嵌入式单片机开发与应用

刘理云 主编

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

嵌入式单片机开发与应用

主　编　刘理云

北京理工大学出版社
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内容提要

目前越来越多的人使用C语言开发单片机应用系统。本书共分7章，第1章介绍了微型计算机基础知识与单片机的基本结构及工作原理，第2章介绍了C51程序设计，第3章介绍了开发编译环境和仿真设计技术，第4章介绍了中断系统、定时/计数器和串行接口等片内功能部件，第5章介绍了C51单片机应用系统扩展设计基础，第6章介绍了C51单片机的基本接口应用技术，第7章介绍了单片机应用系统设计基础。本书例题丰富，每章后面都安排了思考与练习题，便于学生复习、巩固和训练提高。

本书可以作为电子类、机电类和计算机类各专业单片机技术课程的教材，也可作为51系列单片机C程序设计开发的技术人员及高等学校相关专业师生的参考用书。

图书在版编目（CIP）数据

嵌入式单片机开发与应用/刘理云主编. —北京：北京理工大学出版社，2016.1
ISBN 978-7-5682-1115-4

Ⅰ.①嵌…　Ⅱ.①刘…　Ⅲ.①单片微型计算机　Ⅳ.①TP368.1

中国版本图书馆CIP数据核字（2015）第195242号

出版发行／北京理工大学出版社有限责任公司
社　　址／北京市海淀区中关村南大街5号
邮　　编／100081
电　　话／（010）68914775（总编室）
　　（010）82562903（教材售后服务热线）
　　（010）68948351（其他图书服务热线）
网　　址／http：//www.bitpress.com.cn
经　　销／全国各地新华书店
印　　刷／北京泽宇印刷有限公司
开　　本／787毫米/1092毫米　1/16
印　　张／15.5
字　　数／345千字
版　　次／2016年1月第1版　2016年1月第1次印刷
定　　价／48.00元

责任编辑／封　雪
文案编辑／封　雪
责任校对／周瑞红
责任印制／李志强

前言 Preface

现代生产和生活越来越智能化、信息化，且智能化和信息化技术的要求不断提高。单片机作为智能控制的核心，似乎已经没有不使用它的领域了，因此社会需要大量学习和掌握单片机应用技术的专业人才。随着单片机开发技术的不断发展，目前越来越多的人使用C语言编写单片机应用系统程序。

单片机是在一块半导体材料上集成了CPU、存储器、I/O接口等各种功能部件，其应用系统综合性很强，涉及模拟电子技术、数字电子技术、集成电路芯片、传感器、测控技术、程序设计和电路设计，需要使用多种EDA软件。学习单片机原理及应用技术课程，能够促使我们将这些知识融会贯通，自然而然提高解决实际工程技术问题的能力。

本书以C语言作为程序设计语言，以Keil μVision为开发平台，从实用角度出发，较详细地介绍了C51单片机的结构及工作原理、C语言及程序设计方法、系统功能扩展基础、单片机接口技术、应用系统设计方法及仿真软件Proteus ISIS。以培养应用型人才为目标，更新教学内容和教学方法，力求结构新颖、合理，概念条理清晰，应用举例实用；在叙述上深入浅出、通俗易懂，便于读者学习和掌握。

本书共分7章。第1章介绍了微型计算机基础知识，51单片机的内部结构及功能，51单片机的封装和引脚功能、最基本的应用系统——最小系统的电路组成，讨论了时钟和时序以及复位问题。第2章介绍了C语言基本知识及程序设计方法。第3章介绍了Keil μVision开发平台和仿真软件Proteus ISIS。第4章介绍了C51单片机的中断系统、定时/计数器和串行接口的基本概念、结构及工作原理，详细讲述了与中断系统、定时/计数器和串行接口有关的特殊功能寄存器；分析了中断响应过程、外部中断源的扩展方法；分析了定时/计数器、串行接口的各种工作方式以及定时/计数器计数初值的计算方法。第5章介绍了C51单片机应用系统扩展设计基础。第6章介绍了C51单片机的基本接口应用技术，包括开关量接口、键盘接口、显示接口、A/D转换和D/A转换接口的基本方法以及应用实例。第7章从总体设计、硬件设计、软件设计、可靠性设计、系统调试与测试等几个方面介绍了单片机应用系统设计的方法及基本过程，并给出典型设计实例。

本书由刘理云老师编写。在编写过程中，参阅了许多文献资料。编者在此谨向各位作者表示诚挚的感谢。

单片机和电子技术的知识发展迅猛，涉及应用面广，知识更新快，加上编者知识和水平有限，虽经努力，但书中仍难免有疏漏之处，恳请广大读者批评指正。

编　者

目录 Contents

第1章

单片机概述与基本结构

内容提要：本章在介绍了微型计算机的基本结构及主要技术指标、计算机中的数和编码之后，对单片微型计算机做了概述，并介绍了单片微型计算机的发展趋势和应用范围。了解单片机的封装、引脚功能、内部结构及功能是应用单片机的基础。本章第2节先介绍了51单片机的内部结构及功能，51单片机内部由CPU、振荡与时钟电路、程序存储器、数据存储器、定时/计数器、串行口、并行口、总线扩展控制、中断组成；着重讨论了程序存储器、数据存储器、并行口的结构及应用；还介绍了51单片机的封装和引脚功能、最基本的应用系统——最小系统的电路组成，讨论了时钟和时序以及复位问题。

1.1 微型计算机基础知识

1.1.1 微型计算机的基本组成及主要技术指标

1. 微型计算机的基本组成

迄今为止，所有计算机的组成结构都是冯·诺依曼型的，即它是通过执行存储器中的程序而工作的。微机系统包括软件和硬件，如图1.1.1所示。软件分为系统软件、应用软件和程序设计语言。硬件包括主机和外围设备，主机包括CPU、存储器、输入/输出接口、总线；外围设备包括外部设备与辅助设备，其中外部设备包括输入设备、输出设备和辅助存储器，辅助设备包括电源电路和时钟电路。

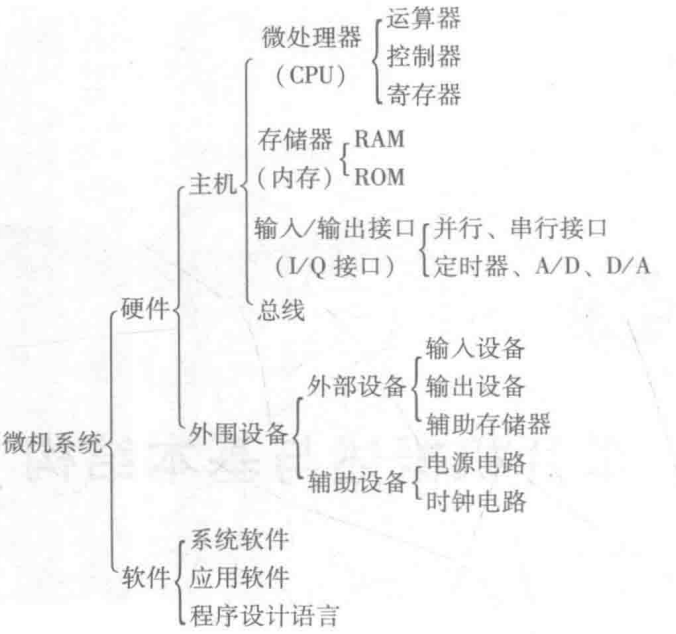


图 1.1.1　微机系统的组成

CPU 是计算机的控制核心，它的功能是执行指令，完成算数运算、逻辑运算，并对整机进行控制。存储器用于存储程序和数据。输入/输出接口（又称 I/O 接口）是 CPU 和外部设备之间相连的逻辑电路，外部设备必须通过接口才能和 CPU 相连，不同的外部设备所用接口不同，每个 I/O 接口也有一个地址，CPU 通过对不同的 I/O 接口进行操作来完成对外部设备的操作。存储器、I/O 接口和 CPU 之间通过总线相连。用于传送程序或数据的总线称为数据总线（Data Bus）；地址总线（Address Bus）用于传送地址，以识别不同的存储单元或 I/O 接口；控制总线（Control Bus）用于控制数据总线上数据流送的方向、对象等。

2. 微型计算机的主要技术指标

微型计算机主要有如下一些技术指标。

（1）字长：CPU 并行处理二进制数据位，由此定为 8 位机、16 位机、32 位机等。

（2）存储容量：存储器单元数，如 256 B、8 KB、1 MB 等（1B 即一个字节，也就是一个 8 位二进制数，是计算机数据的基本单位）。

（3）运算速度：CPU 处理速度，它和内部的工艺结构以及外接的时钟频率有关。

（4）时钟频率：在 CPU 极限频率以下，时钟频率越高，执行指令速度越快。

1.1.2　计算机中的数和编码

1.1.2.1　常用的数制

所谓数制是指数的制式，是人们利用符号进行计数的科学方法。数制有很多种，在计算机的设计与使用中常用到的有十进制、二进制和十六进制。

1. 十进制（Decimal）

十进制数由 0，1，2，3，4，5，6，7，8，9 十个数码组成，按照“逢十进一”的原则计数。任何一个十进制数不仅和构成它的每个数码本身的值有关，而且还和这些数码在数中的位置有关，这就是说，任何一个十进制数都可以展开成幂级数形式。例如 6968 可以写成

$$6968 = 6 \times 10^3 + 9 \times 10^2 + 6 \times 10^1 + 8 \times 10^0$$

上式称为按权展开式。各位数的权值为10的幂，即个位的权为10^0，十位的权为10^1，百位的权为10^2等。例如第四位数字6表示6000，第二位数字6表示60，是因为它们所在位的权不同。

2. 二进制（Binary）

二进制数只有两个数码，即0和1，在电子电路中很容易实现。例如，可以用高电平表示1，用低电平表示0。采用二进制，就可以方便地利用电路进行计数工作，因此，计算机中常用的进位制是二进制。二进制采用“逢二进一”的原则，各位的权值为2的幂。n位二进制正整数可以按权展开写成

$$[a_{n-1}a_{n-2}\cdots a_2a_1a_0] = a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + \cdots a_2 \times 2^2 + a_1 \times 2^1 + a_0 \times 2^0 = \sum_{i=0}^{n-1} a_i \times 2^i$$

其中a_0，a_1，a_2，…，a_{n-2}，a_{n-1}为二进制数的各位数字，取值为0或1。2^0，2^1，2^2，…，2^{n-2}，2^{n-1}为各数位的权。

例如，二进制数1101按权的展开式为

$$1101 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13$$

即二进制数1101在数值上等于十进制数13。

3. 十六进制（Hexadecimal）

十六进制中，包括0，1，2，3，4，5，6，7，8，9，A，B，C，D，E，F 16个数码，采用“逢十六进一”的运算法则，各位的权值为16的幂。n位十六进制正整数的按权展开式为

$$[N]_{16} = \sum_{i=0}^{n-1} a_i \times 16^i$$

其中a_i为十六进制数N的第i位数字，取值为0～F。

例如，数0A3E按权的展开式为

$$0A3E = 10 \times 16^2 + 3 \times 16^1 + 14 \times 16^0 = 2560 + 48 + 14 = 2622$$

即十六进制数0A3E在数值上等于十进制数2622。

为了方便，在数字后面跟一个英文字母表示其数制。其中“D”（Decimal）表示该数为十进制，也可省略，如97D和97都表示十进制数；“B”（Binary）表示该数为二进制，如1011B表示该数为二进制；“H”（Hexadecimal）表示该数为十六进制，如十六进制数“7A”记为“7AH”，同时，以字母开头的十六进制数，在编写程序时必须带有前缀0，以示区别于一般字符串，如十六进制数“FF”记为“0FFH”。

十进制数0～15的二进制、十六进制对应关系，见表1.1.1。

表1.1.1　十进制、二进制、十六进制数码对照表

十进制（D）	二进制（B）	十六进制（H）	十进制（D）	二进制（B）	十六进制（H）
0	0000	0	5	0101	5
1	0001	1	6	0110	6
2	0010	2	7	0111	7
3	0011	3	8	1000	8
4	0100	4	9	1001	9

续表

十进制（D）	二进制（B）	十六进制（H）	十进制（D）	二进制（B）	十六进制（H）
10	1010	A	13	1101	D
11	1011	B	14	1110	E
12	1100	C	15	1111	F

1.1.2.2 常用数制的转换

1. 二、十六进制数转换成十进制数

根据二进制数的一般表达式，将其按权展开再相加，即可得到对应的十进制数。

例 1.1.1 将 11000101B 转换成十进制数。

$$11000101B = 1\times2^7+1\times2^6+0\times2^5+0\times2^4+0\times2^3+1\times2^2+0\times2^1+1\times2^0$$
$$= 128+64+4+1 = 197D$$

十六进制数转换成十进制数时，将 A ~ F 还原成 10 ~ 15，同样按权展开相加，即得对应的结果。

例 1.1.2 将 1FBH 转换成十进制数。

$$1FBH = 1\times16^2+15\times16^1+11\times16^0$$
$$= 256+240+11 = 507D$$

对于含有小数部分的数值，同样采用按权展开相加的方法计算。m 进制数小数点后第 n 位的权值为 m^{-n}。

例 1.1.3 将 1110.101B 转换成十进制数。

$$1110.101B = 1\times2^3+1\times2^2+1\times2^1+0\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}$$
$$= 8+4+2+0.5+0.125 = 14.625D$$

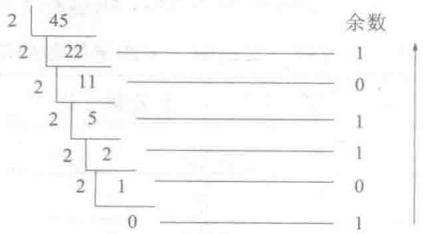
例 1.1.4 将 0FF.8AH 转换成十进制数。

$$0FF.8AH = 15\times16^1+15\times16^0+8\times16^{-1}+10\times16^{-2}$$
$$\approx 240+15+0.5+0.039 = 255.539D$$

2. 十进制数转换成二、十六进制数

十进制整数转换成二进制整数，采用“除二取余”法，即用 2 去除十进制数，取出余数，再用商去除以 2，重复这个过程，直至商为 0。最后，将所得余数按照从后向前的顺序排列即为转换后的二进制数。

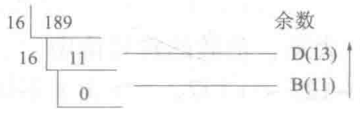
例 1.1.5 将十进制数 45 转换成二进制数。



即 45D = 101101B。

同理，将十进制数“除以 16 取余”即可得到十六进制数。

例 1.1.6 将 189 转换成十六进制数。

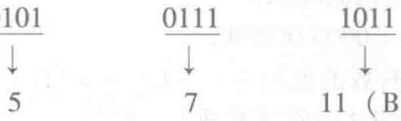


结果 189 = 0BDH。

3. 二进制与十六进制的转换

由于4位二进制数正好可表示0000～1111共16个数字，即十六进制的基本数字0～F。因此，二进制正整数转换成十六进制数时，从最低位开始，4位二进制数为一组，若高位不足4位时加0补足4位，然后每4位转换成相应的十六进制数字，按原来的顺序排列即得十六进制数。

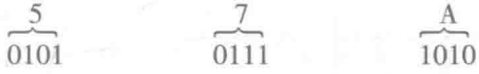
例 1.1.7　将二进制数 10101111011B 转换成十六进制数。



即 10101111011B = 57BH。

十六进制正整数转换成二进制数时，将每位十六进制数转换成4位二进制数。

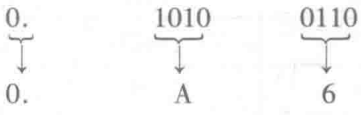
例 1.1.8　将57AH转换成二进制数。



即 57AH = 10101111010B。

将二进制纯小数转换成十六进制数时，从高位开始分组，4位为一组，若低位不足4位，则在低位补零，然后每4位转换成相应的十六进制数字，最后按原顺序写成十六进制数（小数点位置不变）。

例 1.1.9　将0.1010011B转换成十六进制数。



即 0.1010011B = 0.A6H。

1.1.2.3　计算机中的原码、反码、补码

1. 无符号数与有符号数

在字长为8位的微型计算机中，一个字节数据用8位二进制数表示。如果处理的是无符号数，8位二进制数的8位数符都表示数值，从0000 0000B到1111 1111B，表示的数值从0到255。所以，8位二进制数表示的无符号数范围是0～255，共256个数。

如果计算机处理的是有符号数，符号+/－要用1位二进制数表示，这时8位数符的最高位D7表示符号，其他7位表示数值，如图1.1.2所示。D7 = 1表示负数，D7 = 0表示正数。有符号数在计算机中可以用原码、反码、补码三种方法表示，8位二进制数码的不同表达含义见表1.1.2。因为补码表示方法使用较普遍，所以重点讨论补码的使用。

图 1.1.2　数值位和符号位

2. 原码和反码

原码：正数的符号位用“0”表示，负数的符号位用“1”表示，而数值位保持不变。8位二进制原码表示的数的范围是 -127 ~ +127，±0 表示不同。

反码：正数的反码与原码相同。负数的反码，其符号位也用“1”表示，数值位按位求反而得到。8 位二进制反码表示的数的范围也是 -127 ~ +127，±0 表示不同。

3. 补码

（1）正数补码的符号位 D7 =0，负数补码的符号位 D7 =1。

（2）正数的补码表示与原码相同。

（3）负数的补码是符号位不变，数值位按位取反即得反码，然后反码加 1 即得补码；对负数补码求反加 1，回复为该数的原码。

（4）$[+0]_{补} = [-0]_{补} = 0000\ 0000B$。

（5）8 位二进制补码表示的数的范围为：-128 ~ +127。

（6）采用补码后，可以将减法运算转换成加法运算。

表 1.1.2　8 位二进制数码的不同表达含义

8 位二进制数	无符号数	原码	反码	补码
00000000	0	+0	+0	+0
00000001	1	+1	+1	+1
00000010	2	+2	+2	+2
⋮	⋮	⋮	⋮	⋮
01111100	124	+124	+124	+124
01111101	125	+125	+125	+125
01111110	126	+126	+126	+126
01111111	127	+127	+127	+127
10000000	128	-0	-127	-128
10000001	129	-1	-126	-127
10000010	130	-2	-125	-126
⋮	⋮	⋮	⋮	⋮
11111100	252	-124	-3	-4
11111101	253	-125	-2	-3
11111110	254	-126	-1	-2
11111111	255	-127	-0	-1

例 1.1.10　求 $X=5$ 和 $X=-5$ 的补码。

解：（1）$[5]_{补} = 0000\ 0101B$

（2）$X = -5 = 1000\ 0101B < 0$

$[-5]_{反} = 1111\ 1010$

$[-5]_{补} = [-5]_{反} + 1 = 1111\ 1011B$

例 1.1.11　已知 $[X]_{补} = 1111\ 1101B$，求 X 的值。

解：因为补码 1111 1101B 的 D7 = 1，所以是负数。

$[X]_{反} = 1000\ 0010$

$[X]_{补} = 1000\ 0011B = -3$

例 1.1.12　设 X = 97 - 65 = 32，试用 8 位补码运算，并比较其结果。

解：$[97]_{补} = 0110\ 0001B$

$[-65]_{补} = 1011\ 1111B$

则

$$\begin{array}{r} 0110\ 0001 \\ +\ 1011\ 1111 \\ \hline 1\ 0010\ 0000 \end{array}$$

只保留 8 位，所以 $[X]_{补} = 0010\ 0000B = 20H$。

由于 D7 = 0，所以 X 是正数，因此 $X = [X]_{补} = 20H = 32$。

比较两种计算方法，可见结果相同。应该注意：若参与运算的两数或运算结果超出 8 位二进制数的表示范围，则运算结果不正确。

带符号数采用补码表示后，微型计算机的运算只需设置加法器，这样能够简化硬件结构。

1.1.2.4　计算机中的二进制编码

1. 二－十进制码（BCD 码）

人们通常比较习惯读写十进制数，而计算机只能处理二进制数。为此，创立了二进制编码的十进制数（Binary Coded Decimal，BCD），简称二－十进制数，又称 BCD 码。BCD 码有许多种，最常用的是 8421BCD 码。8421BCD 码用 0000H ~ 1001H 代表十进制数 0 ~ 9。4 位二进制数每位的权分别是 8，4，2，1，故得此名。

十进制数与 BCD 码之间的转换十分方便，只要把数符 0 ~ 9 与对应的 0000 ~ 1001 互换就行了。它们之间的对应关系见表 1.1.3。

表 1.1.3　8421BCD 码与十进制数、二进制数的对应关系

十进制数	8421BCD 码	二进制数	十进制数	8241BCD 码	二进制数
0	0000	0000	8	1000	1000
1	0001	0001	9	1001	1001
2	0010	0010	10	0001 0000	1010
3	0011	0011	11	0001 0001	1011
4	0100	0100	12	0001 0010	1100
5	0101	0101	13	0001 0011	1101
6	0110	0110	14	0001 0100	1110
7	0111	0111	15	0001 0101	1111

二进制数不方便直接转换成 BCD 码，先经过十进制数比较好。

例如：0010 1001B = 29H = 41D = [0100 0001] BCD

2. 十进制调整

十进制数只有 0 ~ 9 十个数符，而 4 位二进制数可以表示 16 种状态，所以 1010 ~ 1111

六种状态是多余的非法码。计算机在输入二－十进制数时，依然按二进制运算。但是，4 位二进制数逢 16 进一，而 1 位二－十进制数逢 10 进一，这将有可能产生非法码。因此，51 单片机安排了专门的指令对运算结果进行十进制调整。

加法运算的调整规则是：两个二－十进制数相加后，如果和的高 4 位或低 4 位中出现非法码 1010～1111，则高 4 位或低 4 位加 0110B；如果和的高 4 位（D7）或低 4 位（D3）出现向高位的进位，则高 4 位或低 4 位加 0110B 调整。

例 1.1.13 试用 BCD 码计算 97 加 85，并进行必要的调整。

解： $97=[1001\ 0111]_{BCD}$，$85=[1000\ 0101]_{BCD}$

则

$$\begin{array}{r} [1001\ 0111]_{BCD} \\ +)\quad [1000\ 0101]_{BCD} \\ \hline 1\ 0001\ 1100 \end{array}$$

运算中，低 4 位出现了非法码，而高 4 位的 D7 向高位有进位，需要调整。

$$\begin{array}{r} 1\ 0001\ 1100 \\ +)\quad 0110\ 0110 \\ \hline [1\ 1000\ 0010]_{BCD}=182 \end{array}$$

减法运算的调整方法是：差的高 4 位（D7），或低 4 位（D3）向高位有借位，则高 4 位或低 4 位要减 0110B 调整。

3. ASCII 码

计算机除了要处理数字量之外，还要处理文字、符号等信息，这些信息也必须采用二进制数码表示。计算机处理文字和符号信息时经常使用 ASCII 码，它的全称是美国标准信息交换码（American Standard Code for Information Interchange ASCII），见附录。ASCII 码由 7 位二进制数码构成，共有 128 个字符。ASCII 码主要用于微机、外设通信，以及人机对话等方面。例如，当按下微机的某一键时，键盘中的单片机便会将所按的键码转换成 ASCII 码输入微机。

在计算机通信领域，信息发送和接收过程中经常需要进行信息检验以避免出错。ASCII 码只用到 7 位二进制数，8 位信息的最高位 D7 可以用作奇偶校验位。

在串行通信中，发送端与接收端必须事先协议校验方式。采用奇校验时，发送信息每个字节中“1”的个数必须是奇数，校验位 D7 的状态与其余 7 位信息中“1”的个数有关，若其余 7 位信息中“1”的个数为偶数，则 D7 位置 1；反之，D_7 位清零。字母“O”“K”的 ASCII 码和 8 位奇校验信息如表 1.1.4 所示。

表 1.1.4 字母“O”“K”的 ASCII 码和 8 位奇校验信息

字母	ASCII 码	8 位奇校验信息
O	100 1111	0100 1111
K	100 1011	1100 1011

如果接收端接收信息时，经校验发现“1”的个数为偶数，说明信息在传送过程中出现错误，需要进行相应的出错处理。

若采用偶校验方式时，“1”的个数一定是偶数。奇偶校验方法简单易行，在计算机通信中得到了广泛的应用。

特别指出的是，计算机只识别 0 和 1，是有符号数还是无符号数、是补码还是原码、是