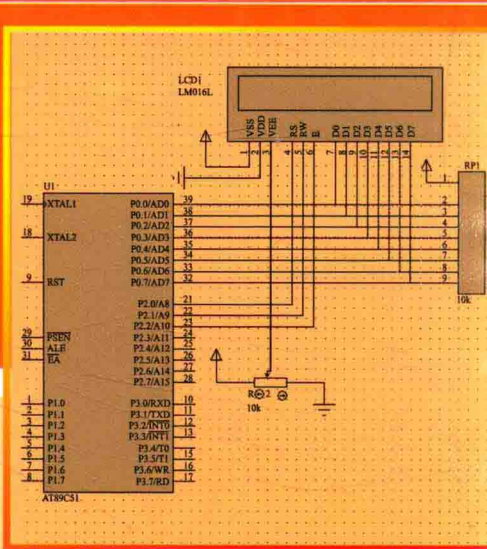


“ 十二五 ” 高等职业教育机电类专业规划教材  
国家示范性高等职业院校建设成果

# 单片机应用与实践教程

DANPIANJI YINGYONG YU SHIJIAN JIAOCHENG

宋国富 主编



“十二五”高等职业教育机电类专业规划教材  
国家示范性高等职业院校建设成果

# 单片机应用与实践教程

宋国富 主 编

张 超 叶林勇 副主编

**中国铁道出版社**  
CHINA RAILWAY PUBLISHING HOUSE

---

## 内 容 简 介

本书主要介绍了单片机的硬件结构及工作原理、基于 C 语言的程序设计、单片机系统扩展技术、中断技术、接口技术、信息转换与传输技术、单片机技术的系统应用等内容。本书以工程实际实训为主线,将 8051 单片机的传统理论贯穿到实训实操过程中。为便于教学组织,特意引入了单片机硬件仿真软件 Proteus 7 Professional 作为实训实操的主要载体,可以使单片机的日常教学摆脱硬件条件的束缚而直接在普通的微机室即可进行。

本书适合作为高等职业院校机电类专业的教材,也可作为部分中职类相关专业教材或参考书,同时还可作为从事电气类专业工作的工程技术人员的自学或参考书。

## 图书在版编目(CIP)数据

单片机应用与实践教程 / 宋国富主编. — 北京 :  
中国铁道出版社, 2015. 1

“十二五”高等职业教育机电类专业规划教材 国家  
示范性高等职业院校建设成果  
ISBN 978-7-113-19797-1

I. ①单… II. ①宋… III. ①单片微型计算机—高等  
职业教育—教材 IV. ①TP368. 1

中国版本图书馆 CIP 数据核字 (2014) 第 309002 号

书 名: 单片机应用与实践教程  
作 者: 宋国富 主编

策 划: 何红艳  
责任编辑: 何红艳  
编辑助理: 绳 超  
封面设计: 付 巍  
封面制作: 白 雪  
责任校对: 汤淑梅  
责任印制: 李 佳

读者热线: 400-668-0820

出版发行: 中国铁道出版社 (100054, 北京市西城区右安门西街 8 号)

网 址: <http://www.51eds.com>

印 刷: 化学工业出版社印刷厂

版 次: 2015 年 1 月第 1 版

2015 年 1 月第 1 次印刷

开 本: 787 mm×1 092 mm 1/16

印张: 13.5 字数: 320 千

印 数: 1~3 000 册

书 号: ISBN 978-7-113-19797-1

定 价: 27.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659

为满足国家对高技能型人才培养的要求,在职业院校教学活动中加大实践教学环节的力度,更好地进行基于项目化教学模式的改革,配合做好国家示范性高等职业院校建设活动,特编写了本书。

单片机技术在工业控制中有着极其广泛的应用,而单片机课程作为一门专业基础课,在电类专业的课程体系中也起着至关重要的作用。在学习本课程之前,先期所要学习的课程主要包括电工基础、模拟电子技术、数字电子技术、自动检测与传感器技术、C 语言程序设计等课程。

本书以单片机系统的工程应用为出发点,将传统单片机系统原理进行了整合,并以实训的形式体现出来,引导学生通过实训实操,主动学习相关原理知识,即按需学习,从而提高学生学习的主观能动性。书中内容组织以突出实践操作技能为主线,实操实操以单片机设计工作现场为背景,教学现场以学生自己操作为主、教师讲授为辅,更好地体现了对学生实践技能的培养。

书中将 8051 单片机的理论体系重新整合,分解成“数制与编码、硬件配置、C 语言程序设计基础、存储器系统、中断技术、定时/计数器、I/O 设备与接口、串行通信、A/D 及 D/A 转换接口和综合实训”等 10 章,且在每章后面(除第 10 章)都配备了技能实训,全书共安排了 18 个基础实训和 4 个综合实训。为使学习更贴近单片机系统实际开发现场,特意安排了 C 语言程序设计的内容,并安排了 Keil C51 开发工具的操作技能。另外,考虑到单片机实验实训环节在具体实施过程中,总会由于诸多硬件方面的问题而使实训案例项目无法实现,从而影响教学效果,本书特意引入了优秀的单片机硬件仿真软件 Proteus 7 Professional 作为实训实操的主要载体,可以使单片机实践教学的实施直接在普通的微机室即可进行。(书中由该软件绘制的电路图的图形符号与国家标准画法不一致,二者对照关系参见附录 B)

本书内容组织原则是如何吸引学生主动学习作为出发点,为此,每个实训的组织均遵循“是什么—为什么—做什么”这样一条主线,层层递进,即先给出所实现任务的所有软件、硬件资源及实施方案,使得学生可以直接利用这些资源得到正确结果,即首先知道结果“是什么”;然后再利用原理解析的形式告诉学生产生这个结果的原因,即“为什么”;等掌握了原理后,再布置一个和原实训相仿的任务,使学生进行创新设计,进一步进行单片机系统应用任务的开发,即“做什么”。

全书由安徽职业技术学院宋国富任主编,并负责统稿。河南职业技术学院张超、三峡电力职业学院叶林勇任副主编。本书的内容主要是编者多年在一线教学中的经验积累。在本书的编写过程中,编者也引用了一些其他类型单片机工程应用方面的案例,并得到了有关院校同行及领导的大力支持,在此深表感谢。特别感谢安徽职业技术学院程周、

洪应、黄有金、谢军等老师的支持与帮助。

由于编者水平有限，书中难免存在疏漏与不足之处，恳请各位专家同行和读者批评指正。

编 者

2014 年 12 月

|                                |    |
|--------------------------------|----|
| 第 1 章 数制与编码.....               | 1  |
| 1.1 不同进位计数制及其转换.....           | 1  |
| 1.1.1 进位计数制.....               | 1  |
| 1.1.2 数制的转换.....               | 2  |
| 1.2 计算机中数的表示方法及运算 .....        | 3  |
| 1.2.1 机器数的特点.....              | 3  |
| 1.2.2 原码、反码和补码.....            | 4  |
| 1.3 BCD 码及 ASCII 码.....        | 5  |
| 1.3.1 8421BCD 编码 .....         | 5  |
| 1.3.2 ASCII 编码 .....           | 5  |
| 1.4 技能实训 .....                 | 7  |
| 【实训 1】Keil C51 的使用方法.....      | 7  |
| 习题 .....                       | 16 |
| 第 2 章 8051 单片机的硬件配置.....       | 18 |
| 2.1 单片机硬件系统的组成.....            | 18 |
| 2.1.1 单片机概述.....               | 18 |
| 2.1.2 微型计算机硬件系统的组成 .....       | 19 |
| 2.1.3 单片微型计算机系统的组成 .....       | 20 |
| 2.2 8051 单片机的引脚功能.....         | 20 |
| 2.2.1 MCS-51 系列单片机简介 .....     | 20 |
| 2.2.2 8051 单片机的内部组成及信号引脚 ..... | 21 |
| 2.3 8051 单片机内部资源分配 .....       | 25 |
| 2.3.1 8051 单片机的存储器空间 .....     | 25 |
| 2.3.2 8051 单片机片内程序存储器 .....    | 29 |
| 2.4 技能实训 .....                 | 29 |
| 【实训 2】并行接口特性.....              | 29 |
| 习题 .....                       | 31 |
| 第 3 章 单片机 C 语言程序设计基础 .....     | 33 |
| 3.1 单片机 C 语言程序设计的一般格式.....     | 33 |
| 3.1.1 单片机 C 语言程序设计的步骤.....     | 33 |
| 3.1.2 单片机 C 语言程序的几个基本概念 .....  | 33 |
| 3.1.3 单片机 C 语言程序的基本结构.....     | 35 |

|       |                                   |    |
|-------|-----------------------------------|----|
| 3.2   | 单片机开发 C 语言程序的数据类型                 | 35 |
| 3.2.1 | 常量和符号常量                           | 36 |
| 3.2.2 | 变量                                | 36 |
| 3.3   | 单片机 C 语言程序的运算符和表达式                | 37 |
| 3.4   | 单片机 C 语言程序的一般语法结构                 | 38 |
| 3.4.1 | 顺序结构                              | 38 |
| 3.4.2 | 分支结构                              | 38 |
| 3.4.3 | 循环结构                              | 40 |
| 3.5   | 8051 单片机的 C 语言程序                  | 41 |
| 3.5.1 | 存储类型及存储区                          | 41 |
| 3.5.2 | 特殊功能寄存器                           | 44 |
| 3.5.3 | 中断标识符                             | 45 |
| 3.6   | C 语言程序在单片机典型系统中的应用                | 45 |
| 3.6.1 | 8×8 LED 点阵显示原理及应用举例               | 45 |
| 3.6.2 | 大屏幕 LED 点阵显示屏工作原理及典型应用            | 48 |
| 3.7   | 技能实训                              | 49 |
|       | 【实训 3】Proteus 7 Professional 软件入门 | 49 |
|       | 【实训 4】跑马灯                         | 55 |
|       | 习题                                | 58 |
| 第 4 章 | 存储器系统                             | 60 |
| 4.1   | 8051 单片机外部总线的扩展                   | 60 |
| 4.1.1 | 8051 单片机的三总线结构                    | 60 |
| 4.1.2 | 典型 8 位地址总线锁存芯片                    | 61 |
| 4.2   | 存储器概述                             | 62 |
| 4.2.1 | 存储器的分类                            | 62 |
| 4.2.2 | 随机存储器                             | 62 |
| 4.2.3 | 只读存储器                             | 64 |
| 4.3   | 存储器扩展的基本方法                        | 66 |
| 4.3.1 | 存储器扩展需要解决的基本问题                    | 66 |
| 4.3.2 | 单片机存储器的片选技术                       | 67 |
| 4.4   | 技能实训                              | 69 |
|       | 【实训 5】单片机存储器的综合扩展                 | 69 |
|       | 【实训 6】I2 总线串行 EEPROM 24C02 实训     | 70 |
|       | 习题                                | 74 |
| 第 5 章 | 中断技术                              | 75 |
| 5.1   | 中断概述                              | 75 |

|       |                        |     |
|-------|------------------------|-----|
| 5.1.1 | 中断的概念                  | 75  |
| 5.1.2 | 中断的功能及特性               | 76  |
| 5.1.3 | 计算机的中断源                | 76  |
| 5.2   | 8051 单片机的中断系统          | 77  |
| 5.2.1 | 8051 单片机的中断系统的结构       | 77  |
| 5.2.2 | 中断源和中断控制               | 78  |
| 5.2.3 | 中断处理过程                 | 80  |
| 5.2.4 | 外部中断源的扩展               | 81  |
| 5.3   | 技能实训                   | 82  |
|       | 【实训 7】多信号彩灯的中断实现       | 82  |
|       | 【实训 8】方波信号发生器的中断实现     | 84  |
|       | 习题                     | 86  |
| 第 6 章 | 定时/计数器                 | 88  |
| 6.1   | 8051 单片机定时/计数器的构成      | 88  |
| 6.1.1 | 8051 单片机定时/计数器的结构与工作原理 | 88  |
| 6.1.2 | 定时/计数器的相关寄存器           | 89  |
| 6.2   | 8051 单片机定时/计数器的工作方式及应用 | 91  |
| 6.2.1 | 定时/计数器的工作方式            | 91  |
| 6.2.2 | 定时/计数器的应用              | 94  |
| 6.3   | 技能实训                   | 96  |
|       | 【实训 9】自动计数器            | 96  |
|       | 习题                     | 99  |
| 第 7 章 | I/O 设备与接口              | 101 |
| 7.1   | 计算机 I/O 接口技术概述         | 101 |
| 7.1.1 | I/O 设备及 I/O 接口         | 101 |
| 7.1.2 | CPU 与外围设备之间的数据传送方式     | 102 |
| 7.2   | 8051 单片机并行 I/O 接口      | 104 |
| 7.2.1 | P0 口 (80H) 的结构和功能      | 104 |
| 7.2.2 | P1 口 (90H) 的结构和功能      | 106 |
| 7.2.3 | P2 口 (A0H) 的结构和功能      | 106 |
| 7.2.4 | P3 口 (B0H) 的结构和功能      | 107 |
| 7.3   | 8051 单片机并行 I/O 接口的应用   | 109 |
| 7.3.1 | 并行 I/O 接口的基本输入/输出原理    | 109 |
| 7.3.2 | 并行 I/O 接口的驱动能力         | 110 |
| 7.4   | 8051 单片机并行 I/O 接口的扩展   | 110 |
| 7.4.1 | 简单 I/O 接口的扩展           | 110 |





|                                   |            |
|-----------------------------------|------------|
| 7.4.2 通用可编程并行接口 8255A 的结构 .....   | 111        |
| 7.4.3 8051 单片机和 8255A 的接口方法 ..... | 116        |
| 7.5 8 段 LED 数码显示技术 .....          | 117        |
| 7.5.1 LED 数码管工作原理 .....           | 117        |
| 7.5.2 LED 数码管的典型应用 .....          | 118        |
| 7.6 键盘接口技术 .....                  | 119        |
| 7.6.1 键盘接口技术及原理 .....             | 119        |
| 7.6.2 独立式按键和矩阵式键盘 .....           | 120        |
| 7.7 技能实训 .....                    | 122        |
| 【实训 10】多个灯的智能控制 .....             | 122        |
| 【实训 11】8255A 扩展的彩灯控制 .....        | 124        |
| 【实训 12】8 段 LED 数码管的显示控制 .....     | 126        |
| 【实训 13】LED 数码管动态显示的串行驱动 .....     | 128        |
| 【实训 14】矩阵式键盘的按键识别 .....           | 132        |
| 习题 .....                          | 135        |
| <b>第 8 章 串行通信 .....</b>           | <b>136</b> |
| 8.1 串行通信的基本概念 .....               | 136        |
| 8.1.1 并行通信和串行通信 .....             | 136        |
| 8.1.2 串行通信的数据传送方向 .....           | 137        |
| 8.1.3 串行通信的工作方式 .....             | 138        |
| 8.1.4 串行通信的波特率 .....              | 138        |
| 8.2 8051 单片机的串行接口 .....           | 139        |
| 8.2.1 串行接口的结构 .....               | 139        |
| 8.2.2 串行通信所用的专用寄存器 .....          | 139        |
| 8.2.3 8051 单片机串行接口的工作方式 .....     | 140        |
| 8.2.4 8051 单片机串行接口的波特率 .....      | 142        |
| 8.2.5 串行通信的标准 .....               | 144        |
| 8.3 技能实训 .....                    | 145        |
| 【实训 15】串行接口的双机通信应用 .....          | 145        |
| 【实训 16】8051 单片机与 PC 的通信 .....     | 149        |
| 习题 .....                          | 152        |
| <b>第 9 章 A/D 及 D/A 转换接口 .....</b> | <b>154</b> |
| 9.1 D/A 转换接口 .....                | 154        |
| 9.1.1 D/A 转换器的转换原理及主要性能指标 .....   | 154        |
| 9.1.2 8 位 D/A 转换器 DAC0832 .....   | 155        |
| 9.2 A/D 转换接口 .....                | 157        |

|                                                  |     |
|--------------------------------------------------|-----|
| 9.2.1 A/D 转换器的转换原理及主要性能指标 .....                  | 157 |
| 9.2.2 逐次逼近式 A/D 转换器 ADC0809 .....                | 159 |
| 9.2.3 AD1674 及其与 8051 单片机接口技术 .....              | 161 |
| 9.3 串行 A/D、D/A 转换接口 .....                        | 164 |
| 9.4 技能实训 .....                                   | 166 |
| 【实训 17】智能信号发生器 .....                             | 166 |
| 【实训 18】智能电压表 .....                               | 169 |
| 习题 .....                                         | 171 |
| 第 10 章 综合实训 .....                                | 173 |
| 10.1 SMC1602A LCM 应用 .....                       | 173 |
| 10.2 LCD12864 图形显示 .....                         | 177 |
| 10.3 DS18B20 温度控制数码管显示 .....                     | 185 |
| 10.4 DS1302 时钟芯片的应用 .....                        | 191 |
| 附录 A Proteus 7 Professional 中常用元器件名称的中英文对照 ..... | 201 |
| 附录 B 图形符号对照表 .....                               | 204 |
| 参考文献 .....                                       | 205 |



# 第1章

# 数制与编码

## 学习目标:

本章主要介绍计算机中关于数的表示方法、几种常用数制的转换、机器数的表示方法和常用编码等内容。使学生通过对数的基础知识的学习,为后续单片机原理的学习打下基础。

## 知识点:

- (1) 二进制、十六进制、十进制表达形式及其相互转换;
- (2) 机器数中关于有符号数的原码、反码、补码的表达形式及其相互转换;
- (3) ASCII 码、BCD 码的表达形式及其相互转换。

## 1.1 不同进位计数制及其转换

### 1.1.1 进位计数制

计算机其实就是一种由数字电路演变而来的,能进行逻辑运算的机器。其处理的信息就是数字电路中所提到的二进制数,而人们常使用的是十进制数,这样,为了能顺利地在人与计算机之间进行信息交换,一定要进行不同进制数之间的转换操作,因此有必要掌握数制及其转换的原理。

进位计数制:按进位的原则进行计数的一种方法。

进位计数制有以下两个特点:

- (1) 有一个固定的基数  $r$ , 数的每一位只能取  $r$  个不同的数字,即所使用的数码为  $0, 1, 2, \dots, r-1$ 。
- (2) 逢  $r$  进位, 它的第  $i$  个数位对应于一个固定的值  $r^i$ ,  $r^i$  称为该位的“权”。小数点左侧各位的权是基数  $r$  的正次幂,依次为  $0, 1, 2, \dots, m$  次幂;小数点右侧各位的权是基数  $r$  的负次幂,依次为  $-1, -2, \dots, -n$  次幂。

### 1. 十进制

十进制的基数为 10, 它所使用的数码为  $0 \sim 9$ , 共 10 个数字。十进制各位的权是以 10 为底的幂, 即每个数所处的位置不同, 它的值是不同的, 每一位数是其右边相邻那位数的 10 倍。

例如, 数 555.55 就是下列多项式的缩写:

$555.55D = 5 \times 10^2 + 5 \times 10^1 + 5 \times 10^0 + 5 \times 10^{-1} + 5 \times 10^{-2}$ 。式中的后缀 D (Decimal) 表示该数为十进制数, 通常, 十进制数不加后缀。

### 2. 二进制

二进制的基数为 2, 它所使用的数码为  $0, 1$ , 共 2 个数字。二进制各位的权是以 2 为底的幂,

即  $2^n, \dots, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, \dots$ 。

例如，二进制数 1011.101 相当于十进制数 11.625，即  $1011.101\text{B} = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 11.625$ 。式中的后缀 B (Binary) 表示该数为二进制数。

二进制数的运算规则类似于十进制，加法为逢二进一，减法为借一为二。利用加法和减法就可以进行乘法、除法以及其他数值运算。

### 3. 十六进制

十六进制的基数为 16，它所使用的数码共有 16 个：0~9、A~F，其中 A~F 相当于十进制数的 10~15。十六进制各位的权是以 16 为底的幂，即  $16^n, \dots, 16^2, 16^1, 16^0, 16^{-1}, 16^{-2}, \dots$ 。

例如，十六进制数 A3E.8F 相当于十进制数 2622.5059，即  $A3E.8\text{FH} = 10 \times 16^2 + 3 \times 16^1 + 14 \times 16^0 + 8 \times 16^{-1} + 15 \times 16^{-2} = 2622.5059$ 。式中的后缀 H (Hexadecimal) 表示该数为十六进制数。十六进制数如是字母打头，则在使用汇编指令时前面加一个 0。例如：0FFFFH (65535)。

在 C 语言中，十六进制数是在前面加前缀“0x”。

## 1.1.2 数制的转换

### 1. 二进制数、十六进制数转换成十进制数

根据定义，只需将二进制数、十六进制数按权展开后相加即可转换成相应的十进制数。

例如：

$$1011\text{B} = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11; 0A4\text{H} = 10 \times 16^1 + 4 \times 16^0 = 164.$$

### 2. 十进制数转换成二进制数、十六进制数

十进制整数转换成二进制数 (或十六进制数) 时，通常采用“除 2 取余” (或除 16 取余) 法，即用 2 (或 16) 连续除十进制数至商为 0，逆序排列余数即可得到用二进制数 (或十六进制数) 表示的整数值。

十进制小数部分转换成二进制数 (或十六进制数) 时，通常采用“乘 2 取整” (或乘 16 取整) 法，即将小数部分十进制数一次又一次地用纯小数部分乘 2 (或 16)，把每次得到的整数按顺序排列即可得到用二进制数 (或十六进制数) 表示的小数值。

例 1.1：将 62.6875 转换成二进制数。

解：

(1) 十进制整数 62 转换为二进制数，采用“除 2 取余”法。

|        |      |                                                                                                                                                                                                                                                                                                                                                 |
|--------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2   62 | 余数=0 | <div style="display: flex; align-items: center; justify-content: center;"> <div style="width: 20px; height: 20px; border: 1px solid black; margin: 0 auto;"></div> <div style="writing-mode: vertical-rl; text-orientation: upright; margin: 0 5px;">↑</div> <div style="writing-mode: vertical-rl; text-orientation: upright;">低位</div> </div> |
| 2   31 | 余数=1 |                                                                                                                                                                                                                                                                                                                                                 |
| 2   15 | 余数=1 |                                                                                                                                                                                                                                                                                                                                                 |
| 2   7  | 余数=1 |                                                                                                                                                                                                                                                                                                                                                 |
| 2   3  | 余数=1 |                                                                                                                                                                                                                                                                                                                                                 |
| 1      | 余数=1 |                                                                                                                                                                                                                                                                                                                                                 |

按余数的逆序排列，即  $62 = 111110\text{B}$ 。

(2) 十进制小数 0.6875 转换为二进制数, 采用“乘 2 取整”法。

|            |        |    |
|------------|--------|----|
| 0.6875     |        |    |
| $\times 2$ |        |    |
| 1.3750     | 整数部分=1 | 高位 |
| 0.3750     |        |    |
| $\times 2$ |        |    |
| 0.7500     | 整数部分=0 |    |
| 0.7500     |        |    |
| $\times 2$ |        |    |
| 1.5000     | 整数部分=1 |    |
| 0.5000     |        |    |
| $\times 2$ |        |    |
| 1.0000     | 整数部分=1 | 低位 |

按整数的顺序排列, 即  $0.6875 = 0.1011\text{B}$ 。最终结果为  $62.6875 = 111110.1011\text{B}$ 。

### 3. 二进制数与十六进制数的相互转换

在计算机中引入十六进制数的本质: 为了更简化地描述二进制数, 即十六进制就是二进制的压缩码, 即 1 位十六进制数等同于 4 位二进制数。所以转换时, 只要根据这个原则就可很容易地实现二者之间的转换操作。

(1) 二进制数转换成十六进制数。转换时, 将二进制数整数部分由右向左每 4 位一段, 最后不足部分左面补零; 小数部分由左向右每 4 位一段, 最后不足部分右面补零, 然后, 每 4 位二进制数用 1 位十六进制数代替, 便转换成了十六进制。例如:

$$1011110101.110\text{B} = 0010\ 1111\ 0101.1100\text{B} = 2\text{F}5.\text{CH}$$

(2) 十六进制数转换成二进制数。转换时, 只要将每位十六进制数用对应的 4 位二进制数代替, 便转换成了二进制数。例如:

$$\text{BD}5.6\text{H} = 1011\ 1101\ 0101.0110\text{B}$$

## 1.2 计算机中数的表示方法及运算

### 1.2.1 机器数的特点

#### 1. 机器数的字长 (位数)

机器数所能表示的数的范围受到计算机字长的限制, 如对于 8 位字长的计算机来说, 机器数的范围为  $(0000\ 0000)_2 \sim (1111\ 1111)_2$ , 即对应十进制数为 0~255。表示一次性能够处理的二进制位数 (8 根导线)。

为了扩大机器数表示的范围, 有时可用 2 个字甚至多个字表示 1 个数, 例如, 对于 8 位字长的计算机来说, 若用 2 个字来表示 1 个正数 (高 8 位, 低 8 位), 其数值范围为 0~65 535。

## 2. 机器数的符号

机器数只用来表示大小, 这时称为无符号数 (unsigned)。当用来既表示正负, 又表示大小时, 这时称为带 (有) 符号数 (signed)。此时, 数的符号在机器中就数码化了, 即将一个字的最高位定为符号位, 其余各位为数值位。最高位为 0 表示正号, 最高位为 1 表示负号。例如:  $N_1 = (+101\ 1001)_2$  的机器数可表示为  $N_1 = (0101\ 1001)_2$ ,  $N_2 = (-110\ 1011)_2$  的机器数可表示为  $N_2 = (1110\ 1011)_2$ 。带符号数也可以用 2 个字来表示 1 个数, 此时, 符号位仍定为 2 个字的最高位。

计算机中数的符号有无取决于用户规定, 而非机器。

### 1.2.2 原码、反码和补码

#### 1. 原码

正数的符号位用 0 表示, 负数的符号位用 1 表示, 其余各位表示数值, 这种表示法称为原码。例如:

$$X_1 = [+100\ 0001]_2 = +65 \qquad [X_1]_{\text{原}} = 0100\ 0001$$

$$X_2 = [-100\ 0001]_2 = -65 \qquad [X_2]_{\text{原}} = 1100\ 0001$$

左边表示的数称为真值, 即为某数的实际算术值; 右边为用原码表示的数, 两者的最高位分别用 0、1 代替符号位的 +、-。

在原码表示法中, 0 有两种表示法, 即  $[0]_{\text{原}} = +0 = 0000\ 0000$ ,  $[0]_{\text{原}} = -0 = 1000\ 0000$ 。

#### 2. 反码

一个数的反码可由原码求得。如果是正数, 则其反码与原码相同; 如果是负数, 则其反码除符号位为 1 外, 其他各数位均按位取反, 即 1 转换为 0, 0 转换为 1。例如:

$$X_1 = +100\ 0001 \qquad [X_1]_{\text{反}} = 0100\ 0001$$

$$X_2 = -100\ 0001 \qquad [X_2]_{\text{反}} = 1011\ 1110$$

如果已知一个数的反码, 求它的真值, 若是正数则可直接求得, 若是负数则可将符号位除外的数值部分各位取反得到负数的原码, 然后再求真值。例如:

$$[X_1]_{\text{反}} = 0100\ 0001 \qquad [X_1]_{\text{原}} = 0100\ 0001 \qquad X_1 = +65$$

$$[X_2]_{\text{反}} = 1011\ 1110 \qquad [X_2]_{\text{原}} = 1100\ 0001 \qquad X_2 = -65$$

在反码表示法中, 0 也有两种表示形式, 即  $[0]_{\text{反}} = +0 = 0000\ 0000$ ,  $[0]_{\text{反}} = -0 = 1111\ 1111$ 。

#### 3. 补码

一个数的补码一般由反码求得。如果是正数, 则其补码与原码相同; 如果是负数, 则其补码为反码加 1, 即“取反后再加上 1”。例如:

$$X_1 = +100\ 0001 \qquad [X_1]_{\text{补}} = 0100\ 0001$$

$$X_2 = -100\ 0001 \qquad [X_2]_{\text{补}} = [X_2]_{\text{反}} + 1 = 1011\ 1110 + 1 = 1011\ 1111$$

“取反加 1”需要做两步运算, 这个过程也可以简化为一步, 即符号位不变, 只对原码各位中最低一位 1 以左的各位求反, 而最低一位 1 和右边各位都不变, 即可得到负数的补码。

已知  $X$  的补码, 求  $X$  的原码时, 可以将  $X$  的补码当作  $X$  的原码形式, 再求一次  $X$  的补码得到, 即  $[[X]_{\text{补}}]_{\text{补}} = [X]_{\text{原}}$ 。

例 1.2: 已知  $[X]_{\text{补}} = 1011\ 1111$ ,  $[Y]_{\text{补}} = 0001\ 0110$ , 求  $X$ 、 $Y$  的真值。

解:

(1)  $[X]_{\text{原}} = [[X]_{\text{补}}]_{\text{补}} = [1011\ 1111]_{\text{补}} = 1100\ 0000 + 1 = 1100\ 0001$ 。

因为  $X$  为负数, 所以  $X$  的真值为  $-100\ 0001\text{B} = -65$ 。

(2)  $[Y]_{\text{原}} = [[Y]_{\text{补}}]_{\text{补}} = [0001\ 0110]_{\text{补}} = 0001\ 0110$ 。因为  $Y$  为正数, 所以  $Y$  的真值为  $0001\ 0110\text{B} = +22$ 。

在补码表示法中, 0 只有一种表示形式, 即  $[0]_{\text{补}} = 0000\ 0000$ 。

对于 8 位有符号数来说, 用补码所表示的数的范围为  $-128 \sim +127$ , 其中,  $-128$  的补码为  $1000\ 0000$ 。16 位数的补码范围是  $-32\ 768 \sim +32\ 767$ 。

需要指出的是, 在计算机中, 所有有符号数均以补码形式表达。因为计算机在数据处理时, 特别是数据运算时, 是把符号位看作数值位来运算的, 只有补码运算才能保证运算结果的正确性, 如下面两个数的加法运算:

$$\begin{array}{r} 1000\ 0001 \\ +\ 0000\ 0001 \\ \hline 1000\ 0010 \end{array} \quad \begin{array}{l} ; (-127) \\ ; (+1) \\ ; (-126) \end{array}$$

如果把上述两数看作无符号数, 则是  $129+1=130$ , 结果显然是正确的。

如果把上述两数看作有符号数, 则  $1000\ 0001$  是  $-127$  的补码,  $0000\ 1000$  是  $+1$  的补码, 而结果  $1000\ 0010$  ( $1000\ 0001 \sim 1111\ 1110$ ) 正好是  $-126$  的补码, 显然, 结果也是正确的。

注意:

如果用十进制描述, 一个  $N$  位二进制负数的补码与其原码的关系为补码  $= 2^N - \text{原码的绝对值}$ 。如 8 位二进制数  $-127$  补码与其原码的关系为  $256 - 127 = 129$ 。即 129 为  $-127$  的补码, 这是在 C 语言中经常用的格式。

## 1.3 BCD 码及 ASCII 码

### 1.3.1 8421BCD 编码

BCD (Binary Coded Decimal) 编码就是用二进制代码表示的十进制数, 即计算机中的十进制数。在 8421BCD 码中, 是用 4 位二进制数  $0000 \sim 1001$  给  $0 \sim 9$  这 10 个数字编码。具体关系见表 1.1。

表 1.1 8421BCD 码与十进制数的关系

| 8421BCD 码 | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 |
|-----------|------|------|------|------|------|------|------|------|------|------|
| 十进制       | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    |

利用表 1.1 可以很容易地实现 8421BCD 码与十进制数之间的转换。

例如:  $(0100\ 1001\ 0111)_{\text{BCD}} = 497$ 。

8421BCD 码不是二进制数, 它只是一种编码, 如果要转换为二进制数, 要先转换为十进制数, 然后再转换为二进制数, 反之过程类似。

### 1.3.2 ASCII 编码

字母与字符用二进制编码的方法很多。目前计算机中用得最广泛的字符编码, 是由美国国家

标准局制定的 ASCII 码( American Standard Code for Information Interchange,美国标准信息交换码 ), 它已被国际标准化组织 ( ISO ) 定为国际标准, 称为 ISO 646 标准。适用于所有西文字母, ASCII 码的二进制位数共有 8 位, 也就是所说的 1 字节 ( byte, 单位符号为 B ), 但参与编码的位数一般只有 7 位, 在西文编码中, 最高位恒为 0。

因为 1 位二进制数可以表示 (  $2^1$  ) =2 种状态: 0、1; 而 2 位二进制数可以表示 (  $2^2$  ) =4 种状态: 00、01、10、11; 依次类推, 7 位二进制数可以表示 (  $2^7$  ) =128 种状态, 每种状态都唯一地编为 1 个 7 位的二进制码, 对应 1 个字符, 这些码字可以排列成 1 个序号 0 ~ 127 ( 00H~7FH )。所以, ASCII 码是用 7 位二进制数进行编码的, 可以表示 128 个字符。表 1.2 是 ASCII 码表。

表 1.2 ASCII 码表

| 代 码     | 字 符 | 代 码     | 字 符   | 代 码     | 字 符 | 代 码      | 字 符 |
|---------|-----|---------|-------|---------|-----|----------|-----|
| 00H(0)  | NUL | 20H(32) | SPACE | 40H(64) | @   | 60H(96)  | `   |
| 01H(1)  | SOH | 21H(33) | !     | 41H(65) | A   | 61H(97)  | a   |
| 02H(2)  | STX | 22H(34) | "     | 42H(66) | B   | 62H(98)  | b   |
| 03H(3)  | ETX | 23H(35) | #     | 43H(67) | C   | 63H(99)  | c   |
| 04H(4)  | EOT | 24H(36) | \$    | 44H(68) | D   | 64H(100) | d   |
| 05H(5)  | ENQ | 25H(37) | %     | 45H(69) | E   | 65H(101) | e   |
| 06H(6)  | ACK | 26H(38) | &     | 46H(70) | F   | 66H(102) | f   |
| 07H(7)  | GEL | 27H(39) | `     | 47H(71) | G   | 67H(103) | g   |
| 08H(8)  | BS  | 28H(40) | (     | 48H(72) | H   | 68H(104) | h   |
| 09H(9)  | HT  | 29H(41) | )     | 49H(73) | I   | 69H(105) | i   |
| 0AH(10) | LF  | 2AH(42) | *     | 4AH(74) | J   | 6AH(106) | j   |
| 0BH(11) | VT  | 2BH(43) | +     | 4BH(75) | K   | 6BH(107) | k   |
| 0CH(12) | FF  | 2CH(44) | ,     | 4CH(76) | L   | 6CH(108) | l   |
| 0DH(13) | CR  | 2DH(45) | -     | 4DH(77) | M   | 6DH(109) | m   |
| 0EH(14) | SO  | 2EH(46) | .     | 4EH(78) | N   | 6EH(110) | n   |
| 0FH(15) | SI  | 2FH(47) | /     | 4FH(79) | O   | 6FH(111) | o   |
| 10H(16) | SLE | 30H(48) | 0     | 50H(80) | P   | 70H(112) | p   |
| 11H(17) | CS1 | 31H(49) | 1     | 51H(81) | Q   | 71H(113) | q   |
| 12H(18) | DC2 | 32H(50) | 2     | 52H(82) | R   | 72H(114) | r   |
| 13H(19) | DC3 | 33H(51) | 3     | 53H(83) | S   | 73H(115) | s   |
| 14H(20) | DC4 | 34H(52) | 4     | 54H(84) | T   | 74H(116) | t   |
| 15H(21) | NAK | 35H(53) | 5     | 55H(85) | U   | 75H(117) | u   |
| 16H(22) | SYN | 36H(54) | 6     | 56H(86) | V   | 76H(118) | v   |
| 17H(23) | ETB | 37H(55) | 7     | 57H(87) | W   | 77H(119) | w   |
| 18H(24) | CAN | 38H(56) | 8     | 58H(88) | X   | 78H(120) | x   |
| 19H(25) | EM  | 39H(57) | 9     | 59H(89) | Y   | 79H(121) | y   |
| 1AH(26) | SIB | 3AH(58) | :     | 5AH(90) | Z   | 7AH(122) | z   |
| 1BH(27) | ESC | 3BH(59) | ;     | 5BH(91) | [   | 7BH(123) | {   |
| 1CH(28) | FS  | 3CH(60) | <     | 5CH(92) | \   | 7CH(124) |     |
| 1DH(29) | GS  | 3DH(61) | =     | 5DH(93) | ]   | 7DH(125) | }   |
| 1EH(30) | RS  | 3EH(62) | >     | 5EH(94) | ^   | 7EH(126) | ~   |
| 1FH(31) | US  | 3FH(63) | ?     | 5FH(95) | _   | 7FH(127) | DEL |



在 ASCII 码表中,第 0~32 (00H~20H) 号及第 127 (7FH) 号这 34 种编码是控制字符或通信专用字符,如控制字符 LF (换行)、CR (回车)、FF (换页)、DEL (删除)、BEL (振铃) 等;通信专用字符:SOH (文头)、EOT (文尾)、ACK (确认) 等。

第 33~126 号 (21H~7EH) 这 94 种编码是字符编码,其中第 48~57 (30H~39H) 号为 0~9 这 10 个阿拉伯数字;65~90 (41H~5AH) 号为 26 个大写英文字母 A~Z,97~122 (61H~7AH) 号为 26 个小写英文字母 a~z,其余为一些标点符号、运算符号等。例如:数字 0~9 的 ASCII 码可表示为十六进制数 30H~39H,大写字母 A~Z 的 ASCII 码可表示为十六进制数 41H~5AH,小写字母 a~z 的 ASCII 码可表示为十六进制数 61H~7AH,回车符为 0DH,换行符为 0AH 等。

## 1.4 技能实训

### 【实训 1】Keil C51 的使用方法

#### 实训目的

学习 Keil C51 软件的使用方法,主要是针对 C 语言及汇编语言的软件开发及程序调试方法,为后续单片机课程的学习打下良好的基础。

#### 实训内容

围绕一个案例,按工程建立→文件建立→文件添加→参数设置→程序输入→程序编译→程序调试→程序运行(包括单步执行及连续执行)→其他设置等过程,全面学习 Keil C51 软件的使用方法。

#### 实训步骤

准备工作:用户需要事先在计算机中建立一个文件夹。对于有还原卡的计算机,建议最好事先在桌面上建立一个文件夹,如这里先建一个 dz 的文件夹,待全部完成后再上传至服务器保存。

(1) 新建工程。双击桌面上的 Keil  $\mu$ Vision2 图标或单击开始菜单,选择 Keil  $\mu$ Vision2 命令都可启动该程序,其启动界面如图 1.1 所示。几秒后即进入 Keil C51 窗口,如图 1.2 所示。



图 1.1 启动 Keil C51 时的界面

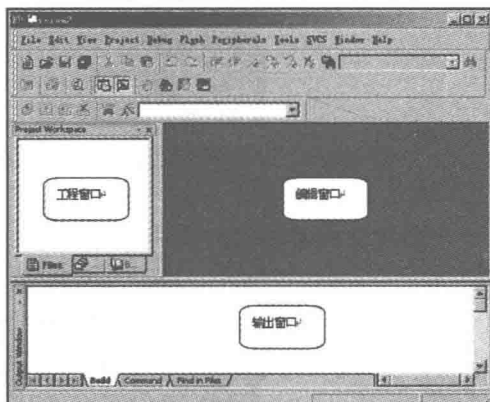


图 1.2 进入 Keil C51 后的窗口