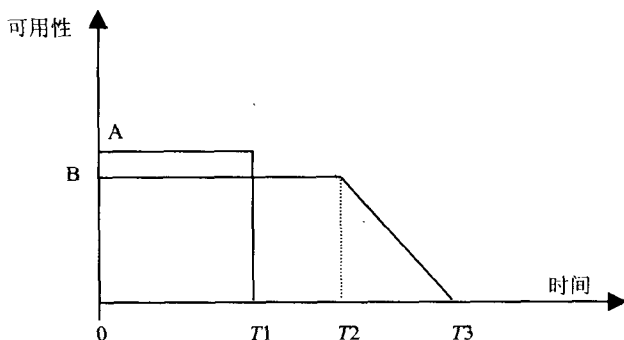


图 1.3 两种任务时限要求

T_1 为强实时任务 A 的时限。当 A 超过 T_1 时，可用性为“0”；B 为弱实时任务， T_2 为 B 任务的时限，当 B 超过时限，可用性和精确度逐步下降，到 T_3 时刻就变为不可再用。

对强实时任务和弱实时任务，通常采用不同的技术。比如，强实时任务可采用外同步方式的通信、共享存储器通信等。而弱实时任务则可采用信箱等比较严格的通信手段。

(3) 可靠性

许多嵌入式系统在高可靠性要求下进行工作，即如果特定任务(称为关键任务)错过了其时限，就会造成灾难性的后果。为保证这些任务满足时限，采用静态分析和资源预分配的方法以及冗余配置，使关键任务的要求总能得到满足。强实时任务并非都是关键任务。只有那些超过时限会造成灾难性后果的强实时任务才是关键任务。

(4) 可预测性

可预测性是指能够确切断定一个任务的完成时间的能力。它是对嵌入式系统的一项重

要的性能要求。在嵌入式系统中,由于各个任务有各种各样的时间约束,任务之间还有嵌套现象,其重要性和紧迫程度也不一样,所以要保证系统的可预测性实际上相当困难。

有些系统只能保证所有的中断(包括中断嵌套)能在其时限内得到响应,但不能保证系统中运行的任务均能满足时限要求,这种可预测性成为微观级的可预测性。如果系统不仅能保证所有中断在预定时间内得到响应,而且能保证所有的任务在其时限内完成,这种可预测性就称为宏观级或系统级的可预测性。

尽管可预测性是每个设计者所追求的目标,但在实际应用中,并不一定要保证所有的任务均是可预测的,只要其中的关键任务能百分之百满足时限要求,而其他的任务总体上满足要求即可。

一个任务除了有时间约束外,还可能有资源约束、优先关系、并行约束、位置约束和紧迫性要求,而这些要求最终都以时间约束的形式表现出来。在不同的系统中,尽管设计的目标不相同,但是,所有的设计者几乎都希望自己设计的系统是可预测,无一例外。可预测性是嵌入式系统的一个很重要的特征,也是评价系统的一个重要标准。

(5) 实时与快速

实时并不等于快速,实时并不意味着时间约束是微秒或纳秒,虽然大家都希望嵌入式系统有相当快的速度,但首先要求的是可预测的。一个系统,不论它的速度有多么慢,只要能满足应用的要求,就是一个能满足要求的系统。当然,速度越快的系统,它的应用领域也越广,实时性也越好。但是,不能以系统的速度快慢来判定它是否是嵌入式系统。

1.2.3 实时系统的分类

实时系统可分为实时过程控制系统和实时信息处理系统。实时过程控制系统是指该系统实时地从外界获得被控制系统的当前状态信息,进行预定的处理,并根据处理结果对外界被控系统进行及时控制,使其处于要求的状态下。实时信息处理系统是指系统仅从外界获取当前的状态信息,并进行相应的处理,但不直接控制外界系统。

从实时系统在教育中的应用来看,它一般作为应用系统的一部分,是处于信息监测和控制部件之间的一个部件,属于嵌入式应用。从不同的角度出发,对实时系统有不同的分类方法。

1. 从系统规模上划分

(1) 单板、单片机实时系统

这是一种最简单的实时系统,一般由传感器、处理器和控制器三部分组成,广泛应用于仪器、仪表、工业控制和智能机器中。这类系统功能简单,但开发容易、体积小、应用范围广。

(2) 嵌入式实时系统

这是典型的实时系统,具有比较精炼而完备的实时操作系统,内核精炼,性能较高,应用系统需要交叉开发。

(3) 大型实时系统

这类系统的特点是操作系统功能强大,应用于大型系统中。其操作系统既有实时操作系统的特征,又有通用操作系统的功能,通常运行在高档微机、小型机或大型机上。尽管

其硬件无法安装在应用系统中，因此形式上不是嵌入式的，但其功能、地位仍属于嵌入式系统。

2. 从实时要求上划分

(1) 强实时系统

系统中至少有一个强实时任务，系统设计围绕着如何满足这些强实时任务的时限而采取一些必要的措施。

(2) 弱实时系统

系统中的实时任务全部是弱任务，尽管它们对时限的要求不如强实时任务那么严格，但它们仍是实时任务，满足时限仍是设计中需要考虑的一个重要因素。

3. 从体系结构上划分

(1) 集中式实时系统

集中式系统的特点是一个完整的控制处理过程由单机完成。由于所有处理工作由单机完成，所以此类系统易于设计和控制，其理论和实践都发展得较为成熟。虽然其优点来自于单机处理，但缺点也在单机处理上。由于应用日益复杂，仅用一台计算机往往无法胜任，无法满足许多复杂的实时应用。

(2) 分布式实时系统

分布式系统的主要特点是多机共享资源，协调一致工作共同完成一个任务。分布式实时系统在保证实时性方面要比集中式系统困难得多。首先是如何保证网络通信时间有界和可预测。其次是调度问题，包括负载平衡、信息传递、减少系统开销、任务粒度划分等。

1.3 嵌入式系统设计要点

嵌入式系统包含硬件和软件两部分：硬件架构上以嵌入式处理器为中心，配置存储器、I/O 设备、通信模块等必要的外设；软件部分以软件开发平台为核心，向上提供应用编程接口(API)，向下屏蔽具体硬件特性的板级支持包(BSP)。在嵌入式系统中，软件和硬件紧密配合，协调工作，共同完成系统预定的功能。

对于不同的市场应用类型，嵌入式系统开发中的嵌入式处理器、实时操作系统、仿真器、调试器以及开发队伍的技术水平和结构比例等要素的选择是至关重要的。应用对象的不同，设计原则及方法就不同，不但取决于技术，同时也包括很多人为的因素在内，如客户的要求，设计者对问题的认识等。但围绕着控制系统这一类嵌入式系统，有一些共性的问题需要我们在设计中给予重视。

1.3.1 嵌入式处理器的选择

嵌入式处理器的选择取决于众多的因素，如下所示：

- 用户要求
- 费用的限制
- 功能与速度要求