

嵌入式

开发专家

基于单片机 8051的 嵌入式 开发指南

胡大可 李培弘 方路平
飞思科技产品研发中心

编著
监制



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

TJ368.1

142

嵌入式

开发专家

基于单片机8051的

嵌入式开发指南

胡大可 李培弘 方路平 编著

飞思科技产品研发中心 监制



00181933



石化 S181933A

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

8051 系列单片机是目前在嵌入式系统领域中,应用最为广泛的微处理机之一。其广泛的使用、低廉的价格、相对简单的开发方法,使它赢得了众多用户的青睐。本书内容由长期从事这一领域开发的多位专家总结实际心得编写而成,书中针对 8051 单片机在嵌入式系统中的开发应用技术,特别对于目前 8051 嵌入式系统开发中流行的实时操作系统——RTX51 和 μ C/OS 的结构和应用做了详细的分析和介绍,使读者能深入掌握嵌入式系统开发中的日渐流行的 C51 编程技术,实时操作系统平台 RTX51 和 μ C/OS 的应用,嵌入式系统设计中软硬件的整合方法,以及集成开发环境 μ Vision 的使用。书中对于 8051 嵌入式系统的应用给出了实例设计参考。飞思在线 <http://www.fecit.com.cn> “下载专区”提供书中范例源代码。

本书可以作为高等院校计算机、自动化及电子技术类专业的教学参考书,也可作为工程技术人员在设计开发中的技术资料。

未经许可,不得以任何方式复制或抄袭本书的部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

基于单片机 8051 的嵌入式开发指南 / 胡大可, 李培弘, 方路平编著. —北京: 电子工业出版社, 2003.1
(嵌入式开发专家)

ISBN 7-5053-8398-1

I.基... II.①胡...②李...③方... III.单片微型计算机, 8051 IV.TP368.1

中国版本图书馆 CIP 数据核字 (2002) 第 105037 号

责任编辑: 郭 晶 陆舒敏

印 刷: 北京东光印刷厂

出版发行: 电子工业出版社 <http://www.phei.com.cn>

北京海淀区万寿路 173 信箱 邮编: 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 19.25 字数: 492.8 千字

版 次: 2003 年 1 月第 1 版 2003 年 1 月第 1 次印刷

印 数: 5000 册 定价: 29.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话: (010) 68279077

“开发专家”是电子工业出版社计算机研发部长期以来精心培育的计算机科学技术类本版品牌。这个品牌是由多个专题系列组成的横向大系列，涵盖了计算机技术的各个方面，特别是一直受到极大关注的程序开发类系列，例如“开发专家之数据库”、“开发专家之网络编程”、“开发专家之 Delphi”、“开发专家之 Sun ONE”、“开发专家之 Oracle”和“嵌入式开发专家”等。这些专题系列基于各自的角度，从纵向上包含了该专题的所有内容。因此，整个“开发专家”的品牌架构纵横交错，囊括了所有的计算机技术和所有的技术层面，海纳百川而又极具可扩展性。

“开发专家”的作者队伍主要依托于“飞思科技产品研发中心”。“飞思科技产品研发中心”由专业的策划人员、权威的技术专家和资深的作者队伍共同组成。在图书的出版上，形成了以研发为基础、以出版为中心、以服务为支持的专业化出版框架和流程。通过深入的市场调查和技术跟踪，在综合了技术需求和读者焦点等因素的基础上，形成各系列丛书的写作重点和大纲，然后聘请业界的最前沿学者进行写作。同时，策划工作全程介入写作进程，严格控制写作质量，用专业的技术背景、深刻的理论基础、具代表性的案例、能为专业读者接受的形式，为读者提供品质最佳的图书产品，体现了出版者和著作者的完美结合。

多年来，我们始终把创造社会效益摆在首位；秉承一切为国内计算机技术专业读者服务的精神，为推动国内信息技术的发展、为体现国内技术的原创水平，穷尽所有的创意与努力，将出版者的命运与读者的支持紧紧地连在了一起。

今天，嵌入式系统已经成为计算机领域的热门话题。嵌入式系统的研究与开发有它独特的内容，包括：嵌入式系统的原理、体系结构、CPU 芯片、外围功能器件、开发工具等各个方面。“嵌入式开发专家”丛书内容覆盖嵌入式系统概念范畴的热点，面向从事嵌入式系统研究设计开发的科学技术人员，也包括大专院校的工科学生。

在此，我们临出版之残酷竞争而不惧，旌旗猎猎而异军突起，这与广大读者的支持是分不开的。为使我们的脚步更坚实、使我们的队伍永远保持活力和创造力，我们期待着您能为我们的前进贡献出您的意见和建议。同时，我们也在等待着您的加入。

电子工业出版社计算机研发部

关于飞思

新世纪之初的北京，一群满怀共同理想的年轻人聚集在飞思教育产品研发中心的旗帜下，他们将新的希望和活力注入了中国 IT 教育产品开发领域。飞思人在为把自己打造成为中国 IT 教育产品研发的精英团队而更加不懈努力。

21 世纪的今天，飞思人在多元化教育产品的开发和出版等方面已经迈出了坚实的第一步，开拓出属于自己的一片天空，初步赢得了涓涓细流。

如今，本着教育为科技服务的宗旨，飞思教育产品研发中心拓展为飞思科技产品研发中心，并以崭新的面貌等待您的支持与关注。

飞思人理念

我们经常感谢生活的慷慨，让我们这些原本并不同源的人得以同本，为了同一个梦想走到一起。

因为身处科技教育前沿，我们深感任重道远；因为伴随知识更新节奏，我们一刻不敢停歇。虽然我们年轻，但我们拥有

“严谨、高效、协作”的团队精神

全方位、立体化的服务意识

实力雄厚的作者群和开发队伍

当然，最重要的是我们拥有：

恒久不变的理想和永不枯竭的激情和灵感

正因如此，我们敢于宣称：

飞思教育 = 丰富的内容 + 完美的形式

这也是您和我共同精心培育的品牌  的承诺。

“问渠哪得清如许，为有源头活水来”。路再远，终需用脚去量；风景再美，均需自然抚育。

年轻的飞思人愿为清风细雨、阳光晨露，滋润您发芽，成长；更甘当坚实的铺路石，为您铺就成功之路。

前言

8051 系列单片机是国内目前应用最广泛的一种 8 位单片机。经过了二十年左右的推广，熟悉它的科技人员非常多，许多高校至今仍然用它作为微处理机原理课程的基本内容。同时，生产以 8051 为内核的单片机的厂家也非常多，并且都做了功能和资源的扩展，形成了自己的技术特色。因此，8051 系列是一个有众多品种、发展历史较长、至今仍有生命力的一批单片机的总称，8051 系列会在嵌入式系统应用领域继续活跃许多年。

用 C 语言程序设计来实现嵌入式系统的系统软件和应用软件开发，可以提高开发调试工作的效率；同时，所产生的技术文档资料也容易理解，便于移植。适用于 8051 系列的 C 语言，目前已开始用得越来越广泛。比较有影响的有 Keil-C 体系。但是，在 8051 系列的嵌入式系统上建立操作系统平台的应用模式推广才刚刚开始。

8051 系列虽然片上资源有限，且为 8 位数据宽度，但是经过多年的实践与改进，已有几个适合它的操作系统开始崭露头角，如：RTX51 与 $\mu\text{C}/\text{OS}$ 。随着功能、资源扩展了的各种 8051 系列新型号芯片的问世，相信这些精巧的 8 位操作系统是会有用武之地的。

应用 8051 系列的嵌入式操作系统，涉及 8051 系列的硬件特性、8051 系列的 C 语言、面向 C 语言的集成调试环境，以及嵌入式操作系统原理等内容。本书集中了这些内容中比较精华和实用的部分，加以重新组织。从嵌入式系统基本概念和 C 程序设计基础入手，逐步深入地阐述了 8051 系列的 C 程序设计环境和建立在它基础上的操作系统。同时，书中给出了嵌入式操作系统的应用实例设计参考，希望本书的内容对推广嵌入式操作系统的应用能有所帮助。

阅读本书的读者最好有 8051 汇编语言程序设计经验，具备 C 语言程序设计的初步知识。如果尚未学过 C 语言，只要仔细阅读书中的有关章节也完全可以达到基本掌握和应用的程度。

本书由飞思科技产品研发中心策划，胡大可、李培弘、方路平编著。在本书写作过程中，得到了研究生郑音飞、徐冰俏、汪强等人的很大帮助，对于他们的辛勤工作在此表示感谢。由于作者的学识水平有限，书中难免有错误和不妥之处，恳请读者提出批评指正。

我们的联系方式:

咨询电话: (010) 68134545 68131648

答疑邮件: support@fecit.com.cn

飞思在线: <http://www.fecit.com.cn> <http://www.fecit.net>

答疑网址: <http://www.fecit.com.cn/question.htm>

源代码下载: <http://www.fecit.com.cn/download.htm>

通用网址: 计算机图书、飞思、飞思教育、飞思科技、FECIT

飞思科技产品研发中心

目 录

第 1 章 嵌入式系统概述	1
1.1 什么是嵌入式系统	1
1.2 嵌入式操作系统	3
1.3 嵌入式系统开发技术	6
1.4 嵌入式系统的应用领域	9
第 2 章 8051 系列单片机	11
2.1 8051 单片机的硬件结构	11
2.2 8051 的时钟与中断系统	15
2.3 8051 的片内外围模块	18
2.4 8051 的程序设计	21
第 3 章 C51 的基本知识	23
3.1 标识符与关键字	23
3.2 数据基本类型	24
3.3 C 语言的运算符	28
3.4 程序设计的三种基本结构	32
3.5 函数	39
3.6 数组	45
3.7 指针	48
3.8 结构和联合	53
3.9 枚举	57
3.10 类型定义	58
3.11 位运算	59
3.12 预处理功能	61
第 4 章 C51 的扩展特性	65
4.1 C51 的特性	65
4.2 C51 的定制文件	79
4.3 C51 的段	81
4.4 C51 的汇编程序接口	83
4.5 C51 编译控制命令	86
4.6 C51 编译控制命令说明	88
第 5 章 C51 的库函数	97
5.1 C51 的库文件	97
5.2 C51 的库函数分类	97
5.3 C51 库函数说明	102
第 6 章 μ Vision2 集成开发环境	129
6.1 μ Vision2 的界面功能	129

6.2	μ Vision2 的基本操作	135
6.3	μ Vision2 的主要功能说明	150
第 7 章	RTX51 实时操作系统	169
7.1	RTX51 系统特点	169
7.2	RTX51 Tiny 使用说明	177
7.3	RTX51 Tiny 系统函数	182
7.4	RTX51 Tiny 应用实例	188
第 8 章	μ C/OS 实时操作系统	199
8.1	μ C/OS 概述	199
8.2	μ C/OS 内核结构	203
8.3	μ C/OS 中的数据结构	215
8.4	μ C/OS 的库函数	218
8.5	μ C/OS-II 的移植与应用示例	246
附录	编译出错信息	275
	致命性错误	275
	句法错误	278
	警告性错误	295

第 1 章 嵌入式系统概述

今天，嵌入式系统（Embedded System）已经成为一个十分引人注目的领域。本书将介绍以 8051 系列单片机为核心的嵌入式系统的设计和应用方法。

在这一章，先对嵌入式系统的基本概念加以阐述。

1.1 什么是嵌入式系统

1.1.1 嵌入式系统概念的产生

1976 年，Intel 公司推出了 8048，称为单片机（SCC，Single Chip Computer）。这个只有 1KB ROM 和 64B RAM 的简单芯片成为世界上第一个单片机，开创了将微处理机系统的各种 CPU 外的资源，如：ROM、RAM、定时器、I/O 端口、串行通信接口及其他各种外围功能模块，集成到 CPU 硅片上的时代。

相对于纯粹的 CPU，人们将它们称之为单片机，以表明它们在一个芯片上不仅有 CPU，还有存储器及 I/O 接口等功能模块。但是，对比今天多如繁星的 CPU 和单片机后，也可以看出它们之间其实并不存在一条不可逾越的鸿沟。

1980 年，Intel 推出了 8 位的 8051 系列。此后，在 1984 年又推出了 16 位的 8096 系列。推出 8096 时，Intel 将它称之为嵌入式微控制器（Embedded Microcontroller）。这可能是“嵌入式”一词第一次在微处理机领域中出现。这个名词的本意是要说明 8096 是一个嵌入到应用系统中的核心控制芯片。

今天，单片机已经成为一个十分庞大的家族。虽然性能各异，但是它们的应用目标几乎是一致的，即嵌入到某一个应用系统中，针对特定的应用目标，利用单片机的软硬件资源，实现检测、控制、计算及通信等功能。

特定的环境、特定的功能，要求单片机与所嵌入的应用环境成为一个统一的整体，并且往往有紧凑、高可靠性、实时性好、低功耗等技术要求。这样一种应用目标使得这一应用领域要去研究它的独特的设计方法和开发技术，这就是今天嵌入式系统这一名称的含义，也是嵌入式系统成为一个相对独立的计算机研究领域的原因。

相对于嵌入式系统，通用计算机系统在应用目标上有巨大的差异。

人们最熟悉的通用计算机就是 PC。PC 的设计目标并不针对专一的应用，甚至不针对特定的开发语言。PC 的设计师希望 PC 具有尽可能通用的特性，如：具有多种开发设计语言及工具，能在差异极大的不同应用领域都发挥出优异的性能，有尽可能快的处理速度，有尽可能大的存储空间，有尽可能强大的外围设备。

这些强大的整机性能指标，对于某一个具体的应用目标来说，也许并不是必要的。而

对于另外一些应用中经常会提出的技术要求，PC 反而是难以满足，如：低成本、低功耗、小体积、高可靠性等等。

1.1.2 林林总总的嵌入式系统

今天，嵌入式系统内涵正在日益深化。如果再将嵌入式系统的技术要素看成仅仅是单片机芯片的硬件特性、单片机的指令系统、单片机各引脚的时序、单片机的外围模块特性，以及单片机的仿真开发工具，那就显得远远不够了。

嵌入式系统是一个应用系统，它应该是一个硬件和软件的统一体。而软件在嵌入式系统中将占有更为重要的位置。

嵌入式系统的软件可以分为系统软件和应用软件两个层次。对软件的这种分割是必要的。当应用问题较为简单时，也许不必有很清晰的软件分层。但是，只要是一个稍微复杂一些的应用系统，就会面对如下的要求：

- 功能的实现尽可能不依赖于具体的硬件环境；
- 系统要求达到更高的安全性、可靠性指标；
- 希望软件设计达到较高的标准化程度；
- 希望提高软件模块的可读性、可移植性和再利用率；
- 希望实现团队式的开发方式。

这时，需要一个系统软件作为硬件和软件的过渡层，来实现这些设计要求提供良好的保障。

当然，系统软件的开销要有一个适当的度。当系统软件做得过于面面俱到时，就必然会消耗大量的系统软硬件资源，同时，必然会降低嵌入式系统的实时性，增加嵌入式系统的规模与成本。

以下是目前比较流行的几种嵌入式系统设计风格。

1. 缩减 PC 系统

所谓缩减 PC 系统，是指利用 PC 体系结构设计的嵌入式系统，例如利用 PC104 模块构成工业控制装置就是比较典型的设计。

这种设计是建立在技术上已非常成熟的 PC 的体系结构之上的，它的硬件环境往往是一台单板化的 PC 系统。利用 DOS 或 Windows 操作系统为应用软件提供平台。

这种设计可以利用 PC 作为开发工具，可以利用众多的 PC 环境软、硬件资源，在成熟的操作系统支持下，系统可以达到较高的可靠性和稳定性，这些显然是它的优点。但是这样的设计目前尚难以实现满足小体积、低功耗、低成本等嵌入式系统的常见技术要求。

2. 高级单片系统

所谓高级单片系统，是指那些准备加载 Linux 操作系统或类 Linux 操作系统的嵌入式系统。它的硬件构成的核心是一个集成了丰富功能的单一芯片，一般数据宽度往往是 32 位。它已经包含了几乎全部的系统硬件，使得只需再增加很少几个器件，如存储器芯片，即可构成全部系统。

生产高级单片的厂家及高级单片型号越来越多了，典型的是以 ARM 或 MIPS 内核为核

心的单片机。在这里已经完全没有有了 PC 体系结构的影子。芯片包容的功能极其丰富，往往除了大容量存储器以外，系统的硬件几乎都集成在一个单片上。它们的寻址空间大，数据总线宽，处理能力强，功耗低。

这些芯片的设计目标是非常明确的，就是为了构成一个嵌入式系统。利用这样的芯片可以设计出非常紧凑的系统，ARM 内核单片在移动电话上的成功是一个有力的佐证。

采用高级单片的设计，大都取 Linux 或类 Linux 操作系统作为系统软件，向应用软件提供 C 语言级的开发平台。在 Linux 操作系统的支持下，系统的可靠性可以得到保证。操作系统带来的优点可以大大加快系统应用软件的开发效率。由于 Linux 的开放性，可以利用的资源也非常丰富。

3. 单片机系统

目前称之为单片机的，是指一些 8 位或 16 位数据宽度、寻址空间较小的芯片。相对来说，它们的处理能力较弱。但是，它们的优点也是不容忽视的，如：低成本、低功耗，片上集成的外围模块功能丰富而实用等。在应用需求恰如其分的场合，选择它们绝对是合理的。

单片机的软件一般应该分层为系统软件和应用软件。系统软件处于底层硬件和高层应用软件之间。但是，单片机的资源有限，设计者必须在系统软件的功能与构成系统软件的软硬件开销之间，仔细地寻求平衡。

单片机的系统软件构成有两种模式：

- 监控程序

用非常紧凑的代码，编写系统的底层软件。这些软件实现的功能，往往主要是实现系统硬件的管理及驱动，并内嵌一个用于系统的开机初始化、程序代码下载及辅助调试等功能的引导（BOOT）模块。

- 操作系统

经过多年的努力，今天已有许多种适合于 8 位至 32 位单片机的操作系统进入实用阶段。在操作系统的支持下，嵌入式系统会具有更好的技术性能，如：程序的多进程结构，与硬件无关的设计特性，系统的高可靠性，软件开发的高效率等等。

1.2 嵌入式操作系统

今天，嵌入式系统的设计者越来越清楚地认识到，在嵌入式系统中引入操作系统的必要性。而嵌入式系统的操作系统与一般概念上的操作系统，即通用计算机的操作系统是有很大差别的。

1.2.1 操作系统的基本概念

操作系统在计算机系统中，是底层硬件和应用软件之间的过渡。它在计算机系统的位置可以用图 1-1 来示意。

应用软件
操作系统
嵌入式系统硬件

图 1-1 操作系统结构

在操作系统的支持下，应用软件可以通过操作系统来与硬件打交道。这为嵌入式系统脱离特定的硬件环境提供了条件，也使得软件的可靠性、安全性增加了。利用操作系统软件的特权性，可以保证嵌入式系统始终工作在有效控制之下。

操作系统的功能主要体现在以下几个方面。

1. 进程管理

进程是一个运行中的程序。在操作系统中，进程具有独立性。多个进程在操作系统的调度下，分时、并发地运行。这样的结构，使得软件的开发可以按相对简捷的功能模块分别进行；可以利用一种所谓的信号灯机制，实现各个进程之间的通信，分配进程对各种资源的占用；可以利用进程调度，避免系统陷入死循环或崩溃；可以将进程设置为不同的优先级别，例如系统级或用户级，来保证系统的安全性。

2. 内存管理

内存管理是将计算机的内存分成若干页面，对各个页面赋予不同的特性和访问逻辑地址。利用内存页的不同特性，可以实现不同的访问特性。例如，可以为特殊的任务分配特定的内存页，同时也避免了其他任务侵入这一内存页。由于内存访问的实时性，这种页面的分配是由硬件实现的。一般来说，是依赖于 CPU 的支持来实现的

3. 文件系统管理

文件系统是计算机系统的一个特殊组成部分。文件系统将计算机管理的大量数据以特定的结构保存在存储系统中，这个特殊的数据结构就是文件。文件系统一般建立在外存储器中，如磁盘、磁带、光盘等等，以满足数据容量的要求。但是，在特殊的情况下，文件系统也可以建立在计算机的内存中。

4. 设备驱动程序

在操作系统的管理下，应用程序不必要也不应该与底层的各种设备直接打交道。应用程序可以经过操作系统提供的设备管理手段，即设备驱动程序，来使用系统的设备。设备驱动程序一般包括对设备的初始化，检查设备状态，控制设备动作，对设备进行读写操作等功能。

5. 系统调用

一个操作系统的各项功能，往往通过一系列应用软件可引用的程序模块来实现，称为系统调用函数或应用编程接口。这些系统调用模块经过比较严格的测试和实用考验，用它们作为整个应用系统的基础可以保障系统的稳定性和可靠性。

一个建立在操作系统平台上的应用软件，具有更多的优良特性：

- 借助于操作系统平台，应用软件获得了良好的可靠性；
- 应用软件可以不依赖或少依赖于系统的底层硬件；

- 利用进程概念开发应用软件，使得应用软件有良好的结构；
- 操作系统提供的功能齐全的系统调用可以加速软件开发；
- 有利于团队式的应用软件系统开发；
- 有利于应用软件的移植。

1.2.2 适合嵌入式系统的操作系统

当年，操作系统诞生的动因是为了解决大型通用计算机的硬件层屏蔽和多用户多任务管理。经过多年的发展和演变，计算机操作系统的功能基本没有很大的变化，但是，它已经从大型计算机领域扩展到微型计算机领域中。当然，这更多的还是得益于微型计算机系统的 CPU 处理能力大大地提高了、微型计算机系统的内存也扩展了。今天，一台微型电脑的内存空间和处理速度，绝对不逊色于当年的一台大型计算机系统。但是，如果要将操作系统加载到一个嵌入式系统上，所面临的问题是全新的，这是操作系统技术在最近几年的重大突破之一。

对于嵌入式系统，不言而喻，它比通用计算机具有更简单的结构。它很可能不配置 CRT 显示器，不需要文件系统，由于内存空间较小也没有存储器管理功能。同时，嵌入式系统总是希望加载的操作系统软件不能占据过大的内存空间，不能消耗过多的系统软硬件资源。这样就要求嵌入式系统的操作系统与传统意义上的操作系统有很大区别，要做到代码量小，对堆栈、寄存器、定时器及中断等系统部件的依赖要少，能在各种不同类型的单片机上实现。

1.2.3 目前流行的嵌入式操作系统

嵌入式系统的操作系统，除了对于缩减 PC 系统目前仍采用 DOS 等 PC 上的流行操作系统以外，基本上有两大趋势：一类是面向高级单片的，另一类是针对 8 位、16 位单片机的。以下是几个目前流行的嵌入式操作系统。

1. Linux

Linux 已经成为 Windows 系统问世以来最热门的操作系统之一。它的开放性使众多的热情开发者为它打造了非常坚实的基础。同时，它也派生出众多的类似系统。

2. μ CLinux

μ CLinux 是一个缩减的 Linux 系统，特别适合于用在不需要内存管理的高级单片嵌入式系统上。类似于 μ CLinux 的 Linux 操作系统的缩减版本，现在越来越多。相信经过一段时间的实践检验，会优选出理想的嵌入式 Linux 操作系统。

3. eCOS

eCOS 是一个代码开放的嵌入式操作系统，具有良好的系统功能和应用支持，可以在许多高级单片上运行。

4. Windows CE

Windows CE 是 Windows 的嵌入式系统版本，具有类似 Windows 风格的用户界面，可以与 Windows 环境下的软件很方便地接口。但是，它的代码是不开放的。

5. VxWorks

VxWorks 是一个功能完善的嵌入式操作系统，但是它的代码也是不开放的。

6. RTX51

RTX51 是专门针对 8051 设计的操作系统，代码紧凑、体积小巧。已经在很多应用中证明这是一个成功的 8 位单片机的操作系统，代码完全开放。

7. μ C/OS

μ C/OS 是一个特殊风格的嵌入式操作系统，它有多种版本，可以适应从 x86 到 8051 的各种不同类型不同规模的嵌入式系统，代码开放。但是，它的一些改进版本，开始放弃代码开放的原则。

如果只是针对 8051 系列构成的嵌入式系统，显然可以选择的合适的操作系统只有 RTX51 和 μ C/OS 等少数几种。这几种操作系统主要是由于受到 8051 本身资源的限制，功能都相对较为简单。但是，它们开放的源代码和较小的代码量，也给嵌入式系统设计者提供了彻底掌握这一操作系统的条件。

对于 8051 系列来说，用它来构成的嵌入式系统，所承担的任务总是符合 8051 所具备的功能的，不应该将要实现的系统功能目标设计得非常复杂和庞大。因此，这几种操作系统的性能完全能够满足以 8051 系列单片机为核心的嵌入式系统的应用需求。

1.3 嵌入式系统开发技术

1.3.1 单片机开发技术

1. 仿真器

从微处理机诞生至今，开发方式一直沿用的是仿真器技术，即利用一台仿真器，模拟取代应用系统的部分电路，可能是 CPU，可能是程序存储器，也可能是某几个部分电路的组合。通过调试主机对取代电路的控制，可以获得程序运行过程的状态，可以控制程序运行的走向，从而达到调试的目的。单片机出现后，仿真器所取代的往往是包含 CPU、存储器和 I/O 功能模块的调试样机的单片机。但是，所依据的仿真器开发调试技术原理并没有改变。

这种开发方式，是从最低层的软硬件调试开始做起，可以控制到每条指令的执行。因此，控制功能彻底，发现故障的能力也很强，但是调试的效率是不高的。这种开发方式，也较为适用于可以插拔的单片机芯片，对于表面贴装的芯片往往难以实现仿真器与调试样机的方便连接。

2. BOOT ROM

这种方式是在嵌入式系统中事先驻留一个 ROM 引导程序，一般称为 BOOT。开机后首先运行 BOOT 程序，实现与调试主机的联机。有的单片机也采用特殊的系统复位启动过程来激活驻留的 BOOT 程序。

利用 BOOT 程序，嵌入式系统可以通过经通信接口的下载来获得系统的全部程序。BOOT 程序一般在完成下载任务后，就将系统的控制权交给下载生成的系统程序，并且不再起作用了。

BOOT ROM 方式要求嵌入式系统必须起码保证能够运行 BOOT 程序，能够实现代码下载功能，这样才能保证后续程序调试过程的进行。

3. JTAG 接口

所谓 JTAG 技术，最初是一种测试技术。它通过一个标准接口，用串行方式来设置和获得元件的输入和输出信号，从而实现对元件初始状态的控制和运行状态的判断。这个标准接口就称之为 JTAG 接口，基本引脚有 4 个。由于巧妙的设计，具有 JTAG 接口的元件可以将各自的 JTAG 接口串联起来，最终的接口引脚仍然是 4 个。因此，系统内的多个具有 JTAG 接口的元件，最终可以只用一个 4 个引脚的 JTAG 接口来实现全部功能。

对于一个单片机，同样可以为它内部的每一个功能模块各设计一个 JTAG 接口，最终在芯片的引脚上引出一个 4 个引脚的 JTAG 接口来实现对芯片内部各功能模块的控制和测试。利用单片机上的 JTAG 接口，可以设置和检查芯片内各个模块的状态，可以在存储器中写入代码。

因此，利用 JTAG 技术可以实现嵌入式系统的调试，而采用这一方法进行嵌入式系统调试，对调试设备的要求也比较简单。目前常用的方式，往往只需要利用一台 PC 作为调试主机，利用 PC 的打印机接口控制信号，经过逻辑组合来产生直接与嵌入式系统连接的 JTAG 接口。

随着单片机封装的变化，可以插拔的单片机越来越少了，传统的经过插座取代方式的仿真器，已变得使用十分不便，而具有 JTAG 接口的单片机却是越来越普及了。

1.3.2 基于操作系统环境的开发技术

前面叙述的调试方法，基本上是针对一个裸机的开发。调试过程中，需要较多地将注意力集中在系统硬件的故障判断、状态检查及汇编级程序开发调试之上。这些调试方法本身，对于开发一个功能复杂程序的直接帮助是很少的，而设计者总是希望将调试的精力尽量放在系统所要实现的功能上。

如果调试者面对的是一个基本没有硬件故障的嵌入式系统，并且已经有了成熟的底层软件可以利用，这时候的注意力就可以集中在引用系统提供的基本功能来实现应用目标的功能需求上，这实际上是一个操作系统应该提供的功能。操作系统一般都会提供系统硬件的测试软件，帮助设计者确认系统硬件的工作状况，实现故障诊断和定位。操作系统也会提供那些与硬件关系最密切的底层软件，供设计者引用。

随着单片机功能的日益增强，硬件设计技术的日趋成熟，基于嵌入式系统的操作系统

平台，其系统软硬件开发方式也逐渐成熟，而基于操作系统的系统开发调试效率远比其他方法来得高。

1.3.3 嵌入式操作系统环境中的程序设计

操作系统向设计人员提供可引用的系统调用，它们实现了系统的基本功能，一般是以子程序或函数的形式提供的。

嵌入式系统的操作系统所提供的系统调用一般是针对输入输出的基本功能，这些基本输入输出功能可以做到可靠而高效。

在嵌入式系统的开发过程中，如果没有操作系统的支持，面向底层硬件的输入输出功能的开发会占去设计人员的大半开发工作量，并且也很难保证所实现输入输出功能的可靠性。而在嵌入式操作系统的支持下，设计者可以从一组可靠的底层函数出发，开发自己的应用程序，使得所设计的软件一开始就建立在一个可靠的基础上。同时，也使设计者将开发过程中的注意力更多地集中在与应用问题直接相关的软件功能实现上，这会大大减轻程序设计调试的工作量。并且，程序的可移植性也会大大增加，因嵌入式系统硬件的变化和功能升级引起的程序修改工作量可以降到最低。

一个好的操作系统，可以提供对系统安全性、可靠性的保障。利用操作系统的安全性、可靠性保障，应用程序可以获得良好的基础，又不必为此开销大量的代码编写和测试工作量。操作系统总是能提供进程管理功能。这样，应用程序的设计开发将更贴近于系统设计的应用目标要求。同时利用操作系统的进程管理，可以提高嵌入式系统的软件稳定性和可靠性。

总之，在操作系统环境下的开发，为设计者将注意力集中到应用问题提供了完善而良好的条件。

1.3.4 嵌入式操作系统环境下的调试

开发一个嵌入式系统，与在一个通用微机上的开发工作是不同的。由于硬件及软件上的不可靠因素，新设计的嵌入式系统往往要经过细致的时序分析来排除硬件故障，这是一项费时费力的开发工作。在嵌入式系统中引入操作系统，为避免这一工作提供了条件。操作系统的基本目标之一是对底层硬件的封装，操作系统在这一点上一般都做得很完善，它的底层软件是经过了许多人、长时间的运行考验的。操作系统使设计者将注意力更多地集中在要实现的应用目标上，而不是系统硬件的调试上。

在一个共同的操作系统平台上，多进程程序设计、团队式开发组织都成为可能。当引入了操作系统的进程概念，嵌入式系统的应用功能实现就可以分解成许多相对独立的进程，在操作系统的统一管理下完成系统的应用目标。这些进程的设计、调试可以相对独立地进行，通过多人的协同作战来提高开发效率。