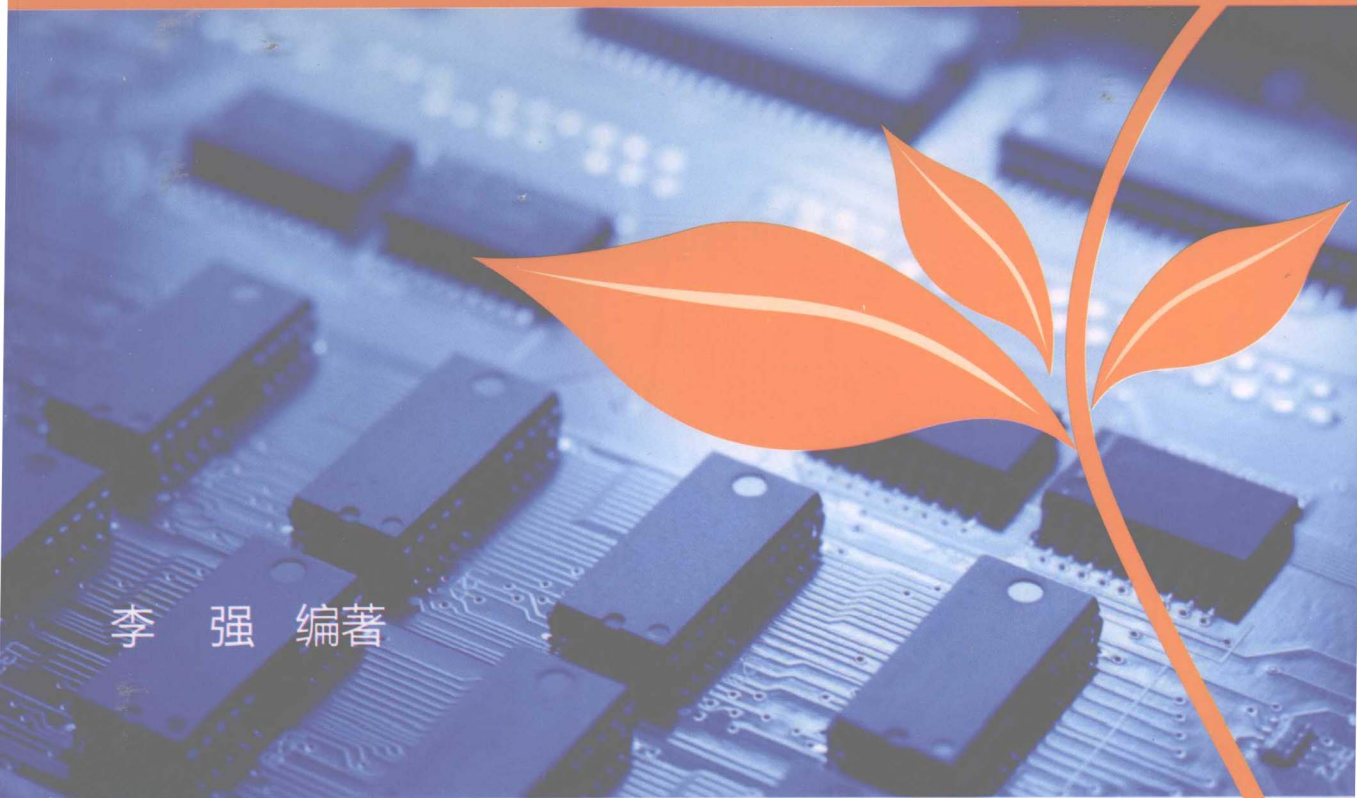




李 强 编著

51 系列单片机 应用软件开发技术



北京航空航天大学出版社

51 系列单片机应用软件 编程技术

李 强 编著

北京航空航天大学出版社

内 容 简 介

嵌入式系统领域里, 51 系列单片机的历史较长, 应用最为广泛。并且, 因其价格低廉, 器件易于采购, 开发工具完善, 而受到众多工程技术人员的青睐。

51 系列单片机体系结构简单, 其应用复杂度适中, 入门容易。因此, 在嵌入式二次开发市场占据很大的份额, 也拥有众多的使用者。

本书以 51 单片机为基本的嵌入式应用平台, 全面研究嵌入式应用程序的编程技术, 探讨嵌入式汇编语言编程、嵌入式 C 语言编程的特点, 以众多的应用实例为背景, 探究嵌入式编程过程中的技术细节。

在本书的汇编语言部分, 以单片机定时器、中断、输入、输出等电路单元的驱动程序设计为实例, 体现出在硬件驱动编程中, 汇编语言为最自然语言的特点。

在本书的 C 语言部分, 主要以过程设计为主线, 探讨 C51 语言软件编程技术。主要内容有: 嵌入式 C51 编译特点、C 源代码的组织方式、开发模式、混合编程、代码运行机制(程序切换)等。

本书可供读者深入学习嵌入式软件编程技术使用, 也可作为高等院校嵌入式系统教学的参考书。

图书在版编目(CIP)数据

51 系列单片机应用软件编程技术/李强编著. —北京:
北京航空航天大学出版社, 2009. 4

ISBN 978-7-81124-624-7

I. 5… II. 李… III. 单片微型计算机—程序设计
IV. TP368.1

中国版本图书馆 CIP 数据核字(2009)第 040187 号

© 2009, 北京航空航天大学出版社, 版权所有。

未经本书出版者书面许可, 任何单位和个人不得以任何形式或手段复制本书内容。
侵权必究。

51 系列单片机应用软件编程技术

李 强 编著

责任编辑 张军香 刘福军 朱红芳

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(100191) 发行部电话: 010-82317024 传真: 010-82328026

http://www.buaapress.com.cn E-mail: bhpress@263.net

北京市媛明印刷厂印装 各地书店经销

*

开本: 787×960 1/16 印张: 28.75 字数: 644 千字

2009 年 4 月第 1 版 2009 年 4 月第 1 次印刷 印数: 4 000 册

ISBN 978-7-81124-624-7 定价: 49.00 元

前 言

嵌入式计算机系统是目前 IP 技术发展的热点之一。随着计算机技术的发展,IP 技术从以 PC 为核心的、通用型、开放式的系统应用阶段,走向以专用计算机为核心的,满足特殊需要的,体积、功耗、性能受约束的嵌入式系统应用阶段。

无论是开放式系统还是嵌入式系统,计算机都是由软、硬件两部分组成的。由硬件实现计算机的基本运算架构,由软件最后确定应用系统的功能。

开放式系统力求计算机硬件组成的通用化,软件开发的标准化。这样,可实现硬件生产批量化,软件开发产业化。以此来降低计算机应用系统的开发、推广成本,实现较高的性能价格比。但这样的模式,使系统的整体优化受到限制。它以最大限度地满足用户基本需求为目的,却很难满足用户的某些特殊需要。直接的表现,就是追求海量数据存储、海量数据处理能力,而将体积和功耗优化降为次要位置。如果应用目标有体积、功耗的限制,则开放式系统的应用效能受限,甚至无法使用。

而嵌入式系统追求的是硬件组成的标准化,软件开发的多样化。硬件组成的标准化,使得嵌入式计算机系统的硬件组成,可以用满足体积、能耗约束的,可以工业化批量生产的、廉价的器件组成;而后,用专用软件确定系统的最后功能,满足最终应用的特殊要求。这种模式,使得用户需求和计算机系统优化达到最佳组合。因此,嵌入式系统可以全方位地嵌入到自然环境和人类社会生活中。未来,它将有更加广泛的应用前景。

对于嵌入式系统而言,软件是其核心内容。软件紧密贯穿整个嵌入式系统体系,决定着它的功能、性能。由于嵌入式系统的软件具有多样性的特点,软件开发时,涉及汇编语言编程技术、嵌入式 C 语言编程技术、系统管理软件编程技术、嵌入式操作系统的应用编程技术、嵌入式数据库技术,甚至,还涉及软件工程等诸多方面。本书及其姊妹篇《51 系列单片机管理软件编程技术》一书(北京航空航天大学出版社将后续出版),从上述多方面对软件编程技术进行详细的专题讨论。

编程技术是要讲究细节的,因此,本书力图以单片机应用实例为背景,对单片机软件编程问题进行实例化、系统化的讨论,并尽量给出完整的源程序代码和完整的编程注释,从而避免过多繁琐、抽象的理论解释。

软件的多样化是指软件本身具有灵活性、多样性,可以满足控制目标千变万化的需求,还可以代替一些硬件的功能。但就软件编程技术本身而言,还是可以找到软件开发中的一些规律、规则和规约的。此所谓万变中的不变,如果掌握了软件基本的编程技术,那么,就可以以不变应万变。这也就是单片机软件编程技术的魅力之所在。



本书采用的实例尽量通用化、实用化和简单化,重点在于编程技术的解析,以减轻读者的阅读负担。注意,本书不是一本语言教科书,是一本分析编程技术的书,因此,在阅读本书时,手边最好有一本 C 语言或者单片机方面的教科书,以方便查阅。

我要感谢所处的时代,从嵌入式系统萌芽阶段起,那时候,是单板机、单片机品种化比较单一的应用时代,及至今天,是 8 位单片机、16 位单片机、32 位单片机、片上系统——SOC(System of Chip)的多样化、多品种时代,二次应用开发工具也日渐成熟。每一阶段都亲身所历。今天,正是嵌入式系统应用走向快步发展的时代,我们已经有了很多优秀的嵌入式应用开发工具,如各种嵌入式 C 语言、C 语言编译器、各种实时操作系统、各种单片机软、硬件仿真软件等。这些工具,使得嵌入式软件开发效率得到了极大的提高。也使我可以方便地表述上述编程问题,最终写成本书。我要对我们同时代的所有同仁表示敬意。希望,本书对嵌入式系统在中国的推广有所裨益。

霍静波和李林参与了本书的编写工作,在此表示衷心的感谢。

目 录

第 1 篇 汇编语言程序设计

引 言	1
第 1 章 深刻理解 51 单片机的指令系统	3
1.1 概 述	3
1.2 单片机指令级逻辑结构	4
1.3 单片机的典型存储结构	8
1.4 指令系统分析	9
1.4.1 指令体系结构	9
1.4.2 指令级计算机的工作机理	10
1.4.3 指令级数据表示	14
1.5 指令编码	17
1.6 寻址方式	20
1.7 指令功能分析	22
1.7.1 数据传送类指令	22
1.7.2 算术运算类指令	25
1.7.3 逻辑运算类指令	26
1.7.4 控制转移类指令	28
1.7.5 位操作类指令	31
1.7.6 控制转移指令的复合应用	34
第 2 章 查表和散转程序设计技术	40
2.1 查表程序设计技术	40
2.1.1 查表编程概述	40
2.1.2 基本查表编程技术	41
2.1.3 扩展查表编程技术	42
2.1.4 扩展数据表查表编程技术	43



2.1.5 数据表元素扩展查表编程技术	44
2.2 散转程序设计技术	45
2.2.1 应用软件的整体构造	45
2.2.2 散转基本编程技术	47
2.2.3 散转程序的扩展编程技术	49
2.2.4 基本查表散转程序编程技术	50
2.2.5 标记查表法散转程序编程技术	51
2.3 AT24XX 系列存储芯片编程	54
2.3.1 AT24 系列存储芯片的特性	54
2.3.2 AT24 系列芯片的读/写操作	55
第3章 中断程序和单片机软件抗干扰程序设计技术	59
3.1 中断概述	59
3.2 中断的基本编程技术	61
3.3 “定时查询”中断扩展编程技术	63
3.4 中断的“外部中断查询”扩展技术	65
3.5 中断编程实例	66
3.6 单片机软件抗干扰技术概述	70
3.7 抗系统“死循环”技术	71
3.7.1 基本软件看门狗技术	71
3.7.2 高级软件看门狗技术	72
3.8 抗系统“跑飞”干扰技术	75
3.8.1 指令冗余法	75
3.8.2 软件陷阱技术	76
第4章 定时/计数器编程技术	77
4.1 软件定时器设计技术	77
4.1.1 软件定时器的应用原理	77
4.1.2 软件定时器的设计	78
4.1.3 软件定时器的应用实例	80
4.2 硬件定时器设计技术	81
4.2.1 硬件定时/计数器的工作机理	81
4.2.2 实时定时/计数器的编程	83
4.3 超长时间实时定时器编程技术	87

4.4 单片机计时综合应用编程实例	88
4.5 软件运行时间分析	93
4.6 DS1302 时钟芯片的应用	97
第 5 章 显示接口编程技术	105
5.1 LED 的显示驱动	105
5.2 独立式 LED 显示器	106
5.2.1 程序逻辑控制法	107
5.2.2 查表逻辑控制法	107
5.3 LED 矩阵显示器	109
5.3.1 程序逻辑控制法	110
5.3.2 查表逻辑控制法	111
5.4 数码管显示器	113
5.4.1 数码显示器静态工作编程模式	116
5.4.2 数码显示器动态工作编程模式	118
5.5 液晶显示器驱动编程	120
5.6 显示编程应用实例	127
第 6 章 键盘接口编程技术	134
6.1 常用键盘的结构	134
6.1.1 独立式键盘	134
6.1.2 行列式键盘	135
6.2 独立式键盘接口的通用编程模式	137
6.3 行列式键盘接口的通用编程模式	138
6.3.1 扫描法	139
6.3.2 反转法	141
6.4 键盘接口的工作方式	143
6.4.1 键盘完全监控工作方式	144
6.4.2 键盘中断工作方式	144
6.5 键盘编程实例	146
第 7 章 汇编语言编码风格	156
7.1 汇编伪指令	156
7.2 汇编编码风格	160



7.2.1 软件格式	160
7.2.2 符号约定	165
7.3 模块化程序设计	169
7.4 汇编程序设计范例	174

第 2 篇 C51 程序设计

引 言	203
-----------	-----

第 8 章 C51 语言基础	204
----------------------	-----

8.1 C 语言概况	204
8.2 C51 语言的特点	206
8.3 C51 程序的组成	210
8.4 C51 源程序的基本语法	218
8.5 Keil 的基本使用	222
8.6 单片机程序设计方法	230
8.6.1 程序设计基本方法	230
8.6.2 结构化程序设计	231

第 9 章 C51 程序设计基础	240
------------------------	-----

9.1 程序的输入/输出	240
9.2 C51 数据类型	245
9.3 伪随机数发生器	249
9.4 C51 运算符	257
9.5 ANSI C 标准测试	260
9.6 C51 程序的基本设计技巧	261

第 10 章 C51 编码风格	266
-----------------------	-----

10.1 编码风格概述	266
10.2 编码风格实例解释	266
10.2.1 源码组织实例	267
10.2.2 实例源码分析	269
10.3 标识符命名法	283
10.3.1 匈牙利法	283

10.3.2	骆驼法	284
10.3.3	下划线法	284
10.3.4	标识符缩写	285
10.3.5	变量命名原则	285
10.4	源代码的组织方式	287
10.4.1	基本方式	287
10.4.2	功能分组方式	288
10.4.3	人名分组方式	299
第 11 章	数组、结构与指针	300
11.1	概 述	300
11.2	数 组	301
11.2.1	一维数组	301
11.2.2	多维数组	308
11.2.3	数组作函数的参数	310
11.3	结 构	311
11.3.1	结构定义	311
11.3.2	结构变量的声明	313
11.3.3	关键字 typedef 的用法	314
11.3.4	结构变量的引用	315
11.3.5	结构变量初始化	315
11.4	结构数组	316
11.4.1	结构数组的定义	316
11.4.2	结构数组的应用	318
11.4.3	结构数组的引用	319
11.4.4	结构作函数参数	319
11.5	指 针	321
11.5.1	指针概念	321
11.5.2	指针的应用	322
11.6	指针应用	327
11.6.1	数组指针	327
11.6.2	结构指针	329
11.6.3	函数指针	332
11.6.4	指针型指针	336



第 12 章 C51 编程模式及程序测试	338
12.1 增量式编程模式	338
12.1.1 选择排序模块的实现	339
12.1.2 希尔排序模块的实现	342
12.1.3 快速排序的实现	346
12.2 排序系统源码的组织方式	350
12.2.1 源代码的人名组织方法	350
12.2.2 头文件编制	351
12.3 C51 程序调试	356
12.3.1 C51 调试环境设置	356
12.3.2 基本调试功能(技术)	358
12.4 C51 程序的排错	367
12.5 C51 程序的测试	375
12.5.1 黑盒法	375
12.5.2 白盒法	383
第 13 章 C51 混合编程技术	386
13.1 参数传递	386
13.2 C51 中的“段”	388
13.3 应用实例	392
13.3.1 无参混合编程(精确延时)	393
13.3.2 有参混合编程(硬件驱动)	398
13.3.3 指针参数混合编程	412
第 14 章 程序切换技术	422
14.1 软件系统运行管理	422
14.1.1 程序切换	422
14.1.2 系统堆栈	423
14.2 子程序切换	424
14.2.1 子程序切换原理	424
14.2.2 子程序的应用	426
14.2.3 系统堆栈的讨论	427
14.2.4 堆栈的应用——现场切换	428

14.3 函数切换·····	429
14.3.1 C51 函数工作原理·····	430
14.3.2 C51 函数扩展·····	432
14.3.3 软件堆栈(模拟堆栈)·····	434
14.4 中断切换·····	437
14.4.1 中断工作机理·····	437
14.4.2 中断现场保护·····	439
14.4.3 C51 中断编程·····	440
14.5 任务切换·····	442
14.5.1 任务的特点·····	442
14.5.2 任务的组成原理·····	443
14.5.3 任务的工作机制·····	444
参考文献·····	447
编后记·····	448

第 1 篇 汇编语言程序设计

引 言

迄今为止,笔者依然认为,汇编语言是理解单片机系统或者计算机系统工作机制的一把利器,也是进行单片机软件开发的一个强大的工具。因为,汇编代码直接工作在单片机的机器层次上,它将单片机看成是可以自动执行指令的机器,用软件去给它智慧,使它能够完成目标管理、自动控制等应用工作。

在硬件驱动的编程中,汇编语言是最自然的语言。虽然,在高级语言中,如 C51 语言中,提供了预定义的符号,如 P0、P3.0 等,还有宏和关键字,如 PBYTE、AT 等,用户可以直接用这些符号、宏或关键字,在源程序中去描述硬件、访问硬件。但是,抽象层次较高的高级语言,毕竟减弱了编程者对硬件工作机制的洞察力。在有实时性响应要求、或者时间精度要求较高的场合,或者进行外围芯片驱动时,高级语言显得并不那么直接,此时,汇编语言却是对这种场合的自然表述工具。

另外,高级语言要求编程者对编译器等编程工具要有一定的理解,对编程者的软件知识、软件能力方面有一定要求。而用汇编语言可避免这些繁琐,只要理解了单片机的基本工作机制,直接使用它,就可入门编程。

编制较简单的单片机软件时,或者在编制接口驱动程序时,采用汇编语言不失为一个好的选择。

在软件的移植应用方面,汇编语言也是不可缺少的工具。比如各种实时操作系统的移植中,那些与硬件紧密相关的代码部分,或者高级语言编译器所不能支持的代码部分,还是需要用汇编语言来编制。

因此,汇编语言是单片机中最基本的编程语言,是我们理解单片机工作机制的重要手段,也是理解单片机各种接口的软件驱动和基本算法实现的基础,更是解决软件编程、软件调试问题的最后手段。在单片机软件设计中,它有不可替代的作用。掌握一种汇编语言,是掌握单片机技术或者其他计算机技术的一把钥匙。

本篇对汇编层次上的单片机基本编程技术进行了全面、系统的探讨。除了最基本的汇编语言章节外,本篇中的其他各章中,都给出了综合性较强的编程实例。给出完整的软件代码,并尽量给出全面的注释。目的在于揭示汇编语言编程中的一些技术细节问题,同时向读者介



绍一些常用器件接口的编程,如 DS1302、LCM1602 的接口编程等。对这些内容的充分理解,有助于学好、用好高级语言——C51 语言。

在程序编制过程中,为了清晰地反映编程技术,一些程序片断没有考虑代码优化问题,但在今天这样的编程技术环境下,以可读性作为软件设计的第一追求目标,可能更加合理。因为,市场上有很多质高价廉的器件供我们选择,不仅可使用普通 8 位机,也可以考虑高速器件,甚至高档器件,如 16 位、32 位单片机等,以解决 CPU 的速度和存储容量问题,这并不一定会增加太多的硬件成本。因此,应用系统的软、硬件之间的界线,如果能够合理地划分,就可以降低软件开发难度,保证开发质量。一味地追求代码优化,并不可取,毕竟,许多的代码优化措施是以牺牲可靠性和可维护性为前提的。希望本篇对读者在编程技术的提高上有所启示。

第 1 章

深刻理解 51 单片机的指令系统

指令系统是单片机最底层的人机界面,读者对某种单片机的应用开发能力,很大程度上取决于对指令系统的理解。本章将对 51 单片机指令系统进行详尽的分析。

1.1 概 述

单片机实质上是一种“时序逻辑电路+指令系统”,是各种外围功能电路尽量在芯片内集成的 IC 器件。由于这种器件有指令系统,因此,它有计算机的特性;又由于它多用于目标控制场合,所以,又称微计算机控制器,简称微控制器。目前,在英文技术资料里将其称为 Micro-controller 或称 Micro Control Unit——微控制单元,简称 MCU。

这种器件可以方便地嵌入到控制目标中,比如,嵌入到仪表中、汽车里,或者空调内,所以又称嵌入式计算机系统——Embedd Computer System。它的出现,改变了传统的电子设计方式,使得过去由多个电路器件才能够完成的电子系统,用一个单片机和少数几个外围器件集成就可以实现。简化了电子系统的设计,并且提高了电子系统的可靠性。

单片机具有计算机的特性,可以满足控制目标的数据运算、数据处理的要求;另一方面,作为电子器件,它也是一种智能器件,它的硬件功能也是可以软件设定的,开发者可以用编程的方法,对其器件功能进行有限设定,以便灵活地满足控制目标的需求。

单片机作为一种半导体器件,它的功能相对固定、通用,以满足工业化批量生产的要求,但它又有指令系统,其最后功能可由指令软件来确定。因此,器件的通用性和指令的灵活性的结合,使得用户可以使用工业化批量生产的廉价器件,来构建符合产品要求的智能平台,并用指令软件来实现最后的具体控制要求。这是单片机应用技术的内涵。伴随着这项技术的广泛应用,嵌入式系统应用技术得到蓬勃发展。至今已经成为计算机技术的一个重要分支。

由于有指令系统,单片机能够很容易地实现数据的数值运算和逻辑运算功能,还能够实现由纯硬件电路难以实现的各种复杂的控制功能。如 PID 控制、模糊控制等。因此其应用愈来愈广泛,有越来越多的人期望加入单片机开发应用者的行列,并掌握单片机开发技术。

如上所述,单片机的软件是单片机控制系统的核心。单片机控制系统功能的最终实现,就体现在软件上。因此,单片机软件的开发是单片机系统开发的核心。夸张一点说,对单片机控制系统开发而言,硬件开发的工作量,是器件采购;软件开发的工作量,就是编制程序。



不管是 C 语言软件开发,还是汇编语言软件开发,软件的基础是指令系统。因此,不管是汇编程序员,还是 C 程序员,深刻地理解一种单片机的指令系统,是掌握软件开发技术的一个重要方面。对于单片机开发者而言,指令系统的理解,是全面掌握单片机开发技术,或者说是完整掌握嵌入式系统开发技术的关键。

汇编语言是与指令系统紧密相关的编程语言,汇编语言编程是以指令系统为核心进行的,程序员按控制系统的目标要求,直接使用指令系统所提供的指令进行程序编制,完成目标的控制需求。汇编语言级软件开发技术,是单片机软件开发的最基础的技术。这种技术,现在依然有着广泛的应用。

要掌握单片机指令系统,汇编语言是一把钥匙,但要掌握好汇编编程技术,就应当理解单片机的工作机理和单片机与其他器件的接口工作方式。汇编语言是理解单片机的基础,亦是单片机软件开发的基础,好的汇编程序员不一定是好的 C 程序员,但好的 C 程序员一定具有汇编方面的基本知识。

要掌握汇编语言级单片机的开发技术,首先要具备一定的电子技术知识,特别是要掌握一些有关电子器件、特别是集成电路器件方面的知识,了解器件市场,掌握单片机系统的硬件设计集成技术;在此基础上,最好能够对单片机的指令系统有比较透彻的理解,以掌握单片机软件开发技术。

用户对单片机指令系统的理解主要涉及以下两个方面:一是对单片机指令系统本身的理解,比如,指令系统的体系结构、指令类别、指令编码格式、指令条数、寻址方式、具体指令的理解、基本编程技术等;二是对单片机的指令级硬件逻辑结构的理解,如基本硬件配置、基本数据宽度、存储器编址方式等,也就是对指令系统所面对的基本硬件平台的理解。本章向读者全面剖析 51 系列单片机的指令系统。

1.2 单片机指令级逻辑结构

1. 软件构成方式

指令系统是直接面向计算机硬件电路的,汇编指令级软件开发,不但要面对控制目标,解决控制目标的数据处理问题,而且,还要使所开发的程序,所构造的数据,适应计算机顺序执行、顺序存储的工作特点。因此,用户在开发软件时,要解决两个方面的问题,一是完成目标控制,二是完成单片机系统本身的管理。如果控制系统比较复杂,控制软件规模比较大,为了解决上述两方面的问题,还需要将控制软件分为应用软件和系统软件两个层次,如图 1.1 所示。

系统软件完成单片机系统本身的管理、监控,使单片机各个部分协调工作,比如 CPU 管理(进程管理)、存储管理、中断管理、外设管理、网络管理等;应用软件完成控制系统的目标管理,比如,控制键盘输入,显示器输出,控制电机转速等。

在单片机系统规模比较小时,系统管理比较简单,系统管理方面的功能,程序员可以自己

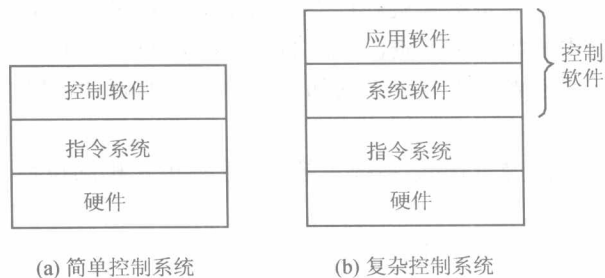


图 1.1 单片机控制软件的构成

编制程序来实现。但对功能复杂、规模较大的控制系统,采用实时操作系统(RTOS——Real Time Operation System)来完成系统本身的管理和各个部分的协调工作,是程序员最好的选择,目前已经有很多可供选择的嵌入式实时操作系统,如 VxWorks、 μ C/OS-II、RTL Linux 等。无论采用什么方式构造控制软件,采用什么样的语言编制程序,程序员对单片机系统的基本硬件环境要有很透彻的理解,这就要求首先要了解单片机的基本硬件逻辑结构。

2. 硬件系统的逻辑组成

所谓单片机的硬件逻辑组成,是指从指令级系统层次来看单片机的组成,它不是指单片机内部具体的微电子电路组成,而是指令级的逻辑组成,是执行指令的硬件单元,也是用指令能够驱动、访问的最低硬件逻辑。这也是单片机二次开发者,所能够驱动的最低层次。它包括两个方面的内容,一是单片机芯片内部的逻辑功能电路配置,包括中央处理器、通用逻辑单元(如中断系统、定时/计数器等、存储器、I/O 接口电路等)和专用逻辑单元(A/D 转换电路、比较电路、看门狗电路等)。二是外围扩展的逻辑单元,也分通用、专用两类功能电路,需要在单片机外部用专用芯片扩展。51 系列单片机芯片内硬件逻辑功能结构如图 1.2 所示。

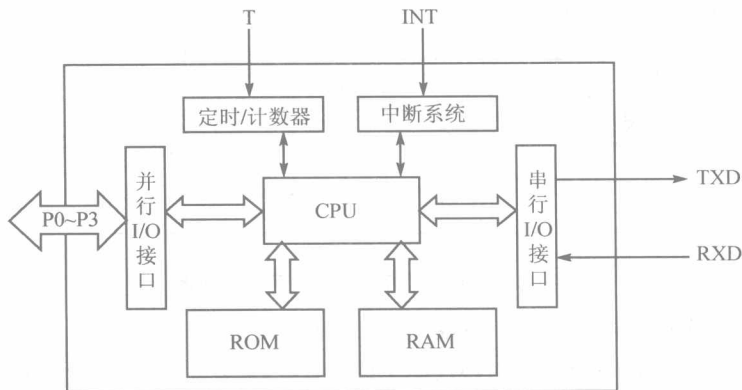


图 1.2 51 系列单片机内部硬件逻辑功能图