

普通高等学校机械专业**卓越**工程师教育培养计划系列教材

单片机原理与 C语言程序设计

付先成 高恒强 蔡红娟 / 编著



华中科技大学出版社

<http://www.hustp.com>

普通高等学校机械专业卓越工程师教育培养计划系列教材

单片机原理与 C 语言程序设计

付先成 高恒强 蔡红娟 编 著

华中科技大学出版社

中国·武汉

内 容 简 介

本书通篇以 C 语言为编程语言,以 89C51 单片机为核心,系统介绍单片机原理及其接口技术。内容包括 89C51 单片机的硬件结构、指令系统、C 语言程序设计、定时器/计数器、中断系统、串行通信、并行总线扩展、键盘与显示接口、A/D 与 D/A 接口,以及单片机应用系统设计实例。

本着使读者能够快速掌握单片机基本知识、培养实际应用开发能力的宗旨,本书力求在内容取舍、编排顺序、问题阐述和教学方法上有所改进。为此,第 5~8 章内容都采用了渐进式的结构安排,帮助学生知识的理解;通过引入 C 语言程序设计和实例分析,培养学生的实际应用开发能力;同时也注重新技术的介绍,引入了彩色液晶接口技术,为现代智能显示终端的接口应用提供了很好的参考,从而激发学生对这门课程的学习兴趣。

本书可作为高等院校机械、电子、控制、计算机等专业的单片机课程教材,也可供从事单片机应用与产品开发的工程技术人员参考。

图书在版编目(CIP)数据

单片机原理与 C 语言程序设计/付先成,高恒强,蔡红娟编著. —武汉:华中科技大学出版社,2015.5
普通高等学校机械专业卓越工程师教育培养计划系列教材
ISBN 978-7-5680-0851-8

I. ①单… II. ①付… ②高… ③蔡… III. ①单片微型计算机-高等学校-教材 ②C 语言-程序设计-高等学校-教材 IV. ①TP368.1 ②TP312

中国版本图书馆 CIP 数据核字(2015)第 099658 号

单片机原理与 C 语言程序设计

付先成 高恒强 蔡红娟 编著

策划编辑:俞道凯

责任编辑:刘 飞

封面设计:刘 卉

责任校对:张 琳

责任监印:张正林

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)81321913

录 排:武汉市洪山区佳年华文印部

印 刷:武汉科源印刷设计有限公司

开 本:787mm×1092mm 1/16

印 张:13.5

字 数:344 千字

版 次:2015 年 8 月第 1 版第 1 次印刷

定 价:28.00 元



本书若有印装质量问题,请向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

前 言

单片机又称为微控制器,是一个单芯片形态、面向控制对象的嵌入式应用计算机系统。它的出现及发展,使计算机技术从通用型数值计算领域进入到智能化的控制领域。因此,计算机技术在两个重要领域——通用计算机领域和嵌入式领域都得到了极其重要的发展,并正在深深地改变着我们的生活。

单片机已成为当今各种新技术的载体,单片机应用成为一项通用的工程应用技术,各个应用领域的工程技术人员都需要掌握单片机应用技术。单片机课程也早已经成为国内高等院校众多理工科专业的必修课程。而在众多单片机类型中,89C51 系列单片机拥有最悠久历史,其具有广泛的实验开发环境、丰富的软硬件设计资源和人力资源,以及简单易学等优点,所以长期以来单片机课程都是以 89C51 系列单片机为学习对象的。

本着使读者能够快速掌握单片机基本知识、培养实际应用开发能力的宗旨,本书力求在内容取舍、编排顺序、问题阐述和教学方法上有所改进。因此,在本书编写过程中,着重突出了以下几个特点。

1. 渐进式的结构安排

本书在结构安排上采用渐进方式,由浅入深,由通性到特性。尤其在介绍单片机硬件接口技术时,都是先通过电路框图等阐述一般性工作原理,然后才开始论述 89C51 单片机相关硬件接口、特殊功能寄存器和工作方式等,最后进行接口技术的实际应用举例,层层深入,丝丝入扣,符合学生的认知规律。

2. 突出实践工程能力的培养

本书通篇以 C 语言为编程语言,从应用的角度介绍了 C 语言程序设计与各种硬件接口设计、各种常用的数据运算和处理程序、接口驱动程序,以及 89C51 单片机应用系统的设计。书中的应用实例,大多来自作者科研工作及教学实践,且经过检验,内容丰富、翔实。这不仅大大激发了学生学习单片机的兴趣,也培养了学生树立工程意识和实现工程技术的能力。本书 1~10 章每章都留有适量的、应用性的习题,以培养学生工程实际分析问题和解决问题的能力。

3. 基础知识与新技术相结合

本书在内容上进行了精心挑选,既着眼于实用性,又注重先进性。本书既重视单片机基础原理知识的提炼与更新,又介绍了单片机的新技术应用,特别是引入了彩色液晶显示接口技术的应用,着重介绍了 QPY 并口彩色液晶模块与 89C51 的接口技术和 SDW 串口智能显示终端与 89C51 的接口技术,为现代智能显示终端的接口应用提供了很好的知识储备。

本书共分 11 章,第 1 章绪论主要是针对那些没有系统学习过计算机基础知识的读者而准备的,这部分内容对后面的学习十分有用。第 2、3、4、5、6、7 章主要讲述 89C51 单片机的硬件和软件知识。第 8、9、10 章主要介绍单片机的扩展知识。第 11 章则主要介绍单片机的应用。

本书可作为高等院校机械、电子、控制、计算机等专业的单片机课程教材,也可供从事单片机应用与产品开发的工程技术人员参考。

本书第 1、2 章由蔡红娟编写,第 3、5、6、10、11 章由高恒强编写,第 4、7、8、9 章由付先成编写。此外,在本书编写过程中,周磊、周志威、钟剑文等也做了大量工作,在此对他们表示深深的谢意。在本书编写过程中,我们还学习和参考了许多单片机方面的教材和资料,受益匪浅,在此向其作者表示感谢。

由于作者水平有限,书中难免会有错误和不足之处,恳请广大读者批评指正。

作者

2015 年 6 月于武汉

目 录

第 1 章 绪论	(1)
1.1 计算机中的数制	(1)
1.2 二进制数的算术运算和逻辑运算	(5)
1.3 二进制编码	(7)
1.4 单片机概述	(9)
习题	(10)
第 2 章 89C51 单片机的硬件结构	(11)
2.1 基本组成	(11)
2.2 I/O 端口	(14)
2.3 时钟电路	(17)
2.4 复位电路	(19)
2.5 最小应用系统	(21)
2.6 存储器空间配置	(21)
习题	(29)
第 3 章 89C51 单片机指令系统	(31)
3.1 指令系统概述	(31)
3.2 指令寻址方式	(32)
3.3 指令分类介绍	(35)
3.4 指令执行过程	(53)
3.5 汇编语言程序设计举例	(54)
习题	(59)
第 4 章 89C51 单片机的 C 语言程序设计	(61)
4.1 单片机程序设计语言概述	(61)
4.2 C51 编程基础	(62)
4.3 C51 编程举例	(70)
4.4 C51 集成开发环境	(73)
习题	(82)
第 5 章 定时器/计数器	(84)
5.1 工作原理	(84)
5.2 特殊功能寄存器	(85)
5.3 四种工作方式	(87)
习题	(92)
第 6 章 中断系统	(94)
6.1 中断概念	(94)
6.2 工作原理	(95)

6.3 特殊功能寄存器.....	(97)
6.4 中断响应及中断服务函数.....	(99)
习题.....	(106)
第 7 章 串行通信	(107)
7.1 串行通信概念	(107)
7.2 工作原理	(110)
7.3 四种工作方式	(113)
7.4 串行通信应用	(118)
7.5 RS-232C 电平标准	(128)
习题.....	(130)
第 8 章 并行总线扩展	(131)
8.1 并行总线结构	(131)
8.2 存储器的扩展	(138)
8.3 并行 I/O 口扩展	(143)
习题.....	(152)
第 9 章 键盘与显示接口	(153)
9.1 键盘接口	(153)
9.2 LED 数码管显示接口	(156)
9.3 彩色 LCD 显示接口.....	(163)
习题.....	(181)
第 10 章 A/D 及 D/A 转换接口	(182)
10.1 A/D 转换器接口	(182)
10.2 D/A 转换器接口	(189)
习题.....	(196)
第 11 章 单片机应用	(197)
11.1 步进电动机控制电路.....	(197)
11.2 频率的测量.....	(202)
11.3 时钟计时.....	(206)
参考文献	(209)

第 1 章 绪 论

1.1 计算机中的数制

日常生活中有许多不同的进位计数制(简称为数制),最常用的数制是十进制,而在计算机中只能采用二进制数。对于计算机,无论是指令、数据或地址的表示,还是它们的存储、加工和传送,都是以二进制数的形式进行的。

1.1.1 不同进制数

1. 十进制数

十进制是最常使用的计数进位制。这种计数进位制的每 1 位数都用 0~9 十个数码中的一个数码来表示,所以计数基数是 10。超过 9 的数则需用多位数表示,其中低位数和相邻高位之间的关系是逢十进一,故称为十进制数,例如:

$$1234=1\times10^3+2\times10^2+3\times10^1+4\times10^0$$

其中,10 称为基数, 10^3 、 10^2 、 10^1 、 10^0 称为各位数的“权”。

2. 二进制数

在二进制数中,每一位仅有 0 和 1 两个可能的数码,所以计数基数是 2。低位和相邻的高位之间的进位关系是逢二进一,故称为二进制数。

例如,一个二进制数 1110 可展开为:

$$1110=1\times2^3+1\times2^2+1\times2^1+0\times2^0$$

3. 十六进制数

十六进制数的每 1 位数都有十六种可能的数码,分别用 0~9、A(10)、B(11)、C(12)、D(13)、E(14)及 F(15)表示。低位数和高一位数之间的关系是逢十六进一。

二进制数书写起来太长,且不利于阅读和记忆;计算机常用十六进制数来缩写二进制数。4 位二进制数有 16 个不同的状态 0000~1111,可分别用十六进制数的 0~9 和 A~F 这 16 个数码来表示这 16 个二进制数 0000~1111 转换成十进制数就是 0~15。计算机采用的是二进制数,而人们最熟悉、最常用的是十进制数,为此要熟练地掌握十进制数、二进制数和十六进制数间的相互转换方法。它们之间的关系如表 1-1 所示。

表 1-1 十进制数、二进制数及十六进制数对照表

十进制数	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
二进制数	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
十六进制数	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

为了区别十进制数、二进制数及十六进制数 3 种数制,可在数的右下角注明其数制,或者在数的后面加一字母来表示,如用 B(binary)表示其是二进制数;用 D(decimal)或不带字母表示其是十进制数;用 H(hexadecimal)表示其是十六进制数。

1.1.2 不同进制数之间的转换

1. 二进制数与十进制数间的相互转换

二进制数转换为十进制数十分简单,只需将二进制数按权展开相加即可,如:

$$\begin{aligned} 1011\text{B} &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 \\ &= 11 \end{aligned}$$

十进制整数转换为二进制数可用除 2 取余法,即用 2 不断地去除待转换的十进制数,直至商等于 0 为止。将所得的各次余数依倒序排列,即可得到所转换的二进制数。

例如,将十进制数 39 转换成二进制数,转换过程如下:

$$\begin{array}{r|l} 2 & 39 \quad \dots\dots 1 \\ \hline 2 & 19 \quad \dots\dots 1 \\ \hline 2 & 9 \quad \dots\dots 1 \\ \hline 2 & 4 \quad \dots\dots 0 \\ \hline 2 & 2 \quad \dots\dots 0 \\ \hline 2 & 1 \quad \dots\dots 1 \\ \hline & 0 \end{array}$$

即 $39 = 10\ 0111\text{B}$

2. 二进制数和十六进制数间的相互转换

二进制整数转换为十六进制数,其方法是,从右(最低位)向左将二进制数分组:每 4 位为 1 组,最后一组若不足 4 位则在其左边添加 0,以凑成 4 位 1 组,每组用 1 位十六进制数表示。如:

$$1111111000111\text{B} \rightarrow 1\ 1111\ 1100\ 0111\text{B} \rightarrow 0001\ 1111\ 1100\ 0111\text{B} = 1\text{FC7H}$$

十六进制数转换为二进制数,只需用 4 位二进制数代替 1 位十六进制数即可。如:

$$3\text{AB9H} = 0011\ 1010\ 1011\ 1001\text{B}$$

3. 十六进制数和十进制数间的相互转换

十六进制数转换为十进制数采用按权展开相加即可。如:

$$\begin{aligned} 1\text{F3DH} &= 16^3 \times 1 + 16^2 \times 15 + 16^1 \times 3 + 16^0 \times 13 \\ &= 4096 \times 1 + 256 \times 15 + 16 \times 3 + 1 \times 13 \\ &= 7997 \end{aligned}$$

十进制数转换为十六进制数可采用“除基取余”法。如将十进制数 39 转换为十六进制数,算式如下:

$$\begin{array}{r|l} 16 & 39 \quad \dots\dots 7 \\ \hline 16 & 2 \quad \dots\dots 2 \\ \hline & 0 \end{array}$$

即 $39 = 27\text{H}$

1.1.3 机器数及其表示形式

1. 机器数与真值

计算机使用的二进制数称为机器数。机器数可分为符号数和无符号数等两种。

对于符号数的表示,二进制数与十进制数一样有正负之分,在书写时需在数的前面加一个正负号,如+105的二进制表示为+01101001,-105的二进制表示为-01101001,这种数值连同符号“+”或“-”作为一个数的表示形式称为真值。然而计算机无法在二进制数前直接“写”上正号或者负号,因此通常用二进制数值来表示符号,即正号用“0”表示,负号用“1”表示,这种数值连同符号数码“0”或“1”一起作为一个数的表示形式就称为机器数,如+105的机器数表示为01101001,-105的机器数表示为11101001。

把机器数的符号位也当做数值的数,就是无符号数。无符号数没有正负的概念,只是一串二进制代码。

机器数有原码、反码和补码三种表示方法。

2. 原码

原码是机器数的原始表示。数值用其绝对值,正数的符号位用0表示,负数的符号位用1表示,这样表示的数就称为原码。如:

$$X1=105=01101001B \quad [X1]_{\text{原}}=01101001B$$

$$X2=-105=-01101001B \quad [X2]_{\text{原}}=11101001B$$

其中,最高位为符号,后面7位是数值。用原码表示时,+105和-105的数值部分相同而符号位相反。

原码表示简单易懂,而且与真值的转换方便。但若是两个异号数相加,或两个同号数相减,就要做减法。为了把减运算转换为加运算,从而简化计算机的结构,就引进了反码和补码的概念。

3. 反码

正数的反码与原码一样,符号位为0,其余位为其数值;负数的反码可按原码转换得到,即符号位不变,数值位按位取反。如:

$$X1=105=01101001B \quad [X1]_{\text{反}}=01101001B$$

$$X2=-105=-01101001B \quad [X2]_{\text{反}}=10010110B$$

4. 补码

正数的补码与原码一样,符号位为0,其余位为其数值;负数的补码为其反码加1。如:

$$X1=105=01101001B \quad [X1]_{\text{补}}=01101001B$$

$$X2=-105=-01101001B \quad [X2]_{\text{补}}=10010111B$$

5. 二进制数的范围

8位二进制数,将其看做无符号数和符号数,它所表示的数的大小是不同的。为加深读者的印象,将其列于表1-2中。

由表1-2可知,8位无符号二进制数的数值范围为0~255。8位二进制数的反码的数值范围为-127~127。8位二进制数原码的数值范围为-127~127。8位二进制数补码的数值范

围为-128~127。

表 1-2 8 位二进制数(2 位十六进制数)的大小

8 位二进制数	2 位十六进制数	无符号数	原码数	反码数	补码数
0000 0000	00	0	0	0	0
0000 0001	01	1	1	1	1
0000 0010	02	2	2	2	2
⋮	⋮	⋮	⋮	⋮	⋮
0111 1101	7D	125	125	125	125
0111 1110	7E	126	126	126	126
0111 1111	7F	127	127	127	127
1000 0000	80	128	-0	-127	-128
1000 0001	81	129	-1	-126	-127
1000 0010	82	130	-2	-125	-126
⋮	⋮	⋮	⋮	⋮	⋮
1111 1101	FD	253	-125	-2	-3
1111 1110	FE	254	-126	-1	-2
1111 1111	FF	255	-127	-0	-1

对于一个 16 位的二进制数,若把它看成无符号数,则其数值范围为 0~65535;若把它看成补码数,则其数值范围为-32768~32767。

6. 二进制数的扩展

二进制数的扩展是指将一个位数较少的数据扩展到位数较多的数的过程,如从 8 位(字节)扩展到 16 位(字),或从 16 位扩展到 32 位(双字)。一个二进制数扩展后,其数的符号和大小应保持不变。

无符号数的扩展是将其左边添加 0。如 8 位无符号二进制数 F8H 扩展为 16 位无符号二进制数,则为 00F8H。

补码表示的二进制数的符号位向左扩展若干位后,所得到的补码数的真值应保持不变。所以,对于用补码表示的二进制数,正数的扩展应该在其前面补 0,而负数的扩展,则应该在前面补 1。例如:68 用 8 位二进制数表示的补码为 44H,用 16 位二进制数表示的补码为 0044H;-68 用 8 位二进制数表示的补码为 BCH,用 16 位二进制数的补码表示为 FFBCH。

1.2 二进制数的算术运算和逻辑运算

因为计算机只能采用二进制数,所以其算术运算和逻辑运算也只能针对二进制数进行,计算机一切复杂的运算和操作都是建立在这些最基本的算术运算和逻辑运算之上的。

1.2.1 二进制算术运算

计算机把机器数均当做无符号数进行运算,即符号位也参与运算。运算的结果要根据运算结果的符号,运算有无进(借)位和溢出等来判别。计算机中设置有这些标志位,标志位的值由运算结果自动设定。

1. 无符号数的运算

无符号数实际上是指参加运算的数均为正数,且整个数位全部用于表示数值。 n 位无符号二进制数的范围为 $0 \sim (2^n - 1)$ 。

(1) 两个无符号数相加,由于两个加数均为正数,因此其和也是正数。当和超过其位数所允许的范围时,就向更高位进位。如:

$$\begin{array}{r}
 127 + 160 = 7FH + A0H \\
 0111\ 1111 \\
 + 1010\ 0000 \\
 \hline
 1\ 0001\ 1111 = 11FH = 256 + 16 + 15 = 287 \\
 \uparrow \text{进位}
 \end{array}$$

(2) 两个无符号数相减,被减数大于或等于减数,无借位,结果为正;被减数小于减数,有借位,结果为负。如:

$$\begin{array}{r}
 192 - 10 = C0H - 0AH \\
 1100\ 0000 \\
 - 0000\ 1010 \\
 \hline
 1011\ 0110 = B6H = 176 + 6 = 182
 \end{array}$$

反过来相减,即 $10 - 192$,运算过程如下:

$$\begin{array}{r}
 0000\ 1010 \\
 - 1100\ 0000 \\
 \hline
 1\ 0100\ 1010 = -10110110B = -B6H = -182 \\
 \uparrow \text{借位}
 \end{array}$$

由此可见,对无符号数进行减法运算,其结果的符号用进位来判别: $CF=0$ (无借位),结果为正; $CF=1$ (有借位)结果为负(对 8 位数值位求补得到它的绝对值)。

2. 补码数的运算

n 位二进制补码数,除去一位符号位外,还有 $n-1$ 位表示数值,所能表示的补码的范围为 $-2^{n-1} \sim (2^{n-1} - 1)$ 。如果运算结果超过此范围就会产生溢出。如:

$$105 + 50 = 69H + 32H$$

$$0110\ 1001$$

$$+ 0011\ 0010$$

$$1001\ 1011 = 9BH = 155 \text{ 或 } -65H = -101$$

若把结果视为无符号数,为 155,结果是正确的。若将此结果视为符号数,其符号位为 1,结果为 -101,这显然是错误的。其原因是和数 155 大于 8 位符号数所能表示的补码数的最大值 127,使数值部分占据了符号位的位置,产生了溢出,从而导致结果错误。

1.2.2 二进制逻辑运算

计算机除了可进行基本的算术运算外,还可对两个或一个无符号二进制数进行逻辑运算。计算机中的逻辑运算,主要有“非”、“与”、“或”和“异或”等 4 种基本运算。下面介绍这 4 种基本逻辑运算。

1. 逻辑“非”

逻辑“非”也称“求反”,常用变量上方加一横来表示。假设有逻辑变量 A,则其逻辑“非”表达式可表示为 $\bar{A}$ 。

对二进制数进行逻辑“非”运算,就是按位求反。例如, $A = 0110\ 1001B$,则执行逻辑“非”运算后为 $\bar{A} = 10010110B$ 。

2. 逻辑“与”

逻辑“与”运算也称为逻辑乘,常用记号“ $\wedge$ ”或“ $\cdot$ ”来表示。1 位二进制数逻辑“与”的规则为

$$0 \wedge 0 = 0, \quad 0 \wedge 1 = 0, \quad 1 \wedge 0 = 0, \quad 1 \wedge 1 = 1$$

01100001B 和 11001011B 的逻辑“与”算式如下:

$$\begin{array}{r} 0110\ 0001 \\ \wedge 1100\ 1011 \\ \hline 0100\ 0001 \end{array}$$

即: $01100001B \wedge 11001011B = 0100\ 0001B$

3. 逻辑“或”

逻辑“或”运算也称为逻辑加,常用记号“ $\vee$ ”或“+”来表示。1 位二进制数逻辑“或”的规则为

$$0 \vee 0 = 0, \quad 0 \vee 1 = 1, \quad 1 \vee 0 = 1, \quad 1 \vee 1 = 1$$

对两个二进制数进行逻辑“或”运算,就是按位求它们的“或”,例如, 01100001B 和 11001011B 的逻辑“或”算式如下:

$$\begin{array}{r} 0110\ 0001 \\ \vee 1100\ 1011 \\ \hline 1110\ 1011 \end{array}$$

即: $01100001B \vee 11001011B = 11101011B$

4. 逻辑“异或”

逻辑“异或”,常用符号“ $\oplus$ ”来表示。1 位二进制数的逻辑“异或”运算规则为

$$0 \oplus 0 = 0, \quad 0 \oplus 1 = 1, \quad 1 \oplus 0 = 1, \quad 1 \oplus 1 = 0$$

对两个二进制数进行逻辑“异或”，就是二进制数按位加但不产生进位。例如 01100001B 和 11001011B 的逻辑异或算式如下：

$$\begin{array}{r} 0110\ 0001 \\ \oplus 1100\ 1011 \\ \hline 1010\ 1010 \end{array}$$

即： $01100001B \oplus 11001011B = 10101010B$

1.3 二进制编码

计算机中，除了数是用二进制表示的以外，字母、各种符号，以及指挥计算机执行操作的指令，也都是用二进制数的组合来表示的，这种组合的二进制数称为二进制编码。

1.3.1 BCD 码

计算机中常用二进制数给十进制数编码。十进制数有 0~9 十个数码。要表示这 10 个数码，需要用 4 位二进制数，这称为二进制编码的十进制数，简称为 BCD 码(binary coded decimal)。用 4 位二进制数编码表示 1 位十进制数的方法很多，较常用的是 8421 BCD 码，即用二进制数 0000~1001 分别对应十进制数 0~9。例如，十进制数 3 的 BCD 码为 0011，数 7 的 BCD 码为 0111，若要表示十进制数 37，则可用 00110111 表示，正好为 1 字节(1B)。BCD 码的特点是：4 位之内为二进制关系，每 4 位之间为十进制关系。

定义 BCD 码是为了便于在计算机中使用我们最熟悉的十进制数，特别是在输入与输出操作中，例如，从键盘输入的十进制数到计算机中就变为 BCD 码形式。当然这需要有相应的转换程序才能实现。有了十进制数的输入和输出，在计算机中就会存在十进制数的存储和计算。但计算机中，BCD 码的运算也把数当做二进制数进行运算，所以计算机中的十进制数计算就存在调整问题，以解决 BCD 码运算时因进位和借位产生的偏差。

1.3.2 ASCII 码

计算机除了能对二进制数运算外，还需要对各种各样的字符进行识别和处理，这就要求计算机首先能够表达这些字符，这些常用字符如下。

数字字符：0, 1, …, 9。

大小写的 26 个英文字符：A, B, …, Z；a, b, …, z。

各种标点符号等。

还有非打印字符：CR(回车)，LF(换行)等。

我们要把这些数字、字母、符号等信息输入到计算机中，首先必须对这些信息进行编码。

目前国际上比较通用的是美国标准学会(ANSI)制定的 ASCII 码，ASCII 是“American standard code for information interchange”的简称，中文全称是“美国信息交换标准代码”。ASCII 码原是美国的字符代码标准，由于使用广泛，早已被国际标准化组织确定为国际标准，成为计算机领域中最重要代码。常用字符的 ASCII 码如表 1-3 所示。

表 1-3 常用字符与 ASCII 代码对照表

ASCII 值	字符	控制字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符	ASCII 值	字符
000	(nul)	NUL	032	(space)	064	@	096	`	128	Ç	160	á	192	À
001	©	SOH	033	!	065	A	097	a	129	ü	161	â	193	Á
002	•	STX	034	"	066	B	098	b	130	é	162	ó	194	Â
003	♥	ETX	035	#	067	C	099	c	131	â	163	û	195	Ã
004	♦	EOT	036	\$	068	D	100	d	132	ä	164	ñ	196	—
005	♣	END	037	%	069	E	101	e	133	ä	165	Ñ	197	†
006	♠	ACK	038	&	070	F	102	f	134	ä	166	ª	198	‡
007	(beep)	BEL	039	'	071	G	103	g	135	ç	167	º	199	£
008	■	BS	040	(	072	H	104	h	136	ê	168	¿	200	€
009	(tab)	HT	041	)	073	I	105	i	137	ë	169	ˆ	201	¥
010	(line feed)	LF	042	*	074	J	106	j	138	è	170	˜	202	ℳ
011	(home)	VT	043	+	075	K	107	k	139	ï	171	½	203	¤
012	(form feed)	FF	044	,	076	L	108	l	140	î	172	¼	204	¥
013	(carriage return)	CR	045	-	077	M	109	m	141	í	173	¡	205	£
014	♪	SO	046	.	078	N	110	n	142	Ä	174	«	206	¢
015	☼	SI	047	/	079	O	111	o	143	Å	175	»	207	£
016	▲	DLE	048	0	080	P	112	p	144	Ê	176	☒	208	ℳ
017	▼	DC1	049	1	081	Q	113	q	145	æ	177	☒	209	¥
018	†	DC2	050	2	082	R	114	r	146	Æ	178	☒	210	¥
019	‡	DC3	051	3	083	S	115	s	147	ø	179	☒	211	¥
020	¶	DC4	052	4	084	T	116	t	148	ö	180	☒	212	¥
021	§	NAK	053	5	085	U	117	u	149	ò	181	☒	213	¥
022	—	SYN	054	6	086	V	118	v	150	û	182	☒	214	¥
023	↑	ETB	055	7	087	W	119	w	151	ü	183	☒	215	¥
024	↑	CAN	056	8	088	X	120	x	152	ÿ	184	☒	216	¥
025	↓	EM	057	9	089	Y	121	y	153	ø	185	☒	217	¥
026	→	SUB	058	:	090	Z	122	z	154	Ü	186	☒	218	¥
027	←	ESC	059	;	091	[	123	{	155	—	187	☒	219	¥
028	↳	FS	060	<	092	\	124		156	£	188	☒	220	¥
029	◆	GS	061	=	093	]	125	}	157	¥	189	☒	221	¥
030	▲	RS	062	>	094	^	126	~	158	Ps	190	☒	222	¥
031	▼	US	063	?	095	—	127	△	159	f	191	☒	223	¥

(blank'FF')

1.4 单片机概述

单片机是一个单芯片形态、面向控制对象的嵌入式应用计算机系统。它的出现及发展使计算机技术从通用型数值计算领域进入到智能化的控制领域。从此,计算机技术在两个重要领域——通用计算机领域和嵌入式领域都得到了极其重要的发展,并正在深深地改变着我们的社会。

1.4.1 单片机——微控制器嵌入式应用的概念

所谓单片机,即把组成微型计算机的各个功能部件,如中央处理器 CPU、随机存储器 RAM、只读存储器 ROM、I/O 接口电路、定时器/计数器以及串行通信接口等集成在一块芯片中,构成的一个完整的微型计算机。因此单片机早期的含义为单片微型计算机(single chip microcomputer),直接译为单片机,并一直沿用至今。

由于单片机面对的是测控对象,突出的是控制功能,所以它从功能和形态上来说都是因控制领域应用的要求而诞生的。随着单片机技术的发展,芯片内已集成了许多面对测控对象的接口电路,如 A/D 转换器、D/A 转换器、高速 I/O 口、PWM、WDT 等。这些对外电路及外设接口已经突破了微型计算机(microcomputer)传统的体系结构,所以更为确切地反映单片机本质的名称应是微控制器(micro control unit,MCU)。

目前按单片机内部数据总线宽度不同,分为 4 位、8 位、16 位及 32 位等单片机。

单片机具有体积小、控制功能强、集成度高、可靠性高、性能价格比高等特点,在工业自动化、自动检测与控制、智能仪器仪表、家用电器、通信和汽车电子等各个方面获得了广泛的应用。

1.4.2 单片机的历史与发展

单片机出现的历史并不长,它的产生与发展和微处理器的产生与发展大体上同步。

1970 年微型计算机研制成功后,随即在 1971 年,美国 Intel 公司生产出了 4 位单片机 4004,它的特点是结构简单,功能单一,控制能力弱,价格低廉。

1976 年,Intel 公司推出了 MCS-48 系列单片机,它以体积小、功能全、价格低等特点获得了广泛的应用,成为单片机发展进程中的一个重要阶段,此可谓是第一代单片机。

在 MCS-48 系列单片机的基础上,Intel 公司在 20 世纪 80 年代初推出了第二代单片机的代表 MCS-51 系列单片机。这一代单片机的主要技术特征是,配置了完美的外部并行总线和串行通信接口,规范了特殊功能寄存器的控制模式,以及为增强控制功能而强化布尔处理系统和相关的指令系统,为发展具有良好兼容性的新一代单片机奠定了基础。

进入 20 世纪 90 年代以后,出现了单周期的 80C51 系列的单片机,其运行速度更快,功能更强。同时,PIC、AVR、MSP430 等系列的单片机也相继出现,芯片性能也逐步提高。

在 2000 年前后,英国 ARM 公司相继推出 ARM7 及 ARM9 系列的芯片,ARM7 和 ARM9 芯片的诞生,一方面将微控制器整体性能由 8 位、16 位提升到 32 位,另一方面也将微控制器由单芯片提升到片上系统(system on chip,SOC)。此后,ARM 公司更是相继推出 Cortex M0、Cortex M3、Cortex M4 等面向工业控制领域应用的单片机,这些芯片因为强大的性能、超低的功耗、低廉的价格,成功得到了世界各大半导体厂商的追捧,目前在各个领域得到

了广泛认可和应用。

89C51 系列单片机历史悠久,其广泛的实验开发环境、丰富的软硬件设计资源和人力资源,以及简单易学等优点,使其直到现在仍不失为单片机中的主流机型。在今后相当长的时间里,单片机应用领域中的 8 位机的主流地位还不会改变。同时,89C51 系列单片机性能本身也在不断提高,比如,深圳宏晶科技近年推出多款基于 89C51 系列的单片机,集成度不断提高,性能不断提升。

习 题

- 1-1 将下列十进制数转换为十六进制数:10,64,80,95,100,128,255
- 1-2 将下列十六进制无符号数转换为十进制数:2CH,64H,D5H,100H,4FEH,CADH
- 1-3 将下列各二进制数转换为十进制数。
①11010101B ②11010011B ③10101011B ④10111101B
- 1-4 将下列各数转换为二进制数。
①215D ②253D ③01000011(BCD) ④00101001(BCD)
- 1-5 将表 1-4 所示十六进制数转换为十进制数。

表 1-4 十六进制数与十进制数转换

十六进制数	0FH	7FH	FFH	3FFH	0FFFH	7FFFH	FFFFH
十进制数							
十六进制数	10H	80H	100H	400H	1000H	8000H	10000H
十进制数							

- 1-6 什么叫原码、反码及补码?
- 1-7 已知 $X=01011001B$, $Y=00110101B$,用算术运算规则求:
(1) $X+Y$; (2) $X-Y$
- 1-8 已知 $X=01111010B$, $Y=10101010B$,用逻辑运算规则求:
(1) $X \wedge Y$; (2) $X \vee Y$; (3) $X \oplus Y$; (4) $\bar{X}$
- 1-9 假设“ $9+7=10$ ”,则“ $3+7$ ”=_____。