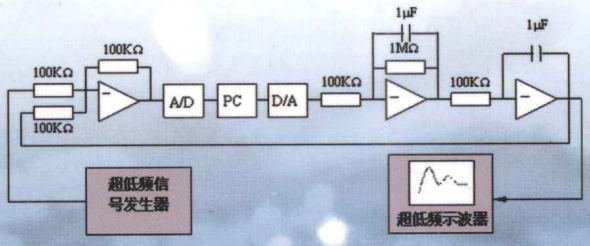
普通高等教育“十一五”电气信息类规划教材

DIANQI XINXILEI

C语言在测量与控制中的应用

■ 王彤 编著



机械工业出版社
CHINA MACHINE PRESS

免费电子课件

普通高等教育“十一五”电气信息类规划教材

C语言在测量与控制中的应用

王 彤 编著
杨旭东 主审

机械工业出版社

本书由浅入深地从编程实践的角度介绍了C语言在测量与控制领域应用的基本方法。这些内容通常是一般C语言程序设计书籍所不涉及，却是测控领域工程技术人员需要掌握的。本书主要包括输入/输出端口的控制，硬件中断程序设计，软件中断程序设计，精确定时程序设计，串口通信程序设计，曲线的绘制，并行接口的使用与步进电动机的控制，数据采集与处理程序设计，闭环控制系统程序设计等内容。书中举了大量的实例，程序简单实用。由于编程涉及测控系统中的硬件设备，所以对一些常用芯片、器件、装置的原理和使用方法进行了简单的介绍。本书还介绍了数据采集与处理系统、计算机闭环控制系统的基本工作原理和设计方法。

本书还包括了实验指导书的内容。实验内容包括基本实验和扩展实验两部分，以利于根据学生的情况因材施教。

本书配有免费电子课件，欢迎选用本书作教材的老师登录 www.cmpedu.com 注册下载或发邮件到 wbj@cmpbook.com 索取。

本书可作为理工科专业研究生或电类专业本科生的教材，授课40学时左右，实验16学时。本书也可作为工程技术人员的一本实用性较强的参考书。

图书在版编目（CIP）数据

C语言在测量与控制中的应用/王彤编著. —北京：机械工业出版社，2009.2

普通高等教育“十一五”电气信息类规划教材

ISBN 978-7-111-26190-2

Ⅰ.C… Ⅱ.王… Ⅲ.①电气测量-C语言-程序设计-高等学校-教材②电气控制-C语言-程序设计-高等学校-教材 Ⅳ.TM921.5-39 TM930.9

中国版本图书馆CIP数据核字（2009）第014414号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）

责任编辑：王保家 版式设计：霍永明 责任校对：陈立辉

封面设计：张 静 责任印制：乔 宇

北京四季青印刷厂印刷（三河市兴旺装订厂装订）

2009年4月第1版第1次印刷

184mm×260mm · 13.25印张 · 328千字

标准书号：ISBN 978-7-111-26190-2

定价：24.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

销售服务热线电话：（010）68326294

购书热线电话：（010）88379639 88379641 88379643

编辑热线电话：（010）88379727

前　言

C 语言是广泛应用的一种程序设计语言，许多计算机控制的实时测控系统的软件是由 C 语言实现的。C 语言应用在实时测控系统时，必然要涉及端口输入/输出控制、硬件中断程序设计和软件中断程序设计。而且，实时测控系统程序设计还经常遇到对一些常用芯片的控制，如串行口、并行口及 A/D 转换器（模/数转换器）、D/A 转换器（数/模转换器）接口的编程等方面的问题。但是，现在 C 语言程序设计的课程和教材通常不涉及这方面的内容，使许多学过 C 语言程序设计的理工科学生却不会编写控制硬件的程序。本书就是想要帮助读者解决这样一些问题。

哈尔滨工业大学硕士研究生的“C 语言在测量与控制中的应用”和本科生的“C 语言测控系统程序设计”选修课已经讲授多次。这些课程都是实践性很强的课程，配有丰富的实验，有利于提高学生的工程实践能力和创新能力。本书是在这些课程的多年教学基础之上编写而成的。

本书注意软件与硬件结合、理论与实践结合，由浅入深地讲解了 C 语言在测控领域应用的方法和特点。书中的计算机采用使用数量最多的 PC 系列微机，并尽量利用 PC 本身的硬件资源，如 8259A 中断控制器、8254 定时器/计数器、打印机接口、串行口等。编程主要采用使用人数最多的 Turbo C 2.0 版本，使本书内容的通用性更好。这样的内容安排，可以使读者动手实践的条件比较容易实现。另外，本书还有配套的实验和多媒体课件可供参考。

本书由王彤编写，杨旭东老师主审。在本书的编写过程中，得到王宝祥、刘志远、陈兴林、伊国兴、张卯瑞、马广程、张广莹、周荻、魏绍义、王强、罗晶、何朕、宋申民、刘贵栋等老师和葛敬嘉、邵春涛、何峰、孙名嘉同学的大力支持和帮助，在此表示衷心的感谢。

本书配有免费电子课件，欢迎选用本书作教材的老师登录 www. cmpedu. com 注册下载或发邮件到 wbj@ cmpbook. com 索取。

由于作者水平有限，加之时间仓促，书中肯定存在错误或不妥之处，恳请读者不吝赐教，批评指正，以便共同探讨，共同提高。

作者

于哈尔滨工业大学 航天学院

作者通信地址 哈尔滨工业大学 329 信箱 邮政编码 150001

E-mail：h7341@ 163. com

目　录

第1章 绪 论

1.1 C语言的特点

C语言是目前使用最广泛的一种程序设计语言，也是测量、控制和通信等领域中最常用的一种程序设计语言。

C语言有如下一些特点：

1）语言表达能力强。C语言表达能力强而灵活，它既有面向硬件和系统，像汇编语言那样可以直接访问硬件的功能，又有高级语言面向用户，容易理解，便于阅读和书写的优点。

2）模块化能力强。C语言程序由函数形式组成，十分有利于把整个程序分割成若干个功能相对独立的程序模块，并且为程序模块之间相互调用和参数传递提供了方便。

3）数据类型丰富。C语言具有现代语言的各种数据类型，基本的数据类型有字符型（char）、整型（int）、长整型（long）、浮点型（float）、双精度型（double），还有无符号字符型（unsigned char）、无符号整型（unsigned int）、无符号长整型（unsigned long）等。在这些基础上可以产生各种构造类型，如数组、指针、结构、联合等。利用这些数据类型可以实现复杂的数据结构，如链表、树等。

4）运算符丰富。C语言的运算符包括的范围很广，除了包括一般高级语言中的算术运算符、逻辑运算符、关系运算符之外，还具有位运算符、指针运算符等。所以C语言的数据处理能力强，具有其他高级语言难以实现的一些功能。

5）可移植性好。C语言本身不依赖于机器硬件，在使用不同CPU的计算机上，C语言程序差别不是很大。这一点与汇编语言不同，汇编语言是一种面向机器的低级语言，汇编语言的许多指令是针对CPU的结构而设计的。因而，不同CPU的汇编语言指令差别较大，移植比较麻烦。

C语言可以在许多操作系统环境下运行，UNIX、DOS、Windows、Linux环境下都可以使用C语言。

6）提供丰富的库函数。每一种C语言版本都提供了丰富的库函数可供用户选择使用，这就大大简化了程序设计的工作。这些库函数通常包括数学函数、字符和字符串操作函数、输入/输出函数、图形函数等。不同的C语言版本还根据计算机硬件和操作系统的环境扩充了一些非标准的库函数，使C语言的功能更完善。

7）执行速度快。C语言的目标代码质量高，执行速度快。以Turbo C为例，其目标代码的效率仅比汇编语言低10%～20%。Turbo C具有寄存器型变量（register），不像其他变量存储在内存中，而是存储在CPU的寄存器中，所以寄存器型变量的存取速度比其他变量更快。由于C语言的执行速度快，所以被广泛使用于对实时性要求较高的实时测控系统程序设计中。

8）控制硬件处理中断的能力强。常用的 C 语言版本在标准 C 函数 ANSI C 的基础上扩充了一些非标准的库函数以增强 C 语言的功能，例如 Turbo C，Borland C++ 扩充了一些控制硬件和处理中断的函数，使其在测控领域得到广泛的应用。这些扩充的函数包括适用于 Intel 86 系列 CPU 的端口输入/输出函数、内存单元读/写函数、处理中断特别是处理硬件中断的函数、串口通信函数，以及关于定时的一些函数等。

另外，C 语言有按二进制的位进行操作的功能，这在控制硬件时特别有用。例如，一个接口芯片的 8 位端口每一位都控制一个外设，如果只想改变其中一位或几位的状态，而不改变其他位的状态，位运算符可以方便快速地完成这个操作，而其他高级语言就没有这样的功能。

9）绘图功能强。C 语言提供了大量的绘图函数，并且可以直接控制显卡等硬件，因此，可方便地绘制曲线和图形，在实时测控系统中可实时地显示被测物理量的变化曲线。

10）可与其他语言混合编程。C 语言是一种编译类语言。编译类语言从编写到执行的过程可分为以下几步：首先，输入源文件；其次，对源文件进行编译或汇编生成目标码文件（文件的扩展名为 OBJ）；然后用连接程序对目标码文件进行连接，生成可执行文件（文件的扩展名为 EXE）后，就可以运行这个可执行文件。

可以用连接程序将由 C 语言生成的目标码模块与其他编译类语言（如汇编语言，FORTRAN 语言，PASCAL 语言）生成的目标码模块连接在一起，生成可执行文件。不同语言的混合编程可发挥各自的特长，提高程序的效率。

1.2 工业 PC

随着个人计算机（Personal Computer，PC）的广泛应用，越来越多的人熟悉 PC 系列微机的软硬件结构和使用方法。工业 PC 即工业个人计算机（Industrial Personal Computer，IPC），它是在商用和家用 PC 的设计基础上进行改造，使其适用于工业测量和控制领域的计算机。工业 PC 继承了广大用户所熟悉的个人计算机丰富的软件资源，使其软件开发方便，兼容性好，用户容易掌握。为了适应工业生产环境，工业 PC 在计算机的结构和性能方面进行了许多改进，使其可靠性、抗干扰性大大增强。

工业 PC 的主要特点如下：

1）取消了 PC 的大主板，将原来的大主板改成通用的底板总线插槽系统。底板上基本没有集成电路芯片，主要是 ISA 总线和 PCI 总线的插槽。

2）将原来主板的功能分成几块插件板，如 CPU 板、存储器板等几块小板，并有紧锁加固装置。小板的抗振动、抗冲击性能好，可用于车载、舰载、机载等振动较大的场合。

3）采用全钢结构密封机箱，抗电磁干扰能力强，电磁兼容性好。内部正压送风散热，并装有滤尘装置，可防止灰尘的进入。

4）电源采用工业电源，性能可靠。可根据所带负载的多少，选择不同功率的电源。

5）可配置时间监视器（Watch dog，俗称看门狗），在系统受到强干扰，程序“跑飞”时，自动重新启动用户程序。

6）可配置多任务实时控制操作系统，调度协调系统资源，可同时完成多个实时性较强的工作任务。

7）配有丰富的工业控制软件，而且控制软件正向结构化、组态化方向发展。工业过程控制组态软件通常包括工业调节回路控制算法组态、图形显示功能组态、统计报表组态等。使用组态化的工业控制软件，可以简化应用控制软件的开发，缩短应用软件开发的周期。

8）系统的硬件配置灵活，扩展方便。许多厂商提供模拟量、数字量、开关量输入/输出通道扩展卡或扩展模块，有些还具有光电隔离功能，抗干扰性更强。还可以扩展时钟功能，RS-232C、RS-485、IEEE488 等通信功能，以及热电偶输入信号调理功能等。

9）机箱有各种不同形式，可供不同场合需要的选择。例如，有适合放入控制柜的上架机式箱，有壁挂式机箱，有与显示器组合在一起的一体式工作站，还有便携式的工业控制机、PC104 嵌入式工业控制机等。

10）冗余性好，后备措施齐全。具有后备电源、存储器信息保护、双机冗余切换等功能。

第2章　端口、内存输入/输出与位运算

2.1　端口输入/输出函数

计算机的CPU要与外设间传递信息，必须通过硬件的接口电路来进行。接口电路通常包含一组寄存器，CPU与外设进行数据传递时，数据信息、状态信息、控制信息等不同的信息要进入接口电路中的不同寄存器，一般称这些寄存器为I/O（输入/输出）端口。所以，一个接口电路芯片往往要占用几个I/O端口地址。计算机是通过不同的端口地址来区分接口电路及其寄存器的。86系列CPU的端口寻址能力为64K位，即可以有65536个端口地址，地址的范围为0H～FFFFH。因为$2^{16}=65536$，所以寻址65536个端口地址要用16条地址线。通常PC系列微机上只使用10条地址线对端口寻址，其寻址范围为0H～3FFH，共1024个端口地址。如果端口的数据通道是8位的，称为8位端口，要占一个I/O端口地址。如果端口的数据通道是16位的，称为16位端口，它的I/O端口地址应该是一个偶地址，而且下一个相邻的奇地址也不许其他端口占用。所以一个16位的端口虽然只有一个端口地址，但实际上相当于占用了两个端口地址。

C语言为了实现控制硬件端口的输入/输出扩展了一些非标准的系统函数。Turbo C提供的端口输入/输出函数主要如下。

1. 8位端口输入函数 inportb( )

函数格式：unsigned char inportb(int portid);

功能：从地址为portid的8位端口输入一个字节（8位），所读的值是该函数的返回值。

该函数由头文件dos.h说明。

用法：
```
#include <dos.h>
unsigned char b;
⋮
b = inportb(端口地址);
```

2. 16位端口输入函数 inport( )

函数格式：int inport(int portid);

功能：从地址为portid的16位端口输入一个字（16位），所读的值是该函数的返回值。

该函数由头文件dos.h说明。

用法：
```
#include <dos.h>
int w;
⋮
w = inport(端口地址);
```

3. 8位端口输出函数 outportb( )

函数格式：void outportb(int portid , usigned char value);

功能：从地址为 portid 的 8 位端口输出一个字节（8 位），其值为 value。

该函数由头文件 dos. h 说明。

用法：#include < dos. h >

⋮

outportb(端口地址,无符号字符型数据或变量);

4. 16 位端口输出函数 outport( )

函数格式：void outport(int portid, int value);

功能：向地址为 portid 的 16 位端口输出一个字（16 位），其值为 value。

该函数由头文件 dos. h 说明。

用法：#include < dos. h >

⋮

outport(端口地址,整型数据或变量);

由不同 C 语言版本提供的非标准函数在名称和使用上可能有些差别，使用时应注意。

Microsoft C 提供的端口 I/O 函数如下：

inp( ) 从 8 位端口输入；

inpw( ) 从 16 位端口输入；

outp( ) 从 8 位端口输出；

outpw( ) 从 16 位端口输出。

这几个函数是由头文件 conio. h 说明的。

Turbo C 为了保持兼容性也提供了 inp( )、inpw( )、outp( )、outpw( ) 函数，但这些函数则由头文件 dos. h 说明。

在 Borland C++ 中，函数 inportb( )、inport( )、outputb( )、outport( ) 由头文件 dos. h 说明；而 inp( )、inpw( )、outp( )、outpw( ) 由头文件 conio. h 说明。

2.2 位运算

在计算机中，数据存储的最基本单位是二进制的位（Bit），一个二进制位可以存储一位二进制数 0 或 1。一个字节（Byte）含有 8 个二进制的位，一个字（Word）含有 16 个二进制的位。C 语言中，字符型变量是二进制 8 位的，整型变量是二进制 16 位的，长整型变量是二进制 32 位的。位运算就是直接对操作数的二进制位进行操作，这种操作通常由机器指令和汇编语言完成。C 语言也可以对二进制位进行操作，这反映了 C 语言除了具备高级语言的特点外，还具有低级语言的某些功能。因为有些集成电路芯片端口的每一位可能控制一个外设，所以按二进制位的逻辑运算对于控制计算机的硬件是很有用的。例如，可编程集成电路芯片通常有一个命令字寄存器或控制字寄存器，寄存器的每一位都影响该芯片工作方式的设置，在对这些硬件芯片进行操作时，可能需要对某一位或某几位进行置 1、清 0 或者取反的操作，有时还要测试端口某一位的状态是 0 还是 1 等。使用位运算可以方便、快捷地完成这些操作。

C 语言提供的位运算符有：按位与“&”，按位或“|”，按位异或“^”，按位非“~”，左移“≪”和右移“≫”。这些位运算符只能用于有符号或无符号的字符型、整型、

长整型数据的运算。

2.2.1 按位与运算的应用

按位与运算符是“&”，在进行按位与运算时，要求有两个运算量，例如 a&b 是将 a 和 b 对应的二进制位上的值进行按位与运算。对应的两个位上的值都是 1 时，结果位才是 1；否则结果是 0。二进制对应位之间的运算规则即 a&b 的真值表如表 2-1 所示。

表 2-1 a&b 的真值表

a 位	b 位	结果位
0	0	0
0	1	0
1	0	0
1	1	1

若 a = 0x55，b = 0xf0

则 c = a&b = 0x50

为了便于观察，将 a&b 写成二进制竖式形式

```
  01010101    (a = 0x55)
&) 11110000   (b = 0xf0)
  ----------
  01010000    (c = 0x50)
```

a 的高 4 位跟 1 相与，所以结果保持不变；而 a 的低 4 位跟 0 相与，所以结果都是 0。

按位与运算可以将某些位屏蔽（清 0），而其他位不变。跟 0 相与的位被清 0，跟 1 相与的位保持不变。

例如，要将端口地址为 0x282 的 8 位端口输入的数据高 4 位清 0，低 4 位不变，相关语句如下：

```
#include <dos.h>
char b;
⋮
b = inportb(0x282);
b = b&0x0f;
```

可编程器件通常有一个状态寄存器，每一位用 0 或 1 表示该芯片的某种状态，例如 A/D 转换是否完成、串行通信线路是否有错、接收数据是否就绪、是否可以发送新的字符等。这就要测试某一位是 0 还是 1。如果状态寄存器是 8 位的，可以设一个 8 位的常量作为一个屏蔽字节。屏蔽字节中对应要测试的位取 1，其他位都是 0。将屏蔽字节与状态寄存器的内容相与，若状态寄存器中被测试的位是 0，则结果为全 0；若状态寄存器中被测试的位是 1，则结果为非 0。

A/D 转换器进行 A/D 转换需要一定时间，必须等 A/D 转换完成之后才能读取转换结果，否则会读出错误的数据。例如，某 A/D 转换器的 A/D 转换完成位寄存器是 8 位的，I/O端口地址为 0x285，其最高位，即 D7 位是 A/D 转换完成位。A/D 转换未完成时 D7 = 0；A/D 转换完成后 D7 = 1。A/D 转换启动之后就应不断查询 A/D 转换完成位是否为 1，如果

是1，读转换后的结果；如果是0，继续查询。相关的程序语句如下：

```
while(!(inportb(0x285) & 0x80));
```

读A/D转换结果的语句；

注意：按位与的运算符与取地址运算符相同，但按位与的运算是二目运算，取地址是单目运算。

还要注意按位与运算跟逻辑与运算的区别。逻辑与运算的运算符为“&&”，逻辑运算的结果是0或者1（非0）。逻辑与运算是二元运算，它的功能是，当两个运算量都是非零值时，运算结果是1；只要有一个运算量是零值时，运算结果就是0。例如：

```
c = (a&&b);
```

当a和b都是非零值时，c的值为1。当a和b中有一个是零值或者都是零值时，运算结果c为0。

2.2.2 按位或运算的应用

按位或运算符是“|”，在进行按位或运算时，要求有两个运算量，例如a|b是将a和b对应的二进制位上的值进行按位或运算。对应的两个位上的值只要有一个是1时，结果位就是1；只有两个值都是0时，结果才是0。二进制对应位之间的运算规则，即a|b的真值表如表2-2所示。

表2-2 a|b的真值表

a 位	b 位	结 果 位
0	0	0
0	1	1
1	0	1
1	1	1

若 a=0x55，b=0xf0

则 c=a|b =0xf5

为了便于观察，将a|b写成二进制竖式形式

```
  01010101    (a=0x55)
| ) 11110000   (b=0xf0)
  11110101    (c=0xf5)
```

a的高4位跟1相或，所以结果都是1；a的低4位跟0相或，所以结果各位都保持不变。

按位或运算可以使某些位置1，而其他位保持不变。跟1相或的位被置1；而跟0相或的位保持不变。

例如，8255并行接口芯片的B口地址为0x61，这是一个双向的8位I/O端口。该端口的最后两位D1和D0控制扬声器发声，其他6位控制其他外设。只有D1和D0都是1时，扬声器才能发声。

欲使8255的B口的D1、D0位置1，其他位不变，相关程序语句如下：

```
#include <dos.h>
```

```
char b;
⋮
b = inportb(0x61);
b = b|0x03;
outportb(0x61,b);
```

还要注意按位或运算跟逻辑或运算的区别。逻辑或运算的运算符为“||”，逻辑或运算的结果是0或者1（非0）。逻辑或运算是二元运算，它的功能是，两个运算量中只要有一个是非零值或两个都是非零值时，运算结果是1；只有当两个运算量都是零值时，运算结果才是0。例如：

```
c = (a || b);
```

当a和b中有一个或两个是非零值时，c的值为1；当a和b都是零值时，c为0。

2.2.3 按位异或运算的应用

按位异或运算符是“^”，在进行按位异或运算时，同样要求有两个运算量，例如a^b是将a和b对应的二进制位进行按位异或运算。按位异或运算的作用是判断两个数据对应位上的值是否“相异”（不同），若相异结果为1，若相同结果为0。即所谓“相同为0，不同为1”。二进制对应位之间的运算规则，即a^b的真值表如表2-3所示。

表2-3 a^b的真值表

a 位	b 位	结 果 位
0	0	0
0	1	1
1	0	1
1	1	0

若 a = 0x55，b = 0xf0

则 c = a^b = 0xa5

写成二进制竖式形式

```
  01010101    (a = 0x55)
^) 11110000    (b = 0xf0)
  ----------
  10100101    (c = 0xa5)
```

a的高4位跟1相异或，结果的高4位与a相比都取反了，0变成1，1变成0，a的低4位跟0相异或，结果都保持不变。

按位异或运算可以使某些位取反，而其他位保持不变。跟1相异或的位取反，跟0相异或的位不变。

数字口某一位的状态从0到1，再从1到0不断变化，就可以产生脉冲信号。打印机接口数字口的地址是0x378，若要求其D6位取反，其他位不变，相关程序语句如下：

```
#include <dos.h>
char b;
⋮
```

```
b = inportb(0x378);
b = b^0x40;
outportb(0x378,b);
```

利用异或运算还可以对数据或文件进行加密和解密。数据 0x55 跟 0xf0 按位相异或之后变成了 0xa5，将这个结果再与 0xf0 按位相异或就又变成了 0x55，即数据复原了。对数据或文件加密时，可选择一个若干位的“密码”对欲加密的数据或文件进行按位异或运算，加密后的数据或文件就面目全非了。解密时再用这个密码进行按位异或运算，数据或文件就又恢复如初了。

2.2.4 按位非运算的应用

按位非运算符是“~”，按位非运算也叫做按位取反运算，按位非运算符“~”是个单目运算符，运算量写在运算符之后。它将运算量的每个二进制位都取反，即 1 变成 0，0 变成 1。

若　a = 0x55　对应二进制形式为　01010101

则　~a = 0xaa　对应二进制形式为　10101010

注意：按位非运算符“~”将运算量的每个二进制位都毫无例外地取反，而按异或运算符“^”却可以使某些位取反其他位不变，因而更具有选择性。

还要注意按位非运算跟逻辑非（或称逻辑反）运算的区别。逻辑非运算的运算符为“!”，逻辑运算的结果是 0 或者 1（非 0），逻辑非运算是一元运算，它的功能是，当运算量是非零值时，运算结果是零；反之，当运算量是零值时，运算结果是 1。例如：

```
!(a%b)
```

在上述表达式中，若 a 能被 b 整除，则（a%b）的结果为 0，而其逻辑反为 1，所以上述运算表达式的结果为 1。如果 a 不能被 b 整除，则（a%b）的结果不为 0，而其逻辑反为 0，所以上述运算表达式的结果为 0。

2.2.5 按位左移运算和按位右移运算的应用

按位左移运算符“≪”和按位右移运算符“≫”分别用于将变量的每一位向左移动和向右移动。

左移运算的一般表达式是：

被移位变量名≪左移的位数；

左移后右端空出的位补 0。左移的位数一般情况下不超过被移位变量的位数。左移一位相当于乘 2，当然这个结果是在没有溢出和进位的情况下才是对的。

右移运算的一般表达式是：

被移位变量名≫右移的位数；

无符号数右移后左端的空位补 0。有符号数左端最高位（MSB）是符号位，0 表示正数，1 表示负数。有符号数右移之后补充符号位，即正数补 0，负数补 1。这样，右移之后仍保持符号不变。右移的位数一般情况下不超过被移位变量的位数。右移 1 位相当于除以 2 之后的商。

左移和右移的运算速度要比乘法和除法的运算速度快得多。因此，在要求运算速度比较快的实时测量和控制程序中，经常使用左移和右移代替乘以和除以 2 的整数次幂的运算。

例 2-1　计算计算机字长，即一个整型数所占的二进制数位数。

该程序首先对无符号整型变量 word 赋 0 值，word 的各个二进制位都是 0，然后对 word 取反，这样 word 的各个二进制位都是 1。最后测试 word 不为 0 的二进制数的位数，得到该计算机 CPU 的字长。这个程序中用到了按位取反和移位运算。

```
#include "stdio. h"
#include "dos. h"
main( )
{
    unsigned word,n;
    word = 0;
    word = ~word;   /* word 变为全 1 */
    n = 1;
    while( ( word = word≫1) ! =0)
        n + + ;
    printf("word length = % d\n",n);
    getch( );
}
```

例 2-2 循环移位的实现。

若要实现循环移位，即移出位的数补充到移位后的空位，可编写一段小程序。下面的程序中，j 是被移位的无符号字符型变量，初值是 0x58，n = 4 是循环移位的次数，先循环右移 4 次，再循环左移 4 次。

```
#include "stdio. h"
#include "conio. h"
main( )
{   unsigned char i,j = 0x58,n = 4;
    printf("0x% x",j);                  /* 用十六进制数的形式显示要移位的变量 */
      for(i = 1;i < = n;i + + )         /* n 是循环右移的次数 */
  {
    if(j&0x01) j = ((j≫1) |0x80);   /* 若移出的是 1,左端空位补充 1 */
    else j = j≫1;
    printf("0x% x ",j);}                /* 用十六进制数的形式显示移位后的数 */
    printf("\n");
    printf("0x% x",j);                  /* 用十六进制数的形式显示要移位的变量 */
    for(i = 1;i < = n;i + + )           /* n 是循环左移的次数 */
    {
    if(j&0x80) j = ((j≪1) |0x01);   /* 若移出的是 1,右端空位补充 1 */
    else j = j≪1;
    printf("0x% x",j);}
  getch( );
}
```