

嵌入式系统 工程案例教程

赖晓晨 江贺 韩瑜 编著

- 从工程角度介绍嵌入式系统设计的完整流程
- 内容涵盖从需求分析到测试的全部环节
- 所有案例均经过精心筛选，并已通过调试



机械工业出版社
China Machine Press

高等院校计算机课程案例教程系列

嵌入式系统 工程案例教程

赖晓晨 江贺 殷瑜 编著

- 从工程角度介绍嵌入式系统设计的完整流程
- 内容涵盖从需求分析到测试的全部环节
- 所有案例均经过精心筛选，并已通过调试



机械工业出版社
China Machine Press

嵌入式系统是以应用为中心、计算机技术为基础，软、硬件可剪裁，适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。

本书共6章：第1章介绍几种嵌入式处理器，包括8051单片机、AVR单片机、ARM处理器LPC2136等，对各种处理器的结构、引脚描述、功能模块做了较为详细的描述；第2章介绍嵌入式系统的开发工具与开发环境，以及嵌入式Linux开发的基础知识；第3章以部分典型硬件模块为例，介绍电路仿真工具Proteus的使用方法；第4章介绍一个多核心单片机实验平台的设计实现过程；第5章以嵌入式操作系统FreeRTOS和基于ARM7内核的嵌入式处理器LPC2136为例，介绍操作系统移植过程；第6章介绍一个嵌入式图形系统的设计实现过程。

本书可以作为高等院校计算机及相关专业学生的教材，同时也可以作为相关技术人员的参考用书。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目（CIP）数据

嵌入式系统工程案例教程 / 赖晓晨，江贺，韩瑜编著. —北京：机械工业出版社，2012.9
(高等院校计算机课程案例教程系列)

ISBN 978-7-111-39876-9

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

中国版本图书馆CIP数据核字（2012）第229492号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：马超

三河市杨庄长鸣印刷装订厂印刷

2012年10月第1版第1次印刷

185mm×260mm·17.75印张

标准书号：ISBN 978-7-111-39876-9

定价：35.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com

前 言

嵌入式系统是嵌入式计算机系统的简称,近十几年来,它逐渐为人所知,但这一概念很早就出现了。20世纪70年代以来,以单片机等微控制器为基础的各类设备已经广泛应用于生产当中。随着应用问题的复杂程度不断增加和处理器功能的不断提升,嵌入式系统程序员逐渐在嵌入式硬件平台上引入嵌入式操作系统,使得嵌入式系统逐渐步入成熟期,真正成为一个软硬件结合的“系统”。20世纪90年代以后,系统实时性越来越重要,并且软件规模不断扩大,实时内核逐渐发展为实时多任务操作系统,成为嵌入式软件平台的主流。进入21世纪后,嵌入式处理器的性能不断提升,价格不断下降,各种具有图形用户界面的嵌入式操作系统不断涌现,使嵌入式系统迅速扩展到消费类数码产品领域,走进了千家万户。

嵌入式系统种类很多,存在形态各不相同,其概念经过长期演变,已经趋近于统一。IEEE对嵌入式系统的定义为:“用于控制、监视或者辅助操作机器和设备的装置。”国内普遍接受的嵌入式系统定义为:“以应用为中心、以计算机技术为基础、软件/硬件可裁剪,以及适应应用系统对功能、可靠性、成本、体积、功耗严格要求的专用计算机系统。”在我们身边存在各种嵌入式系统。在消费类数码产品、交通工具、工业测控仪表等领域中,嵌入式系统都处于核心位置,与此形成反差的是,我国每年的嵌入式人才需求缺口均以百万计。因此,加速嵌入式人才培养,提高嵌入式课程体系教学质量,是高校嵌入式专业教师面临的一项重要课题。

在嵌入式系统学习中,需要掌握3个方面的基本内容。首先,应对常用的嵌入式处理器有所了解,熟悉一种或几种典型处理器的使用方法,包括使用该处理器完成基本电路设计以及程序编写等内容。其次,应了解嵌入式操作系统的工作原理,掌握典型嵌入式操作系统的移植过程和驱动程序设计方法。最后,应熟悉一种编程语言,能熟练运用该语言设计嵌入式程序,控制硬件模块运行。需要指出的是,对于实际嵌入式系统来说,嵌入式技术仅是一种实现工具,嵌入式系统的核心问题是特定应用的背景行业相关问题,例如核心算法设计,而非嵌入式计算机技术。但是,即使作为“常规”设计部分,嵌入式技术仍涵盖很多内容,并有大量异于桌面或网络编程的独特之处,需要认真学习,并在实践中不断总结经验。

笔者在多年实践过程中,设计了多个实用嵌入式系统,并已作为案例充实到嵌入式课程教学中。考虑到学习对象应具有典型性,因此,精心选择了4个嵌入式系统作为案例在本书中介绍。在处理器方面,详细介绍应用最广泛的8051单片机和ARM处理器的使用方法;在硬件电路设计方面,阐述了十余种典型电路和接口的工作原理与设计方法;在嵌入式操作系统方面,以FreeRTOS为例,描述了完整的移植过程和驱动设计框架;在硬件仿真方面,介绍了最流行的仿真工具Proteus的使用流程。最后,由于嵌入式图形系统已成为智能手机等消费类数码设备的典型配置,所以介绍了拥有完全自主知识产权的嵌入式图形系统EGUI的架构及设计方案。以上案例的软件部分均由C语言实现,这是目前嵌入式领域的主流编程语言,也是嵌入式程序员必须掌握的基本工具。

本书第1章介绍后续章节涉及的几种嵌入式处理器,包括第4章用到的8051单片机和AVR单片机,第3章和第5章用到的ARM处理器LPC2136等。本章对各种处理器的结构、引脚描述、

功能模块做了较为详细的介绍。

第2章介绍嵌入式系统的开发工具与开发环境,以及嵌入式Linux开发的基础知识。在嵌入式系统硬件设计过程中,需要使用原理图和PCB的绘制工具,本章将介绍Altium Designer的使用方法,第4章的硬件电路设计部分即由此工具实现;在嵌入式软件开发时,需用到集成开发环境,本章将介绍Keil的用法,第3章、第4章和第5章的代码开发即由此工具实现;Proteus是著名的电路仿真工具,本章将介绍其用法,在第3章中会使用Proteus设计多种电路,并完成仿真;本章的后续内容为Linux开发工具链的使用,以及Linux的典型编程方法,这部分内容是第6章的基础。

第3章以部分典型硬件模块为例,介绍电路仿真工具Proteus的使用方法。在嵌入式系统的设计过程中,由于受设计人员的经验限制,硬件模块制作完成后往往并不完全符合要求,此时需要重新设计硬件电路,造成时间、材料方面的浪费,并会影响嵌入式软件开发者的工作进度。采用电路仿真软件,可以在PC上模拟电路的运行,提早发现硬件电路设计隐患,有效降低开发成本。本章介绍了LED点阵、键盘、LCD、脉宽调制电动机控制等多种典型模块的Proteus仿真方法。

第4章介绍一个多核心单片机实验平台的设计实现过程。该平台可以支持8051和AVR两种单片机,包含了十几种典型的硬件模块和接口。本章详细介绍了该平台的需求分析、系统设计、硬件模块详细设计、编码实现等内容,所有电路和代码均经过实际调试,运行稳定、可靠,读者可将这些模块的软硬件设计方案直接应用在自己的系统中。

第5章以嵌入式操作系统FreeRTOS和基于ARM7内核的嵌入式处理器LPC2136为例,介绍操作系统移植过程。本章首先分析了FreeRTOS的结构,然后实现了该系统的全部模块在LPC2136处理器上的移植过程。由于不同的设备有不同的硬件模块,这些模块都需要驱动程序来管理和控制,本章设计了一个符合FreeRTOS的驱动模块框架,并介绍了典型设备的驱动程序设计过程。由于FreeRTOS并不包括文件系统模块,所以本章还介绍了FatFS文件系统的移植过程。本章的最后部分测试了FreeRTOS内核、驱动和文件系统的工作状况。

第6章介绍一个嵌入式图形系统的设计实现过程。随着处理器运算速度的不断提高和功能的不断增强,嵌入式设备已经可以支持大屏幕液晶显示器,由此带来的问题是需要一个统一、易用、高效的图形界面系统。本章介绍嵌入式图形系统EGUI的设计实现过程,该系统采用C/S架构模式,包括客户端模块、服务器模块、绘图模块、控件库模块、交互模块等几个部分,为嵌入式应用程序员提供了一套结构简单、使用方便的轻量级嵌入式图形API,实现了在字符界面Linux操作系统之上显示窗口、按钮、图标等图形功能,可用于图形模式的嵌入式系统设计。

韩瑜编写了第1章和第2章,赖晓晨编写了第4章和第6章,江贺编写了第3章和第5章。此外,罗龙、李瑾、张家宁、刘德超、张树威、朱飞虹等绘制了本书所有图形;谷丹阳、陈向东、丁兰、高畅、陈飞、闫顺、刘鹏、柳继委、孙达等参与了本书实例的硬件设计、代码编写和测试工作;陈扬、祝洪宾、王昭森、余亮、李丽坤、杜春明等参与了书稿的校对工作。感谢以上人员付出的辛勤劳动,是大家的共同努力,才使本书得以面世。

本书实例设计及编写历时将近4年,在这个过程中,得到了很多人的帮助,在此向大家致以衷心的感谢!首先感谢我的家人,是你们教导我做一个人,并引导我走上今天的人生道路;其次感谢大连理工大学软件学院的领导和同事们,是你们为我提供了一个宽松的工作环境,并在书稿撰写过程中提出了很多建设性意见;特别感谢机械工业出版社王春华编辑在本书构思、

立项、审校、出版过程中给予的无私帮助；最后也是最重要的，感谢我的学生（不管是曾经的、现在的，还是将来的），是你们带给了我工作的乐趣和动力！

由于作者经验不足，加之时间仓促，书中不可避免会有谬误之处，恳请读者批评指正。所有关于本书的意见和建议，请发送电子邮件到 far.away@tom.com，希望在和读者交流的过程中能有所裨益。

赖晓晨

目 录

前言

教学建议

第1章 嵌入式系统硬件基础 1

1.1 嵌入式系统组成 1

1.2 8051 系列单片机 3

1.2.1 8051 系列单片机简介 3

1.2.2 AT89C51 的引脚 3

1.2.3 AT89C51 的 CPU 结构 4

1.2.4 AT89C51 的存储器 6

1.2.5 AT89C51 的 I/O 端口 7

1.2.6 AT89C51 的中断系统 8

1.2.7 AT89C51 的定时器/计数器 10

1.2.8 AT89C51 的串行接口 11

1.3 AVR 系列单片机 12

1.3.1 AVR 系列单片机简介 12

1.3.2 ATmega32 的引脚 12

1.3.3 ATmega32 的 CPU 结构 13

1.3.4 ATmega32 的存储器 14

1.3.5 ATmega32 的中断 14

1.3.6 ATmega32 的定时器/计数器 17

1.3.7 ATmega32 的 I/O 端口 17

1.3.8 ATmega32 的串行接口 18

1.4 LPC2136 嵌入式处理器 20

1.4.1 LPC2136 处理器简介 20

1.4.2 LPC2136 的引脚 21

1.4.3 LPC2136 的结构 22

1.5 习题 27

第2章 工具软件与开发环境 28

2.1 Altium Designer 28

2.1.1 Altium Designer 简介 28

2.1.2 原理图设计 29

2.1.3 PCB 设计 32

2.2 Keil 开发环境 37

2.2.1 Keil 简介 37

2.2.2 Keil 工程 38

2.3 Proteus 43

2.3.1 Proteus 开发环境简介 43

2.3.2 基于 Proteus 的电路仿真 45

2.3.3 Proteus 电路仿真设计实例 48

2.4 Linux 工具链 50

2.4.1 vim 51

2.4.2 gcc 53

2.4.3 创建和使用库 54

2.4.4 gdb 58

2.4.5 Makefile 59

2.5 Linux 编程方法 62

2.5.1 文件和目录 63

2.5.2 I/O 操作 66

2.5.3 进程 69

2.5.4 线程 72

2.5.5 套接字 75

2.6 习题 78

第3章 嵌入式系统仿真设计 79

3.1 LED 点阵电路仿真 79

3.1.1 LED 点阵工作原理 79

3.1.2 LED 点阵仿真电路介绍 80

3.1.3 LED 点阵电路仿真程序设计 81

3.1.4 LED 点阵电路仿真结果 82

3.2 键盘电路仿真 83

3.2.1 键盘工作原理 84

3.2.2 键盘电路介绍 85

3.2.3 键盘电路仿真程序设计 86

3.2.4 键盘电路仿真结果 87

3.3 点阵型 LCD 电路仿真 88

3.3.1 PG12864F 模块工作原理 88

3.3.2 LCD 显示电路介绍 93

3.3.3 点阵型 LCD 电路仿真程序
设计 94

3.3.4 点阵型 LCD 电路仿真结果	98	4.4.4 操作流程	154
3.4 脉宽调制器控制直流电动机		4.5 习题	155
仿真	99	第 5 章 嵌入式操作系统移植	156
3.4.1 脉宽调制器工作原理	99	5.1 FreeRTOS 结构	156
3.4.2 脉宽调制器控制直流电动机		5.1.1 FreeRTOS 简介	156
电路介绍	104	5.1.2 FreeRTOS 组织结构	157
3.4.3 脉宽调制器控制直流电动机		5.1.3 FreeRTOS 内核	158
仿真程序设计	105	5.2 FreeRTOS 移植	165
3.4.4 脉宽调制器控制直流电动机		5.2.1 FreeRTOS 移植简介	165
仿真结果	106	5.2.2 启动代码	165
3.5 习题	107	5.2.3 开关中断	169
第 4 章 多核心单片机实验平台	108	5.2.4 临界区的进入与退出	169
4.1 多核心单片机实验平台需求		5.2.5 任务栈初始化	170
分析	108	5.2.6 上下文切换	170
4.1.1 硬件需求分析	108	5.2.7 时钟中断	171
4.1.2 软件需求分析	109	5.3 FreeRTOS 设备驱动程序设计	172
4.2 多核心单片机实验平台系统		5.3.1 FreeRTOS 设备驱动程序设计	
设计	110	简介	172
4.2.1 系统结构	110	5.3.2 设备驱动框架模型	172
4.2.2 处理器资源分配	110	5.3.3 设备驱动框架设计	174
4.2.3 AVR 转接板设计	111	5.3.4 设备驱动框架驱动模块实例	178
4.3 多核心单片机实验平台系统		5.4 FreeRTOS 文件系统	186
实现	112	5.4.1 FAT 文件系统原理	187
4.3.1 基础电路	112	5.4.2 FatFs 应用程序调用接口	189
4.3.2 流水灯模块	114	5.4.3 SPI 驱动模块	190
4.3.3 键盘模块和数码管模块	115	5.4.4 SD 卡驱动模块	192
4.3.4 点阵 LCD 模块	118	5.4.5 FatFs 接口模块实现	200
4.3.5 LED 点阵模块	123	5.5 系统测试	201
4.3.6 温度采集模块	126	5.5.1 测试环境	201
4.3.7 语音模块	129	5.5.2 内核及驱动框架 API 测试	201
4.3.8 继电器模块	136	5.5.3 FatFs 测试	203
4.3.9 串口模块	137	5.6 习题	207
4.3.10 蜂鸣器模块	140	第 6 章 嵌入式图形系统	208
4.3.11 红外模块	142	6.1 EGUI 简介	208
4.3.12 步进电动机模块	146	6.2 EGUI 需求分析	208
4.4 综合实例	149	6.2.1 运行环境需求	208
4.4.1 功能描述	149	6.2.2 核心模块需求	208
4.4.2 软件流程	149	6.2.3 外部接口需求	209
4.4.3 典型代码分析	150	6.3 EGUI 系统设计与实现	211

6.3.1 总体架构	211	6.4 EGUI 客户端编程框架	260
6.3.2 文件组织结构	213	6.5 系统测试	262
6.3.3 基础数据结构模块	216	6.5.1 系统测试之功能需求	262
6.3.4 绘图模块	224	6.5.2 系统测试之系统设计	263
6.3.5 服务器模块	230	6.5.3 系统测试之系统实现	263
6.3.6 客户端模块	241	6.5.4 系统测试之功能测试	266
6.3.7 控件库模块	248	6.6 习题	268
6.3.8 交互模块	257	参考文献	269

嵌入式系统硬件基础

与网络协议栈的组织方式相类似，嵌入式系统也由相互关联的几个层次组成。其中，硬件电路是嵌入式系统的基础，而微处理器是硬件电路的核心。本章按照由总体到局部、由简单到复杂的顺序，依次介绍嵌入式系统的组成、8051 系列单片机、AVR 系列单片机，以及 ARM 嵌入式处理器的组成结构和内部模块。本章内容是后续章节的硬件基础。

1.1 嵌入式系统组成

嵌入式系统一般由嵌入式处理器、外围硬件模块、嵌入式操作系统和嵌入式应用软件等几个层次组成。在设计嵌入式系统时，应根据嵌入式系统的规模、速度、功能特点等性能要求选择合适的嵌入式处理器，根据功能需求选择必需的硬件模块，根据系统的复杂程度决定是否需要移植嵌入式操作系统，根据应用需求设计嵌入式软件。

1. 嵌入式处理器

嵌入式处理器是嵌入式系统的核心部分，可以分为 4 类：嵌入式微处理器（Embedded Micro-Processor Unit, EMPU）、微控制器（MicroController Unit, MCU）、数字信号处理器（Digital Signal Processor, DSP）及片上系统（System on Chip, SoC）。

（1）嵌入式微处理器

嵌入式微处理器由通用计算机的中央处理器演变而来，它只保留和嵌入式应用相关的主要功能，大幅减小系统的体积，并在功耗、工作温度范围、抗电磁干扰、可靠性等方面做了较大优化，以满足嵌入式应用的特殊要求。嵌入式微处理器主要有 Am186/88、386EX、SC-400、Power PC、68000、MIPS 系列等。

（2）微控制器

微控制器即单片机，它基于某种微处理器内核，在片内整合了计算机系统的各个部分，如总线、总线控制逻辑、I/O、串行口、ROM/EPROM、RAM、Flash、EEPROM、定时器/计数器、“看门狗”、脉宽调制（Pulse Width Modulation, PWM）输出、模/数转换（A/D）、数/模转换（D/A）等。各大微控制器设计厂商都提出了自己的产品系列，比较有代表性的包括 8051、MCS-96/196、AVR、C166/167、MC68HC05/11/12/16、68300 等。嵌入式微控制器占据了约 70% 的市场份额。

（3）数字信号处理器

数字信号处理器在系统结构和指令方面进行了特殊设计，特别适合于执行 DSP 算法，在数字滤波、FFT、谱分析等方面得到大量应用。典型的 DSP 包括德州仪器公司设计的 TMS320 系列、摩托罗拉公司的 DSP56000 系列、英飞凌公司的 TriCore 等。

（4）片上系统

随着超大规模集成（Very Large Scale Integration, VLSI）电路技术的发展，工程师可以在一个芯片上集成更多的晶体管，设计单片的复杂系统已经成为可能，这就是片上系统技术，它将一个完整的嵌入式系统设计在一个芯片上，使得嵌入式系统的规模、体积进一步缩减，功能进一步增强，留给系统设计者的工作量进一步减小，系统运行可靠性和效率进一步提高。SoC 的典型产品包括 ARM 系列芯片，片内集成了极其丰富的硬件资源，系统设计者只需配置少量的外围硬件，就可形成完整的嵌入式系统。德州仪器公司的 CC2430 处理器是 SoC 的另一个杰出代表，芯片内部集成了一个 8051 处理器内核，以及一个高效的 2.4GHz 无线收发模块，

并集成了 ZigBee 协议栈，向用户提供模块化编程接口，可以直接用于无线网络应用中，大幅缩短开发周期。

2. 外围硬件模块

嵌入式系统的外围硬件模块视具体应用目的不同而不同，通常包括以下几类：

(1) 电源

嵌入式系统的电源可采用两种方式设计。对于位置固定的设备来说，可以直接从工频电源取电，经降压、整流、滤波、稳压后使用。对于附近没有供电线路的设备，以及手持式嵌入式设备，则可采用电池供电模式。

(2) 输入电路

这部分的功能是将外界信息转换成为二进制信息，并输入计算机。典型模块包括按键、鼠标、触摸屏、拨码开关等。

(3) 输出电路

这部分的功能是将计算机内部的二进制信息转换为人能够接受的形式并输出。典型模块包括 LED、LCD、蜂鸣器等。

(4) 接口电路

这部分的功能是提供设备间的通信接口，完成数据传送。典型模块包括 USB 接口、PS/2 接口、串口、IDE 接口、红外接口、1394 接口、CF 卡接口、网络接口、CAN 总线接口、RS422 接口、RS485 接口等。各种接口功能由处理器内部或外部的控制器实现，用户需读/写控制器完成通信传输功能。

(5) 存储电路

这部分的功能是完成信息存储，包括永久性存储器以及易失性存储器。典型模块包括 ROM、EPROM、EEPROM、Flash 等。Flash 存储器的应用越来越广泛，目前的处理器内部一般都集成了较大容量的 Flash，用于存储程序和需要永久保存的数据，当片内 Flash 容量不够时，可在外围电路中扩展 Flash。

(6) 其他硬件逻辑电路

例如，A/D 转换电路、电动机驱动电路、时钟日期生成电路等。

3. 嵌入式操作系统

嵌入式操作系统是系统软件，一般固化在 Flash 存储器中。常见的嵌入式操作系统有 Linux、uClinux、PalmOS、Symbian、eCos、 μ C/OS II、VxWorks、pSOS、Nucleus、ThreadX、Rtems、QNX、INTEGRITY、OSE、C Executive 等。智能手机和平板电脑近来发展迅速，运行于其上的 Android、iOS、Windows Phone 等嵌入式操作系统正在成为应用热点。

(1) VxWorks 操作系统

VxWorks 由美国风河公司设计，是一款优秀的强实时性嵌入式操作系统，可靠性高，内核可裁剪至很小。在北美地区，VxWorks 占据了嵌入式操作系统的大部分市场，美国的“勇气”号火星车即使用 VxWorks。VxWorks 及相关开发工具的价格很高，小型公司往往难以承受。

(2) 嵌入式 Linux 操作系统

由于具有开源、免费、可移植性好的特点，使 Linux 操作系统不但可以应用于服务器、PC，还可应用于嵌入式设备。Linux 操作系统经过修改后有各种变体，适用于不同环境，如 Android 系统即基于 Linux 内核。学习嵌入式 Linux 要注重 3 个方面：Linux 移植、Linux 驱动设计，以及 Linux 内核裁剪和优化。

(3) μ C/OS II

μ C/OS II 规模较小、结构简单、源码公开，适合初学者学习，很多国内高校以其为代表来开设嵌入式操作系统课程。 μ C/OS II 运行可靠，已经通过美国宇航局认证。

(4) Android

Android 是开源操作系统，以 Linux 内核为基础，由 Andy Rubin 开发，最初主要支持手机，后由 Google 公司收购，逐渐扩展到平板电脑等设备上。2011 年第一季度，Android 的市场份额首次超过塞班系统，跃居全球第一。2012 年 2 月，Android 占据全球智能手机操作系统市场 52.5% 的份额。

(5) iOS

iOS 是由苹果公司设计的手持设备操作系统，属于类 UNIX 操作系统，最初发布于 2007 年，专为 iPhone 使用，后来陆续发布到 iPod touch、iPad 等苹果公司的系列产品中。根据 Canalys 的数据显示，2011 年 11 月，iOS 已经占据了全球智能手机系统市场份额的 30%，美国的市场占有率为 43%。

(6) Windows Phone

Windows Phone 是微软公司的手机操作系统，正式发布于 2010 年，被诺基亚等手机厂商支持，在办公软件、电子邮件、网络浏览器等方面有较好表现，是 Android 与 iOS 的主要竞争对手。

(7) FreeRTOS

FreeRTOS 是完全免费的轻量级操作系统，具有源码公开、可移植、可裁剪、调度策略灵活的特点，可以方便地移植到各种嵌入式处理器。FreeRTOS 提供包括任务管理、时间管理、信号量、消息队列、内存管理、记录功能等在内的各种功能，可基本满足较小系统的需要。FreeRTOS 的突出特点是结构简单、清晰，支持多种处理器，适于作为例子来分析嵌入式操作系统的原理、移植和设计。

4. 嵌入式应用软件

嵌入式系统的应用软件是实现应用功能的关键，一般具有固态化存储、高质量、高可靠、高实时等特点。由于应用软件设计随需求不同而不同，此处不再赘述。

1.2 8051 系列单片机

本节介绍 8051 系列单片机的基本硬件结构，主要包括 8051 单片机的性能特点、引脚配置、微处理器的基本结构和存储器的结构等内容，同时简要介绍 I/O 端口、中断系统、定时器/计数器和串行通信接口等片内资源的工作原理。这部分内容是读者进行后续开发的基础。

1.2.1 8051 系列单片机简介

8051 系列单片机是目前应用范围最广泛的 8 位单片机之一，尤其是 ATMEL 公司的 AT89C51 单片机，采用 Flash 存储技术，可重复编程且功耗小，成为 8051 系列单片机的主流芯片。下面以 AT89C51 单片机为例介绍 51 系列单片机的基本结构。

1.2.2 AT89C51 的引脚

AT89C51 系列单片机的不同型号引脚相互兼容，通常有两种封装形式：双列直插式封装 (DIP) 和方形扁平封装 (PQFP/TQFP)，AT89C51 多采用前者，如图 1-1 所示。

AT89C51 单片机共有 40 个引脚。分别为：2 个电源引脚，4 个控制引脚，2 个时钟引脚，32 个

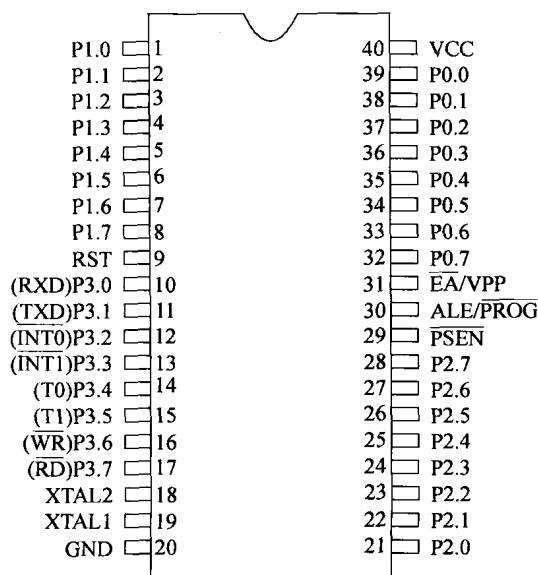


图 1-1 AT89C51 单片机的 40 引脚 DIP 封装形式

I/O 引脚。引脚的功能如下：

(1) 电源引脚

- 1) VCC：正电源端，接 +5V 电压。
- 2) GND：接地端。

(2) 控制引脚

1) RST：复位输入端，高电平有效。当振荡器工作时，若该引脚上持续出现两个机器周期以上的高电平将使单片机复位。

2) ALE/PROG：地址锁存允许信号/编程脉冲输入端。当访问外部存储器时，ALE 为低 8 位地址锁存允许的控制信号，协助 P0 端口实现复用功能。当不访问外部存储器时，ALE 端以振荡器频率的 1/6 输出固定的正脉冲信号，它可用作对外部输出脉冲或用于定时目的。需要注意的是，访问外部数据存储器时，将跳过一个 ALE 脉冲。

PROG 为引脚复用功能，在对 Flash 存储器编程期间，此引脚还用于输入编程脉冲。

3) PSEN：外部程序存储器选通信号。当从外部程序存储器读取指令或数据时，该引脚输出一个负脉冲，用于选通外部程序存储器，该信号在每个机器周期内两次有效。但在访问外部数据存储器时，PSEN 信号无效。

4) EA/VPP：外部程序存储器允许访问控制端/编程电压输入端。当 EA 为高电平时，CPU 默认读取片内程序存储器的指令，若地址范围超过 4KB，则自动转为访问外部程序存储器。当 EA 为低电平时，处理器只访问外部程序存储器。

VPP 为引脚复用功能，在 Flash 编程期间，此引脚也用于施加 12V 编程电压。

(3) 时钟引脚

1) XTAL1：反相振荡放大器的输入及内部时钟工作电路的输入端。当采用内部时钟振荡方式时，该引脚连接石英晶体振荡器和振荡电容。

2) XTAL2：反相振荡器放大器的输出端。当采用内部时钟振荡方式时，该引脚连接石英晶体振荡器和振荡电容。当采用外部时钟源时，XTAL1 接外部时钟源的信号，XTAL2 悬空。

(4) I/O 引脚

P0、P1、P2、P3 四组 8 位 I/O 并行端口。

1.2.3 AT89C51 的 CPU 结构

AT89C51 单片机内部由微处理器、数据存储器（RAM）、程序存储器（ROM）、并行 I/O 接口、串行 I/O 接口、定时器/计数器、中断系统和特殊功能寄存器（SFR）组成。这些部件通过片内单一总线连接而成，其结构仍然采用 CPU 加外围部件的模式，但 CPU 对各功能部件的控制方式是通过特殊功能寄存器集中控制，内部结构如图 1-2 所示。

- 微处理器：8 位 CPU，是单片机的核心部件，由控制器和运算器组成，具有位寻址功能。
- 数据存储器：片内 128B 存储容量，片外最多可扩展 64KB。
- 程序存储器：片内集成 4KB 的 Flash 存储器，片外最多可扩展 64KB。
- 并行 I/O 接口：P0、P1、P2、P3 四组 8 位双向 I/O 接口。
- 串行 I/O 接口：一个全双工 UART（通用异步接收发送器）串行 I/O 口，具有 4 种工作模式，用于实现单片机之间或单片机与微机之间的串行通信。
- 定时器/计数器：两个 16 位定时器/计数器，具有 4 种工作模式。
- 中断系统：5 个中断源，2 级中断优先级中断控制器。
- 特殊功能寄存器：21 个特殊功能寄存器，离散分布在内部 RAM 的 80H ~ FFH 地址单元中。SFR 反映了单片机的运行状态，CPU 通过 SFR 实现对单片机内部各个部件的控制。

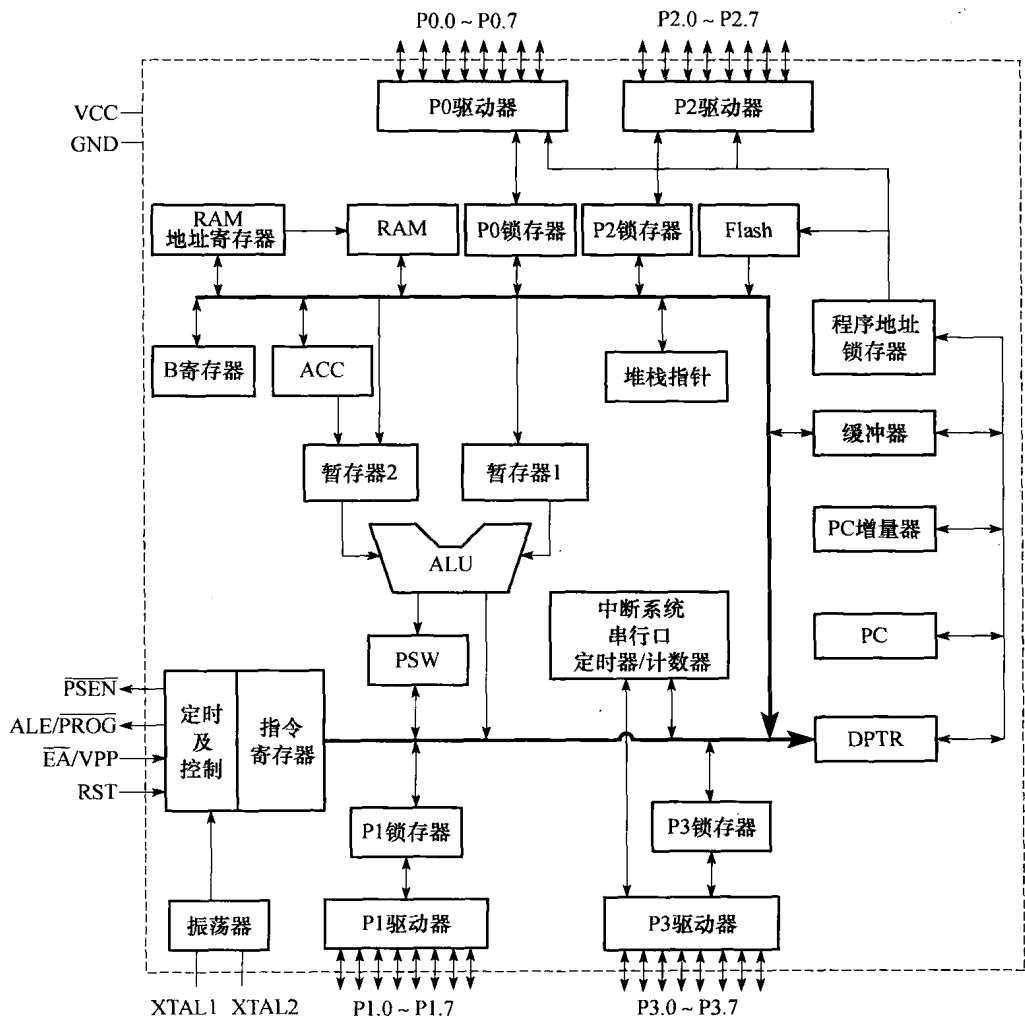


图 1-2 AT89C51 的内部结构

1. 运算器

运算器主要对数据进行算术、逻辑和位操作运算，由算术逻辑单元（Arithmetic Logic Unit, ALU）、累加器 A、暂存器（TMP1 和 TMP2）及程序状态字（PSW）组成。

1) 算术逻辑单元：支持对 8 位数据进行加、减、乘、除等基本算术操作，也能进行逻辑与、逻辑或、逻辑异或、循环、求补、清零等逻辑操作，同时还可对变量进行位操作。

2) 累加器 A：又可记为 ACC，是 CPU 最常用的寄存器，在 CPU 运算时用于存放操作数或结果，同时 CPU 通过累加器 A 与外部存储器、I/O 接口交换信息。

3) 暂存器：负责在单片机进行各种运算时暂存数据，该过程对用户透明。

4) 程序状态字：8 位寄存器，用于存放程序运行的状态信息，可按位寻址，PSW 中各状态位保存当前指令执行后的状态，表 1-1 说明了各标志的具体定义。

表 1-1 程序状态字

标志位	位地址	功能
Cy	PSW. 7	进位标志位
Ac	PSW. 6	辅助进位标志位
EO	PSW. 5	用户设定标志位
RS1	PSW. 4	寄存器组选择位
RS0	PSW. 3	寄存器组选择位
OV	PSW. 2	溢出标志位
—	PSW. 1	保留位
P	PSW. 0	奇偶校验位

2. 控制器

控制器用于识别指令，根据指令控制单片机的各个功能部件协调工作，从而使单片机能实现相应的功能。控制器由程序计数器（PC）、指令寄存器（IR）、指令译码器（ID）、定时控制逻辑电路和振荡器（OSC）等组成。

程序计数器存放将要执行指令的地址，不可访问。程序顺序执行时，PC 自动加 1；程序跳转或调用子程序时，PC 自动更改为指向转移目的地地址。

控制器大体工作流程是，先根据 PC 中的地址把将要执行的指令从存储器中取出，存放在指令寄存器中，然后指令译码器进行译码，定时控制逻辑电路与 OSC 协调工作，产生执行指令所需的全部时序信号。

1.2.4 AT89C51 的存储器

AT89C51 存储器结构采用哈佛结构，即数据存储器和程序存储器分开的结构形式。

1. 程序存储器

片内存储器为 4KB 的 Flash 存储器，地址范围 0000H ~ 0FFFH（地址默认为字节地址）用于存放用户程序和内部常数。外部扩展时，最大支持扩展 64KB 空间，CPU 是否访问外部程序存储器，取决于EA的电平状态。当EA为高电平时，CPU 默认访问片内 ROM，若地址范围超出片内地址范围，则 CPU 自动转为访问片外 ROM；当EA为低电平时，CPU 默认访问片外 ROM。程序存储器结构如图 1-3 所示。

在程序存储器中，某些地址单元具有特殊功能。

- 0000H：CPU 复位后，PC = 0000H，即程序从 0000H 开始执行指令。使用时，通常在这一入口地址处存放一条无条件跳转指令，使程序跳转到用户安排的中断程序起始地址，或者从 0000H 起始地址跳转到用户设计的初始程序地址。
- 以下 5 个地址被固定用于中断源的入口地址。
 - 0003H：外部中断 0 入口。
 - 000BH：定时器 0 溢出中断入口。
 - 0013H：外部中断 1 入口。
 - 001BH：定时器 1 溢出中断入口。
 - 0023H：串行口中断入口。

2. 数据存储器

片内 RAM 共有 128 个地址单元，字节地址范围为 00H ~ 7FH。其中地址 00H ~ 1FH 为 4 组工作寄存器，可通过 PSW 中的 RS0、RS1 来选择当前工作区。地址 20H ~ 2FH 为可位寻址区，指令可以对这 16 个字节单元的 128 位进行按位寻址，对应的位地址为 00H ~ 7FH，同时这 16 个单元也可以按字节寻址。地址 30H ~ 7FH 为用户 RAM 区，用作堆栈、数据缓冲区。外部扩展时，片外最大支持扩展 64KB。数据存储器的结构如图 1-4 所示。

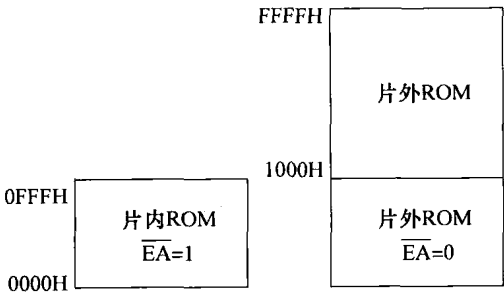


图 1-3 程序存储器结构图

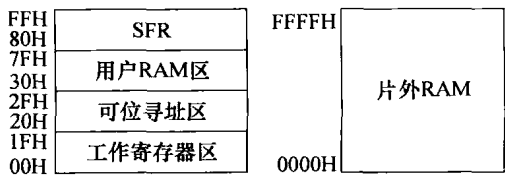


图 1-4 数据存储器结构图

3. 特殊功能寄存器

AT89C51 共有 21 个特殊功能寄存器，分布在内部 RAM 的 80H ~ FFH 地址中。

SFR 反映了单片机的运行状态，这些寄存器的功能已进行了专门的规定，用户不能修改。其中字节地址的末位是 0H 或 8H 的寄存器可按位寻址。SFR 的名称及分布见表 1-2。

表 1-2 特殊功能寄存器定义

符号	名称	字节地址	位地址
B	B 寄存器	F0H	F7H ~ F0H
A	累加器	E0H	E7H ~ E0H
PSW	程序状态字	D0H	D7H ~ D0H
IP	中断优先级控制	B8H	BFH ~ B8H
P3	P3 口	B0H	B7H ~ B0H
IE	中断允许控制	A8H	AFH ~ A8H
P2	P2 口	A0H	A7H ~ A0H
SBUF	串行数据缓冲器	99H	-
SCON	串行控制	98H	9FH ~ 98H
P1	P1 口	90H	97H ~ 90H
TH1	定时器/计数器 1（高 8 位）	8DH	-
TH0	定时器/计数器 0（高 8 位）	8CH	-
TL1	定时器/计数器 1（低 8 位）	8BH	-
TL0	定时器/计数器 0（低 8 位）	8AH	-
TMOD	定时器/计数器方式控制	89H	-
TCON	定时器/计数器控制	88H	8FH ~ 88H
PCON	电源控制	87H	-
DPH	数据指针（高 8 位）	83H	-
DPL	数据指针（低 8 位）	82H	-
SP	堆栈指针	81H	-
P0	P0 口	80H	87H ~ 80H

注：“-”表示不可按位寻址。

表 1-2 中的一部分寄存器（如累加器 A、PSW）在前面小节已经介绍，另一部分寄存器（如 TCON、SCON）将在后续小节介绍。

- 堆栈指针（SP）：8 位寄存器，不可按位寻址。堆栈是一种后进先出的数据结构，设置堆栈主要用于存放临时数据，或者存放中断服务程序和子程序的返回地址。堆栈指针保存了堆栈顶部地址。SP 所指向的堆栈结构属于向上增长型，单片机复位后，SP 初始化为 07H，即从 08H 开始就是堆栈数据区，但这个位置与工作寄存器组 01 相同，为避免冲突，常在程序运行之前把 SP 的值重置，一般设置在 60H ~ 70H 较为适宜。
- 数据指针（DPTR）：16 位寄存器，由 DPH 和 DPL 组成，用来作为对外部 RAM 及 I/O 接口进行寻址的地址指针。

对外部数据寻址时，使用 MOVX 指令，例如：

```
MOVX  A , @DPTR      ;读外部 RAM 的数据
MOVX  @DPTR , A       ;写数据到外部 RAM
```

同时，DPH 和 DPL 也可以作为两个独立的 8 位寄存器来使用。

1.2.5 AT89C51 的 I/O 端口

I/O 端口是单片机对外部电路实现控制和进行信息交换的通路，AT89C51 片内有 4 组 8 位并

行 I/O 端口 P0、P1、P2 和 P3。每组 I/O 端口均为 8 位，每位由锁存器、输出驱动器和输入缓冲器等电路组成。每组 I/O 端口与特殊功能寄存器中的 P0、P1、P2 和 P3 对应，片内 RAM 地址为 80H、90H、A0H 和 B0H。同时，还可对每组 I/O 端口进行按位寻址。当用作数据输出时，锁存器对端口引脚上的数据进行锁存；当用作输入时，缓冲器对端口引脚上的数据进行缓冲。

1) P0 口：对应 P0.0 ~ P0.7。P0 口为 8 位三态 I/O 端口，作为通用 I/O 端口使用，需要加上拉电阻，作为输入时应先把端口的输出锁存器置 1。当访问外部扩展存储器时，可作为地址总线低 8 位与数据总线的分时复用口。P0 口可驱动 8 个 TTL 负载。

2) P1 口：对应 P1.0 ~ P1.7。P1 口为 8 位准双向 I/O 端口，一般作为通用 I/O 端口使用，内部具有上拉电阻，作为输入端口时应先将端口的输出锁存器置 1。P1 口可驱动 4 个 TTL 负载。

3) P2 口：对应 P2.0 ~ P2.7。P2 口为 8 位准双向 I/O 端口，可作为通用 I/O 端口使用，内部具有上拉电阻，作为输入端口时应先将端口的输出锁存器置 1。当访问外部扩展存储器时，可作为地址总线的高 8 位使用。P2 口可驱动 4 个 TTL 负载。

4) P3 口：对应 P3.0 ~ P3.7。P3 口为 8 位准双向 I/O 端口，可作为通用 I/O 端口使用，内部具有上拉电阻，作为输入端口时应先将端口的输出锁存器置 1。P2 口可驱动 4 个 TTL 负载。同时它是双功能复用口，常使用其第二功能，见表 1-3。

表 1-3 P3 口第二功能定义

引脚	第二功能名称	功能
P3.7	$\overline{\text{RD}}$	外部数据存储器读选通
P3.6	$\overline{\text{WR}}$	外部数据存储器写选通
P3.5	T1	定时器/计时器 1 输入端
P3.4	T0	定时器/计时器 0 输入端
P3.3	$\overline{\text{INT1}}$	外部中断 1
P3.2	$\overline{\text{INT0}}$	外部中断 0
P3.1	TXD	串行输出端
P3.0	RXD	串行输入端

1.2.6 AT89C51 的中断系统

CPU 在执行程序的过程中，若出现异常情况或特殊请求，会暂时中止当前的工作，转向执行处理异常或特殊请求的程序（中断服务程序），处理结束后再返回原程序被中止的位置继续执行，这一过程叫做中断。中断系统是指能实现中断处理过程的功能部件和中断处理程序。中断系统使单片机具有实时处理能力，能对外界发生的事件进行及时处理。

AT89C51 的中断系统具有 5 个中断源和 2 个中断优先级，可实现两级中断服务程序嵌套。CPU 支持中断屏蔽和优先级设置，用户可通过程序设置中断允许或屏蔽，以及中断的优先级。

AT89C51 的 5 个中断源如下：

- 外部中断 $\overline{\text{INT0}}$ (P3.2) 和 $\overline{\text{INT1}}$ (P3.3)，中断请求标志为 IE0 和 IE1。
- 定时器/计数器 T0 和 T1 计数溢出中断，中断标志位为 TF0 和 TF1。
- 串口的发送 TXD 和接收 RXD 中断（只占一个中断源），中断标志位为发送中断 TI 或接收中断 RI。

这些中断请求源的标志位分别位于特殊功能寄存器 TCON 和 SOCN 中。

TCON 为定时器/计数器的控制寄存器，字节地址为 88H，可按位寻址。TCON 的高 4 位用于定时器的中断控制，低 4 位用于外部中断的控制，其格式见表 1-4。

表 1-4 TCON 的格式

D7	D6	D5	D4	D3	D2	D1	D0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

- TF1：定时器/计数器 T1 溢出中断请求标志，T1 溢出时，由硬件置位，TF1 = 1；中断响应后，由硬件自动清零，即 TF1 = 0。