

电气自动化技术快速入门丛书

单片机控制技术 快速入门

陈 洁 陈玉红 编著



Multisim Education Edition

Version 10.1



附 光 盘
CD-ROM



中国电力出版社
CHINA ELECTRIC POWER PRESS

To Completion

电气自动化技术快速入门丛书

单片机控制技术 快速入门

陈 洁 陈玉红 编著



中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 提 要

本书以 MCS-51 内核单片机的基本功能为框架,从计算机基本知识——数的进制及其编码、布尔代数、门电路与组合电路、触发器与时序电路、数字系统简介等着手,进而介绍了 AT89S52、STC90C52RC/RD+、STC11F60XE、STC12C5A60S2 四款 8051 内核单片机的资源,Keil μ Vision2 等三款常用的学习和开发软件,单片机最小系统板等实验器材,汇编语言和 C51 语言的程序设计方法及基本接口电路等。主要内容包括计算机基础知识、认识 8051 单片机、常用软件介绍、实验工具的制作与使用、程序设计基础、延时程序、键盘和显示、中断系统、串行口通信、A/D 转换与 PWM。书中以汇编语言为主,兼顾 C51 语言,列举的实例大多给出了 PROTEUS 仿真,以及在单片机最小系统板上搭建接口电路进行实验的方法和步骤,内容深入浅出、图文并茂、可操作性强。

本书适合广大生产一线的电气与电子工程技术人员及爱好者学习之用,也可作为电子电工类相关专业师生的参考资料。

图书在版编目 (CIP) 数据

单片机控制技术快速入门/陈洁,陈玉红编著. —北京:
中国电力出版社,2015.4

(电气自动化技术快速入门丛书)

ISBN 978-7-5123-7166-8

I. ①单… II. ①陈… ②陈… III. ①单片微型计算机-计
算机控制 IV. ①TP368.1

中国版本图书馆 CIP 数据核字 (2015) 第 022668 号

中国电力出版社出版、发行

(北京市东城区北京站西街 19 号 100005 <http://www.cepp.sgcc.com.cn>)

北京市同江印刷厂印刷

各地新华书店经售

*

2015 年 4 月第一版 2015 年 4 月北京第一次印刷

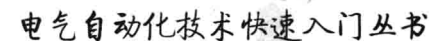
787 毫米×1092 毫米 16 开本 37.25 印张 1006 千字

印数 0001—3000 册 定价 80.00 元 (含 1CD)

敬 告 读 者

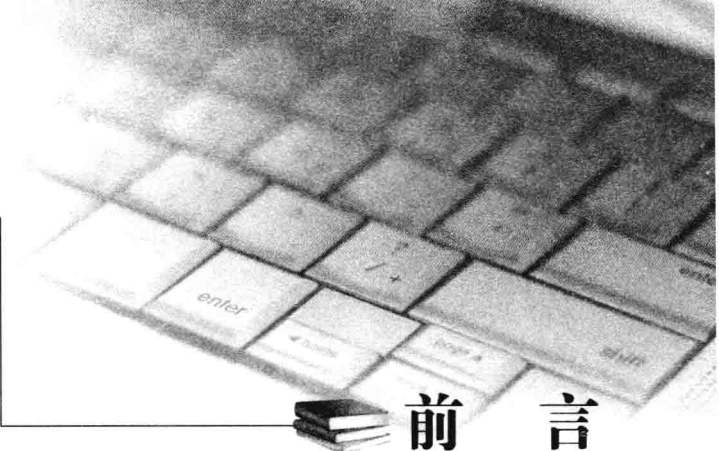
本书封底贴有防伪标签,刮开涂层可查询真伪
本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究



电气自动化技术快速入门丛书

单片机控制技术快速入门



前言

本书是电气自动化技术快速入门丛书之一，是一本以学习单片机在电气控制技术中的应用为主的入门书。书中内容以 MCS-51 内核单片机的基本功能为框架，从计算机基础知识着手，以实例来说明单片机指令和功能的应用，书中列举的实例大多给出了 PROTEUS 仿真，以及在单片机最小系统板上搭建接口电路进行实验的方法和步骤，图文并茂，可操作性强，使具有一定电气技术知识、初中基础以上的读者就能够按照书中内容按部就班地进行单片机应用技术的学习，在动手实践中逐步掌握单片机应用的基本知识，从而初步掌握单片机的应用技术。书中内容注重基础知识结合应用实例，强调动手实践能力，可供广大电工或电子工程技术人员及爱好者学习之用，也可作为电子电工类中专、高职、高专等相关专业单片机课程学习和设计的参考用书。

全书共分为 10 章，第 1 章介绍了学习单片机技术的一些必需的基础知识，包括数的进制、布尔代数、常用数字电路等；第 2 章介绍了 MCS-51 单片机的组成结构及指令系统，以及 4 款单片机的资源；第 3 章讨论了单片机应用开发中常用的 3 个软件的使用方法；第 4 章介绍了学习单片机应用技术过程中进行动手实验所需要的器材、工具及仪器仪表等；第 5 章讨论了单片机开发中的汇编语言和 C51 语言程序设计基础；第 6 章介绍了延时程序的设计方法，分别讨论了利用循环指令和定时器进行延时的程序设计；第 7 章讨论了单片机应用系统中经常用到的人机互动方式——键盘和显示；第 8 章讨论了单片机的中断系统；第 9 章讨论了单片机的串行通信功能；第 10 章讨论了单片机的 A/D 和 D/A 转换及 PWM 功能。

本书第 1、2、3 章和第 9、10 章由陈洁编写，第 4、5、6、7、8 章由陈玉红编写，全书由陈洁统稿。编著过程中参考了有关专业的书籍和资料，在此对这些文献资料的作者表示衷心的感谢！

中国电力出版社的编辑对本书的编写工作给予了大力支持，在此表示感谢！

由于编著者水平有限，加之时间仓促，书中难免存在不足之处，恳请广大读者予以批评指正，邮箱：chenzhu_167@126.com。

陈洁 陈玉红

目 录

前言

第 1 章 计算机基础知识	1
1.1 数、进制及编码	1
1.1.1 十进制数	1
1.1.2 二进制数	1
1.1.3 十六进制数和八进制数	2
1.1.4 二进制数与十进制数之间的转换	3
1.2 二进制码	4
1.2.1 二—十进制码 (BCD 码)	4
1.2.2 ASCII 码	5
1.2.3 格雷码	6
1.2.4 带符号数的表示	7
1.2.5 二进制数的算术运算	8
1.2.6 二进制数的逻辑运算	10
1.3 布尔代数	14
1.3.1 基本逻辑运算	14
1.3.2 逻辑函数	14
1.3.3 规则与公式	15
1.3.4 逻辑函数的化简	16
1.4 门电路与组合电路	18
1.4.1 基本门电路	18
1.4.2 组合电路器件	21
1.5 触发器与时序电路	27
1.5.1 基本触发器	27
1.5.2 时序电路器件	29
1.6 数字系统简介	31
1.6.1 数字密码锁	31
1.6.2 单片机密码锁	35
1.6.3 微计算机原理	35

第 2 章 认识 8051 单片机	44
2.1 MCS-51 单片机概述	44
2.1.1 单片机内部总体结构	44
2.1.2 输入输出结构	45
2.1.3 时钟和复位电路	47
2.1.4 存储器结构	48
2.1.5 特殊功能寄存器	48
2.2 MCS-51 指令系统	49
2.2.1 数据传送类指令	50
2.2.2 算术操作类指令	51
2.2.3 逻辑操作类指令	52
2.2.4 布尔变量操作类指令	53
2.2.5 控制转移类指令	54
2.3 AT89S52 单片机资源	55
2.3.1 AT89S52 单片机特性	56
2.3.2 AT89S52 的封装及引脚功能	57
2.3.3 存储器组织	60
2.3.4 特殊功能寄存器	63
2.3.5 时钟电路	70
2.3.6 复位电路和看门狗电路	71
2.4 STC 单片机资源	79
2.4.1 STC90C52RC/RD+	79
2.4.2 STC11F60XE	89
2.4.3 STC12C5A60 系列单片机	100
第 3 章 常用软件介绍	119
3.1 Keil μVision2 软件的使用介绍	119
3.1.1 认识 μ Vision2 软件	119
3.1.2 创建或打开项目	126
3.1.3 新建或打开文件	126
3.1.4 文件添加到项目中	130
3.1.5 项目设置	131
3.1.6 编译链接	139
3.1.7 程序调试	140
3.1.8 观察窗口	154
3.2 Proteus 7.9 软件的介绍	157
3.2.1 认识 Proteus 7.9 界面	157
3.2.2 Proteus 7.9 中的库	164
3.2.3 激励源	165
3.2.4 虚拟仪器	171
3.2.5 仿真电路图绘制	172

3.2.6	源程序文件编辑	175
3.2.7	生成目标代码文件	177
3.2.8	仿真操作	177
3.3	Mulsim 11 软件的使用介绍	179
3.3.1	Multisim 11 工作界面	181
3.3.2	页面属性	181
3.3.3	元器件库	186
3.3.4	虚拟仪器	189
3.3.5	仿真电路绘制	190
3.3.6	源程序文件编辑	193
3.3.7	MCU 代码管理	194
3.3.8	源程序编译	197
3.3.9	仿真操作	197
第 4 章	实验工具的制作与使用	200
4.1	单片机最小系统	200
4.1.1	电路原理	200
4.1.2	RS-232 简介	207
4.2	并口下载器电路	210
4.2.1	芯片介绍	210
4.2.2	电路原理	212
4.2.3	并行通信口简介	212
4.3	系统板和下载板制作	214
4.3.1	元器件选择	214
4.3.2	制作安装	220
4.4	其他常用工具及仪表	223
4.4.1	常用工具	223
4.4.2	常用仪表	224
4.5	代码下载软件的使用	225
4.5.1	代码下载软件的安装	225
4.5.2	软件工作界面	227
4.5.3	代码读取与下载	228
4.6	STC-ISP 下载软件的使用	231
4.6.1	器材连接	231
4.6.2	软件使用	231
第 5 章	程序设计基础	239
5.1	程序的结构	239
5.1.1	程序设计的步骤	239
5.1.2	流程图	240
5.1.3	顺序结构	241
5.1.4	选择结构	241

5.1.5	循环结构	241
5.2	汇编语言程序设计基础	242
5.2.1	常用伪指令	242
5.2.2	语句格式	244
5.2.3	程序风格	244
5.2.4	汇编	246
5.2.5	程序调试	246
5.3	MCS-51 汇编程序基本结构	246
5.3.1	顺序程序	246
5.3.2	分支程序	247
5.3.3	散转程序	249
5.3.4	循环程序	250
5.3.5	子程序和参数传递	251
5.3.6	查表程序	253
5.3.7	数制转换	254
5.4	C51 语言程序设计基础	255
5.4.1	C51 语言的标识符和关键字	255
5.4.2	数据类型	257
5.4.3	地址类型	259
5.4.4	运算操作符、表达式及其优先级	261
5.4.5	函数	262
5.4.6	程序格式	264
5.5	C51 程序结构	265
5.5.1	顺序程序	265
5.5.2	选择结构程序	265
5.5.3	循环结构程序	268
第 6 章	延时程序	271
6.1	循环指令延时程序	271
6.1.1	循环结构程序	271
6.1.2	指令解释	272
6.1.3	延时时间量计算	273
6.1.4	延时程序流程图	274
6.1.5	程序调试	274
6.1.6	延时子程序	284
6.1.7	指令解释	288
6.1.8	程序状态字	289
6.2	灯光流水控制	290
6.2.1	控制要求	290
6.2.2	实现电路	291
6.2.3	编程思路	292

6.2.4	程序编制	292
6.2.5	指令解释	295
6.2.6	程序调试	297
6.2.7	其他变化方式的流水灯程序	303
6.3	定时器延时程序	308
6.3.1	定时器/计数器 T0 和 T1	308
6.3.2	20ms 定时程序	311
6.3.3	程序解释	312
6.3.4	延时程序流程图	313
6.3.5	程序调试	313
6.3.6	定时器/计数器 T2	320
6.3.7	方波信号产生程序	322
6.4	延时方法小结	329
6.4.1	循环程序延时	330
6.4.2	定时器延时	331
第 7 章	键盘和显示	333
7.1	键盘	333
7.1.1	独立键盘	333
7.1.2	矩阵键盘	335
7.2	LED 数码管	336
7.2.1	数码管显示开关状态	336
7.2.2	按钮进行加/减数	349
7.2.3	矩阵键盘和键值显示	360
7.2.4	60 秒倒计时电路	371
7.3	按钮控制继电器的吸合和释放	387
7.3.1	基本要求	387
7.3.2	实现电路	387
7.3.3	编程思路	389
7.3.4	程序编制	389
7.3.5	指令解释	390
7.3.6	调试操作	391
7.4	LCD 显示模块及其应用	399
7.4.1	1602 液晶显示模块	399
7.4.2	12864 液晶显示模块	423
第 8 章	中断系统	440
8.1	中断的概念	440
8.2	MCS-51 中断系统	447
8.2.1	中断请求源及其标志	447
8.2.2	中断控制	449
8.2.3	程序解释	451

8.3 PWM 脉冲发生器	454
8.3.1 控制要求	454
8.3.2 实现电路	454
8.3.3 编程思路	454
8.3.4 程序编制	454
8.3.5 指令解释	458
8.3.6 程序调试	459
8.4 竞赛抢答器	465
8.4.1 控制要求	465
8.4.2 实现电路	465
8.4.3 编程思路	465
8.4.4 程序编制	467
8.4.5 程序调试	469
第 9 章 串行口通信	479
9.1 通信的基本概念	479
9.1.1 通信的分类	479
9.1.2 数据传送的方向	479
9.1.3 信号的调制和解调	480
9.1.4 异步通信的协议	481
9.2 单片机串行接口	483
9.2.1 串行控制寄存器 SCON	484
9.2.2 串行口工作模式	485
9.2.3 串行数据帧和波特率发生器	500
9.3 串行通信实例	505
9.3.1 单片机与三菱 PLC 通信	505
9.3.2 单片机间主从通信	522
第 10 章 A/D 转换与 PWM	532
10.1 转换器及其主要技术指标	532
10.1.1 A/D 转换器	532
10.1.2 D/A 转换器	533
10.2 A/D 转换芯片及其接口	534
10.2.1 ADC0832 芯片	534
10.2.2 TLC549 芯片	537
10.3 STC12C5A60S2 系列单片机的 A/D 转换器	542
10.3.1 转换器的结构	542
10.3.2 与 A/D 相关的寄存器	543
10.3.3 一个转换程序实例	546
10.4 数字电压表	549
10.4.1 TLC549+STC90C52RC	549
10.4.2 STC12C5A60S2	552

10.5 PWM 功能 557

10.5.1 普通 MCS-51 单片机实现 PWM 的方法 557

10.5.2 STC12C5A60S2 的 PWM 功能 560

10.6 D/A 转换方法 574

10.6.1 扩展芯片法 574

10.6.2 PWM 滤波法 578

参考文献 583

计算机基础知识

扳手指进行计数可以说是每个幼儿的第一堂数学课，老师当然是孩子的父母。伸出自己的双手，从扳下第一个手指开始数数，直到把两只手的手指都扳到，才数到“10”，同时我们也知道一双手有两只手。这是我们无意中最早接触到的数的进制问题。在我们的日常生活中诸如此类的情形还真不少，如一副手套有两只，一条香烟是十包，坐满一张八仙桌有八个人，旧时称满十六两才是一斤，一小时有六十分钟、一年有二十四个节气等。

1.1 数、进制及编码

数制是人们利用符号来计数的科学方法。针对不同类别的计数对象，人们习惯上有不同的进制。生活中经常接触到的数的进制有六十进制、二十四进制、十二进制、十进制、八进制、二进制。

1.1.1 十进制数

日常生活中人们习惯于用十进制数进行计数，十进制是用十个不同的数字 0, 1, 2, 3, ..., 9 来表示数的。任何一个数都可以用上述十个数字按一定规律排列起来表示，其计数规律是“逢十进一”，即 $9 + 1 = 10$ ，这里，右边的“0”为个位数，左边的“1”为十位数，也就是 $10 = 1 \times 10^1 + 0 \times 10^0$ 。所谓十进制就是以 10 为基数的计数体制。因此，某一个数字处于不同的位置(数位)时，它代表的数值是不同的。例如，数 $70 = 7 \times 10^1 + 0 \times 10^0$ ，数 $700 = 7 \times 10^2 + 0 \times 10^1 + 0 \times 10^0$ 。数 567 可写为 $567 = 5 \times 10^2 + 6 \times 10^1 + 7 \times 10^0$ 。

十进制数表示法，也可扩展到表示小数，不过这时小数点右侧的各位数码要乘以基数的负幂次，例如，数 3.142 表示为 $3.142 = 3 \times 10^0 + 1 \times 10^{-1} + 4 \times 10^{-2} + 2 \times 10^{-3}$ 。

一般，任意十进制数 N 可表示为

$$(N)_D = \sum_{i=-\infty}^{\infty} K_i \times 10^i$$

式中： K_i 为基数“10”的第 i 次幂的系数。

1.1.2 二进制数

生活中用于照明的电灯有两种明确的状态，即开和关。灯要么开着点亮，要么关掉熄灭。事件要么是真，要么是假。数字电路或数字系统就是利用晶体管的饱和和截止两种稳定状态来工作的。当遇到计数问题时经常采用二进制数，有时也采用八进制数和十六进制数。二进制数与十进制数的区别在于数码的个数和进位的规律不同，十进制数用 10 个数码，并且“逢十进一”；而二进制数则用两个数码 0 与 1，并且“逢二进一”，即 $1 + 1 = 10$ （读为“壹零”）。必须注意，这里的“10”与十进制数的“10”是完全不同的，它并不代表“拾”。右边的“0”表示 2^0 位的数，左边的“1”表示 2^1 位的数，也就是 $10 = 1 \times 2^1 + 0 \times 2^0$ 。因此，所谓二进制就是以 2 为基数的计数体制。二进制数可表示为



$$(N)_B = \sum_{i=-\infty}^{\infty} K_i \times 2^i$$

式中： K_i 为基数“2”的第 i 次幂的系数。

这样，我们可将任一个二进制数转换为十进制数，例如，二进制数 1001 转换为十进制数等于

$$1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9$$

从数字电路的角度看来，采用十进制是不方便的。因为构成数字电路的基本想法是把电路的状态跟数字对应起来，而十进制的 10 个数字，必须由 10 个不同的而且能严格区分的电路状态与之对应，虽然可以把一定电压范围内的某一段分别对应一个数字，但这样将在识别技术上带来许多困难，而且也不经济。因此在数字电路中一般不直接采用十进制。由于二进制具有一定的优点，因此它在计算技术中被广泛采用。

①二进制的数字装置简单可靠，所用元件少。二进制只有两个数码 0 和 1，因此它的每一位数都可以用任何具有两个不同稳定状态的元件来表示，如三极管的饱和与截止、继电器接点的闭合和断开、灯泡的亮和不亮等。只要规定其中一种状态表示 1，另一种状态表示 0，就可以表示二进制数。这样，数码的存储和传送，就可以用简单而可靠的方式进行。

②二进制的基本运算规则简单，运算操作简便。

但是，采用二进制也有一些缺点。用二进制表示一个数时位数较多，例如，十进制数 49 表示为二进制数时，即为 110001，使用起来不方便也不习惯。因此，在运算时原始数据大多采用人们习惯的十进制数，送入机器由机器将十进制的原始数据转换成数字系统能接受的二进制数。而在运算结束后，再将二进制数转换为十进制数，表示最终结果。

1.1.3 十六进制数和八进制数

表达同一个数值时，使用二进制数的位数很多，不便书写和记忆，因此在数字计算机系统中常采用十六进制数或八进制数来表示二进制数。上述十进制和二进制的表示法可以推广到十六进制数和八进制数。

十六进制数采用十六个数码，而且“逢十六进一”。这种数制中有 16 个不同的数字：0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (对应于十进制中的 10)、B (11)、C (12)、D (13)、E (14)、F (15)。它是以 16 为基数的计数体制。例如，将十六进制数 4E6 转换为十进制数为

$$4 \times 16^2 + 14 \times 16^1 + 6 \times 16^0 = 1254$$

十六进制与二进制之间的转换也比较方便。例如， $(01011001)_B$ 写成十六进制数为

$$(0101 \ 1001)_B = [(1 \times 2^2 + 1 \times 2^0) \times 16^1 + (1 \times 2^3 + 1 \times 2^0) \times 16^0]_D = (59)_H$$

可以看出，每 4 位二进制数对应于 1 位十六进制数，如

$$(1001 \ 1100 \ 1011 \ 0100)_B = (9CB48)_H$$

同理，对于八进制数，可将 3 位二进制数分为一组，对应于 1 位八进制数。对于上述二进制数，可写成 $(10 \ 011 \ 100 \ 101 \ 101 \ 001 \ 000)_B = (2345510)_O$

为便于对照，将十进制、二进制、八进制及十六进制之间的关系列于表 1-1 中。

表 1-1 几种数制之间的关系对照表

十进制数	二进制数	八进制数	十六进制数
0	00000	00	00
1	00001	01	01
2	00010	02	02



续表

十进制数	二进制数	八进制数	十六进制数
3	00011	03	03
4	00100	04	04
5	00101	05	05
6	00110	06	06
7	00111	07	07
8	01000	10	08
9	01001	11	09
10	01010	12	0A
11	01011	13	0B
12	01100	14	0C
13	01101	15	0D
14	01110	16	0E
15	01111	17	0F
16	10000	20	10

1.1.4 二进制数与十进制数之间的转换

既然同一个数值可以用二进制和十进制两种不同形式来表示,那么两者之间就必然有一定的转换关系。现以二、十进制之间的转换为例来说明不同数制之间的转换方法。对于十进制整数 N 可用二进制表示成

$$(N)_D = b_n \times 2^n + b_{n-1} \times 2^{n-1} + \cdots + b_1 \times 2^1 + b_0 \times 2^0$$

式中: b_n 、 b_{n-1} 、 $\cdots$ 、 b_1 、 b_0 是二进制数各位的数字。将等式两边除以 2, 得

$$\frac{1}{2} (N)_D = b_n \times 2^{n-1} + b_{n-1} \times 2^{n-2} + \cdots + b_1 \times 2^0 + b_0 \times 2^{-1}$$

由此可知, 将十进制数除以 2, 其余数为 b_0 。将上式的商再除以 2, 得

$$\frac{1}{2^2} (N)_D = b_n \times 2^{n-2} + b_{n-1} \times 2^{n-3} + \cdots + b_2 \times 2^0 + b_1 \times 2^{-1}$$

其余数为 b_1 。不难推知, 将十进制整数每除以一次 2, 就可根据余数得到二进制数的一位数字。因此, 只要连续除以 2 直到商为 0, 就可由所有的余数求出二进制数。例如, 根据上述原理可将 $(37)_D$ 按如下的步骤转换为二进制数

$$\begin{array}{rcl}
 2 \overline{) 37} & \cdots \cdots & \text{余 } 1 \cdots \cdots b_0 \\
 2 \overline{) 18} & \cdots \cdots & \text{余 } 0 \cdots \cdots b_1 \\
 2 \overline{) 9} & \cdots \cdots & \text{余 } 1 \cdots \cdots b_2 \\
 2 \overline{) 4} & \cdots \cdots & \text{余 } 0 \cdots \cdots b_3 \\
 2 \overline{) 2} & \cdots \cdots & \text{余 } 0 \cdots \cdots b_4 \\
 2 \overline{) 1} & \cdots \cdots & \cdots \cdots b_5
 \end{array}$$

所以 $(37)_D = (b_5 b_4 b_3 b_2 b_1 b_0)_B = (100101)_B$ 。

对于小数可写成

$$(N)_D = b_{-1} \times 2^{-1} + b_{-2} \times 2^{-2} + \cdots + b_{-(n-1)} \times 2^{-(n-1)} + b_{-n} \times 2^{-n}$$

将上式两边分别乘以 2, 得

$$(N)_D = b_{-1} \times 2^0 + b_{-2} \times 2^{-1} + \cdots + b_{-(n-1)} \times 2^{-(n-2)} + b_{-n} \times 2^{-(n-1)}$$

由此可见, 将十进制小数乘以 2, 取其个位数为 b_{-1} 。不难推知, 将十进制小数每次除去上次所得积中之个位数连续乘以 2, 直到满足误差要求进行“四舍五入”为止, 就可将十进制小数转换成二进制小数。例如, 可按如下步骤将 $(0.706)_D$ 转换成误差 ϵ 不大于 2^{-10} 的二进制小数。

$$0.706 \times 2 = 1.412 \cdots \cdots 1 \cdots \cdots b_{-1}$$

$$0.412 \times 2 = 0.824 \cdots \cdots 0 \cdots \cdots b_{-2}$$

$$0.824 \times 2 = 1.648 \cdots \cdots 1 \cdots \cdots b_{-3}$$

$$0.648 \times 2 = 1.296 \cdots \cdots 1 \cdots \cdots b_{-4}$$

$$0.296 \times 2 = 0.592 \cdots \cdots 1 \cdots \cdots b_{-5}$$

$$0.592 \times 2 = 1.184 \cdots \cdots 1 \cdots \cdots b_{-6}$$

$$0.184 \times 2 = 0.368 \cdots \cdots 1 \cdots \cdots b_{-7}$$

$$0.368 \times 2 = 0.736 \cdots \cdots 1 \cdots \cdots b_{-8}$$

$$0.736 \times 2 = 1.472 \cdots \cdots 1 \cdots \cdots b_{-9}$$

由于最后的小数小于 0.5, 根据“四舍五入”的原则, b_{-10} 应为 0。所以, $(0.706)_D = (0.101101001)_B$, 其误差 $\epsilon < 2^{-10}$ 。

对于不同数制的数, 为了在书写上能够加以区别, 常在数的后面用字母来表示不同的数制。字母“B”代表二进制, “Q”(或“O”)代表八进制, “D”代表十进制, “H”代表十六进制。

1.2 二进制码

数字系统中的信息可分为两类, 一类是数值, 另一类是文字符号(包括控制符)。数值信息的表示方法已如前述。为了表示文字符号信息, 往往也采用一定位数的二进制数码来表示, 这个特定的二进制码称为代码。建立这种代码与十进制数值、字母、符号的一一对应关系的过程称为编码。

若所需编码的信息有 N 项, 则需要的二进制数码的位数 n 应满足如下关系: $2^n \geq N$, 下面介绍几种常见的码。

1.2.1 二—十进制码 (BCD 码)

在这种编码中, 用 4 位二进制数 $b_3 b_2 b_1 b_0$ 来表示十进制数中的 0~9 这 10 个数字中的某一个。若这 10 个数码与自然二进制数一一对应, 则编码关系见表 1-2 中左边一列。二进制数码每位的值称为权或位权, b_0 位的权为 $2^0 = 1$, b_1 位的权为 $2^1 = 2$, b_2 位的权为 $2^2 = 4$, b_3 位的权为 $2^3 = 8$ 。例如, 二进制码 0111 所表示的十进制数为 $8 \times 0 + 4 \times 1 + 2 \times 1 + 1 \times 1 = 7$, 因此这种 BCD 码称为 8421 BCD 码, 它是一种最基本的 BCD 码, 应用也较为普遍。在一般情况下, 十进制码与二进制码之间可用下式来表示:

$$(N)_D = W_3 b_3 + W_2 b_2 + W_1 b_1 + W_0 b_0 \quad (1)$$

式中: $W_3 \sim W_0$ 为二进制码中各位的权。

8421 BCD 码是由 4 位二进制数的 0000 (0) 到 1111 (15) 十六种组合中的前十种组合, 即 0000 (0) ~ 1001 (9), 其余六种组合是无效的。十六种组合中选取十种有效组合方式的不同, 可以得到其他二—十进制码, 如表 1-2 中的 2421 码、5421 码等。余 3 码是由 8421 码加 3 (0011) 得来的, 不能用式 (1) 来表示其编码关系, 因而它是一种无权码。



表 1-2

几种常见的码

$b_3 b_2 b_1 b_0$ $2^3 2^2 2^1 2^0$	代码对应的十进制数				
	自然二进制码	二—十进制数			
		8421 码	2421 码	5421 码	余三码
0000	0	0	0	0	
0001	1	1	1	1	
0010	2	2	2	2	
0011	3	3	3	3	0
0100	4	4	4	4	1
0101	5	5			2
0110	6	6			3
0111	7	7			4
1000	8	8		5	5
1001	9	9		6	6
1010	10			7	7
1011	11		5	8	8
1100	12		6	9	9
1101	13		7		
1110	14		8		
1111	15		9		

十进制数 67.34 和 76.43 的 8421BCD 码分别是 $(0110\ 0111.0011\ 0100)_{\text{BCD}}$ 和 $(0111\ 0110.0100\ 0011)_{\text{BCD}}$ 。

1.2.2 ASCII 码

ASCII 码就是美国标准信息交换码。它使用 7 位二进制代码来表示,故可表示 128 个不同字符,它们分别是 26 个大写英文字母、26 个小写英文字母、10 个十进制数字、7 个标点符号、9 个运算符号、50 个其他符号(打印格式符、控制符等),见表 1-3。如要确定某个数字、字母或符号的 ASCII 码,可以先在表 1-3 中找到这个字符,然后将该字符所在列的 3 位二进制数与行所对应的 4 位二进制数左右排列起来,得到的 7 位二进制代码就是该字符的 ASCII 码,如字符“1”的 ASCII 码为 011 0001,小写英文字符“c”、“h”、“e”、“n”的 ASCII 码分别为 110 0011、110 1000、110 0101、110 1110。

表 1-3

ASCII 码(美国标准信息交换码)表

位	000	001	010	011	100	101	110	111
0000	NUL	DEL	SP	0	@	P	,	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u



续表

位	000	001	010	011	100	101	110	111
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(	8	H	X	h	x
1001	HT	EM	)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[	k	{
1100	FF	FC	,	<	L	\	l	
1101	CR	GS	_	=	M	]	m	}
1110	SO	RS	.	>	N	`	n	~
1111	SI	US	/	?	O	_	o	DEL

在计算机中传送 ASCII 码时,通常采用 8 位二进制数码,其最高位用作奇偶校验位,以便于检查代码在传送过程中是否发生差错。

1.2.3 格雷码

实用上,还有一种常见的无权码叫格雷码,格雷码有许多种,这种码的特点是:相邻的两个码组之间仅有一位不同。其中一种典型的编码如表 1-4。在典型的 n 位格雷码中,0 与最大数 (2^n-1) 之间也仅有一位不同,故它是一种“循环码”。格雷码的这个特点使它在代码形成与传输中引起的误差较小,因此常用于模拟量的转换中。当模拟量发生微小变化而可能引起数字量发生变化时,格雷码仅改变一位,这样与其他码同时改变两位或多位的情况相比更为可靠,即可减少出错的可能性。

表 1-4

格雷码与二进制码对照表

十进制数	二进制码	格雷码
	b_3 b_2 b_1 b_0	G_3 G_2 G_1 G_0
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 1
3	0 0 1 1	0 0 1 0
4	0 1 0 0	0 1 1 0
5	0 1 0 1	0 1 1 1
6	0 1 1 0	0 1 0 1
7	0 1 1 1	0 1 0 0
8	1 0 0 0	1 1 0 0
8	1 0 0 1	1 1 0 1
10	1 0 1 0	1 1 1 1
11	1 0 1 1	1 1 1 0
12	1 1 0 0	1 0 1 0
13	1 1 0 1	1 0 1 1
14	1 1 1 0	1 0 0 1
15	1 1 1 1	1 0 0 0