第 1 章　单片机的 C 语言概述

1．写出一个单片机的 C 语言程序的构成。

```
#include <   >                    /*预处理命令*/
long  fun1( );                    /*函数说明*/
float  fun2( );
int  x，y;
float  z;
fun1( )                           /*功能函数 1*/
{   ...
}
main( )                           /*主函数*/
{   ...
}
fun2( )                           /*功能函数 2*/
{   ...
}
```

2．标准 C 语言程序主要的结构特点

标准 C 语言的主要结构特点有以下几点：

① 语言简洁、紧凑，使用方便、灵活

标准 C 语言共有 32 个关键字、9 种控制语句。程序书写形式自由，与其他高级语言相比较，程序简练、简短。

② 运算符、表达式丰富

标准 C 语言包括 34 种运算符，而且把括号、赋值、强制类型转换等都作为运算符处理。表达式灵活、多样，可以实现各种各样的运算。

③ 数据结构丰富，具有现代化语言的各种各样的数据结构

标准 C 语言的数据类型有整型、实型、字符型、数组类型、指针类型等，并能用来实现各种复杂的数据结构。

④ 可进行结构化程序设计

标准 C 语言具有各种结构化的程序语句，如 if...else 语句、while 语句、do...while 语句、switch 语句、for 语句等。

⑤ 可以直接对计算机硬件进行操作

标准 C 语言允许直接访问物理地址，能进行位操作，能实现汇编语言的大部分功能，可以对硬件直接进行操作。

⑥ 生成的目标代码质量高，程序执行效率高

众所周知，汇编语言生成的目标代码的效率是最高的。但据统计表明，对于同一个问题，用 C 语言编写的程序生成目标代码的效率仅比汇编语言编写的程序低 10%～20%。而 C 语言编写程序比汇编语言编写程序方便、容易得多，可读性强，开发时间也短得多。

⑦ 可移植性好

不同的计算机汇编指令不一样，用汇编语言编写的程序用于另外型号的机型使用时，必须改写成对应机型的指令代码。而标准 C 语言编写的程序基本上都不用修改就可以用于各种机型和各类操作系统。

4．C51 语言和汇编语言的比较

使用 C51 语言进行嵌入式系统的开发，有着汇编语言所不可比拟的优势：

① 编程调试灵活方便；

② 生成的代码编译效率高；

③ 模块化开发；

④ 可移植性好；

⑤ 便于项目的维护；

5．单片机的 C 语言和标准 C 的比较

答：单片机的 C 语言和标准 C 的比较主要有以下几点不同：

① C51 中定义的库函数和标准的 C 语言定义的库函数不同；

② C51 中的数据类型和标准 C 的数据类型也有一定的区别；

③ C51 变量的存储模式与标准 C 中变量的存储模式不一样；

④ C51 与标准 C 的输入/输出处理不一样；

⑤ C51 与标准 C 语言在函数使用方面有一定的区别。

6. 单片机的 C 语言的特点

单片机的 C 语言的特点主要体现在以下几个方面：

① 无需了解机器硬件及其指令系统，只需初步了解 MCS-51 的存储器结构；

② C51 能方便的管理内部寄存器的分配、不同存储器的寻址和数据类型等细节问题，但对硬件控制有限；而汇编语言可以完全控制硬件资源；

③ C51 在小应用程序中，产生的代码量大，执行速度慢；但在较大的程序中代码效率高；

④ C51 程序由若干函数组成，具有良好的模块化结构，便于改进和扩充；

⑤ C51 程序具有良好的可读性和可维护性；而汇编语言在大应用程序开发中，开发难度增加，可读性差；

⑥ C51 有丰富的库函数，可大大减少用户的编程量，显著缩短编程与调试时间，大大提高软件开发效率；

⑦ 使用汇编语言编制的程序，当机型改变时，无法直接移植使用，而 C 语言程序是面向用户的程序设计语言，能在不同机型的机器上运行，可移植性好。

7. 使用 KeilC51 开发工具开发软件的流程

使用Keil Software工具时，用户的项目开发流程和其它软件开发项目的流程极其相似，主要包括以下几个步骤：

① 创建一个项目，从器件库中选择目标器件并配置工具软件的设置；

② 用C语言或汇编语言创建源程序；

③ 用项目管理器生成用户的应用；

④ 修改源程序中的错误；

⑤ 调试链接后的应用。

一个完整的8051工具集的框图可以很好地表述此开发流程，如图1-1所示。

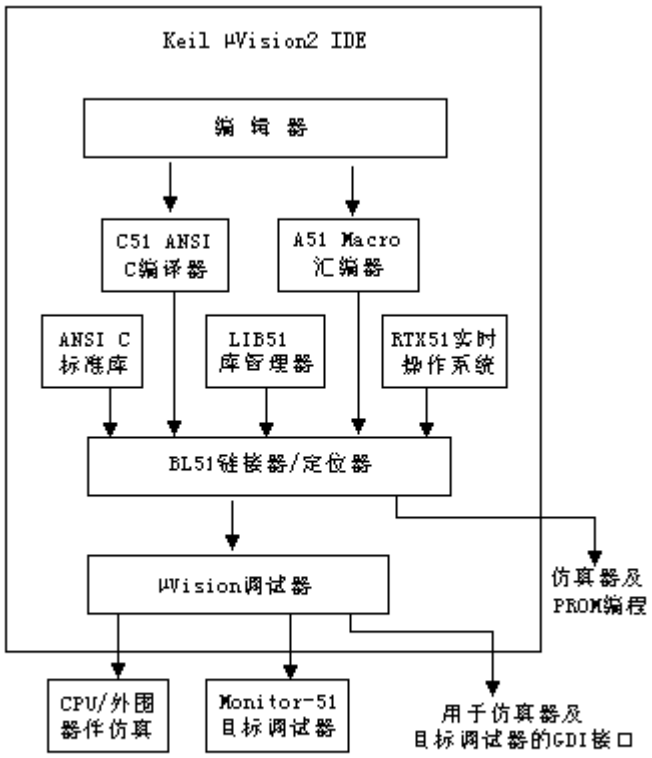

图1-1　使用μVision2开发软件流程

第 2 章　C51 语言程序设计基础

1．哪些变量类型是51单片机直接支持的？

C51编译器支持的数据类型有：位型（bit）、无符号字符型（unsigned char）、有符号字符型（signed char）、无符号整型（unsigned int）、有符号整型（signed int）、无符号长整型（unsigned long）、有符号长整型（signed long）、浮点型（float）和指针型等。

C51编译器支持的数据类型、长度和值域如表2-1所示。

表2-1　C51的数据类型

数据类型	长度/bit	长度/byte	值域
bit	1		0,1
unsigned char	8	1	0～255
signed char	8	1	-128～127
unsigned int	16	2	0～65 535
signed int	16	2	-32 768～32 767
unsigned long	32	4	0～4 294 967 295
signed long	32	4	-2 147 483 648～2 147483 647
float	32	4	±1. 176E-38～±3. 40E+38（6 位数字）
double	64	8	±1. 176E-38～±3. 40E+38（10 位数字）
一般指针	24	3	存储空间 0～65 535

2．C51的数据存储类型

答：我们都知道，8051单片机存储区可分为内部数据存储区、外部数据存储区以及程序存储区。8051单片机内部的数据存储区是可读写的，8051派生系列最多可有256字节的内部数据存储区，其中低128字节可直接寻址，高128字节（从0x80到0xFF）只能间接寻址，从20H开始的16字节可位寻址。内部数据区可分为3个不同的存储类型：data、idata和bdata。

外部数据区也是可读写的，访问外部数据区比访问内部数据区慢，因为外部数据区是通过数据指针加载地址来间接访问的。C51 提供两种不同的存储类型 xdata 和 pdata 访问外部数据。

程序存储区是只能读不能写。程序存储区可能在 8051 单片机内部或者在外部或者内外都有，这由 8051 单片机的硬件决定。C51 提供了 code 存储类型来访问程序存储区。

每个变量可以明确地分配到指定的存储空间，对内部数据存储器的访问比对外部数据存储器的访问快许多，因此应当将频繁使用的变量放在内部存储器中，而把较少使用的变量放在外部存储器中。各存储区的简单描述如表 2-2 所示。

表 2-2　C51 存储类型与 8051 存储空间的对应关系

存储区	描述
DATA	片内 RAM 的低 128 字节，可在一个周期内直接寻址
BDATA	片内 RAM 的位寻址区，16 字节
IDATA	片内 RAM 的 256 字节，必须采用间接寻址
XDATA	外部数据存储区，使用 DPTR 间接寻址
PDATA	外部存储区的 256 个字节，通过 P0 口的地址对其寻址。使用 MOVX @Ri，需要两个指令周期
CODE	程序存储区，使用 DPTR 寻址。

以上介绍的是 C51 的数据存储类型，C51 存储类型及其大小和值域如表 2-3 所示。

表2-3 C51存储类型及其大小和值域

存储类型	长度/bit	长度/byte	值域
data	8	1	0～255
idata	8	1	0～255
pdata	8	1	0～255
code	16	2	0～65 535
xdata	16	2	0～65 535

3．C51对51单片机特殊功能寄存器的定义方法

MCS-51通过其特殊功能寄存器（SFR）实现对其内部主要资源的控制。MCS-51单片机有21个SFR，有的单片机还有更多的SFR，它们分布在片内RAM的高128字节中，其地址能够被8整除的

SFR一般可以进行位寻址。关于MCS-51单片机的特殊功能寄存器参看附录A。对SFR只能用直接寻址方式访问。C51允许通过使用关键字sfr、sbit或直接引用编译器提供的头文件来实现对SFR的访问。

（1）使用关键字定义sfr

为了能直接访问特殊功能寄存器SFR，C51提供了一种自主形式的定义方法。这种定义方法与标准的C语言不兼容，只适用于对8051系列单片机进行C编程。这种定义的方法是引入关键字“sfr”，语法如下：

sfr 特殊功能寄存器名字 = 特殊功能寄存器地址；

如：

```
sfr  SCON=0x98;          /*串口控制寄存器地址98H*/
sfr  TMOD=0X89;          /*定时器/计数器方式控制寄存器地址89H*/
```

（2）通过头文件访问SFR

8051系列单片机的寄存器数量与类型是极不相同的，因此对单片机特殊功能寄存器的访问可以通过对头文件的访问来进行。

为了用户处理方便，C51编译器把MCS-51单片机的常用的特殊功能寄存器和特殊位进行了定义，放在一个“reg51.h”或“reg52.h”的头文件中。当用户要使用时，只需要在使用之前用一条预处理命令“#include <reg51.h>”把这个头文件包含到程序中，然后就可以使用特殊功能寄存器名和特殊位名称了。用户可以通过文本编辑器对头文件进行增减。

（3）**SFR中位定义**

在8051单片机的应用问题中，经常需要单独访问SFR中的位，C51的扩充功能使之成为可能，使用关键字“sbit”可以访问位寻址对象。特殊位（sbit）的定义，像SFR一样不与标准C兼容。

与SFR定义一样，用关键字“sbit”定义某些特殊位，并接受任何符号名，“=”号后将绝对地址赋给变量名。这种地址分配有三种方法：

第一种方法：

sbit 位名=特殊功能寄存器名^位置；

当特殊功能寄存器的地址为字节（8位）时，可使用这种方法。特殊功能寄存器名必须是已定义的SFR的名字。“^”后的“位置”语句定义了基地址上的特殊位的位置。该位置必须是0～7的数。如：

第二种方法：

sbit　位名=字节地址^位置；

这种方法是以一个整常数为基地址，该值必须在0x80～0xFF之间，并能被8整除，确定位置的方法同上。

第三种方法：

sbit　位名=位地址；

这种方法将位的绝对地址赋给变量，地址必须在0x80～0xFF之间。

4．C51对51单片机片内I/O口和外部扩展的I/O口的定义方法

C51对51单片机片内I/O口的定义方法是将片内I/O口看成SFR。

C51对51单片机片外I/O的访问有两种比较常用的访问方法：

（1）绝对宏

C51编译器提供了一组宏定义来对51系列单片机的code、data、pdata和xdata空间进行绝对寻址。在程序中，用“#include<absacc.h>”即可使用其中声明的宏来访问绝对地址，包括CBYTE、XBYTE、PWORD、DBYTE、CWORD、XWORD、PBYTE、DWORD，具体使用方法参考absacc.h头文件。其中：

CBYTE以字节形式对code区寻址；CWORD以字形式对code区寻址；

DBYTE以字节形式对data区寻址；DWORD以字形式对data区寻址；

XBYTE以字节形式对xdata区寻址；XWORD以字形式对xdata区寻址；

PBYTE以字节形式对pdata区寻址；PWORD以字形式对pdata区寻址；

（2）_at_关键字

可以使用关键字_at_对指定的存储器空间的绝对地址进行访问，一般格式如下：

[存储器类型]　数据类型说明符　变量名　_at_地址常数；

其中，存储器类型为C51能识别的数据类型，如省略则按存储器模式规定的默认存储器类型确定变量的存储器区域；数据类型为C51支持的数据类型；地址常数用于指定变量的绝对地址，必须位于有效的存储器空间之内；使用_at_定义的变量必须为全局变量。

5．C51对51单片机位变量的定义方法

答：除了通常的C数据类型外，C51编译器支持bit数据类型。

采用关键字“bit”进行定义。如：

```
bit  direction_bit;          /* 将direction_bit定义为位变量 */
bit  lock_pointer;           /* 将lock_pointer定义为位变量 */
bit  display_invers;         /* 将display_invers定义为位变量 */
```

6．C51 和Turbo C 的数据类型和存储类型有哪些异同点？

C51增加了位变量，取消了布尔变量。

7．C51 的data、bdata、idata 有什么区别？

data、bdata、idata是表明数据的存储类型，

data是指片内RAM的低128字节，可在一个周期内直接寻址；

bdata是指片内RAM的位寻址区，16字节；

idata是指片内RAM的256字节，必须采用间接寻址。

8．C51中的中断函数和一般的函数有什么不同？

C51编译器允许用C51创建中断服务函数，中断函数是由中断系统自动调用的。

中断函数的定义格式为：

函数类型　函数名　interrupt　n　using n

其中：

interrupt和using为关键字；

interrupt后面的n 为中断源的编号，即中断号；

using后面的n所选择的寄存器组，取值范围为0～3。

定义中断函数时，using是一个选项，可以省略不用。如果不用using选项，则由编译器选择一个寄存器组作为绝对寄存器组。

8051的中断过程通过使用interrupt关键字和中断号（0～31）来实现，中断号告诉编译器中断函数的入口地址。

9．C51采用什么形式对绝对地址进行访问？

答：绝对地址的访问包括片内RAM、片外RAM及I/O的访问。C51提供了两种比较常用的访问绝

对地址的方法。

（1）绝对宏

C51编译器提供了一组宏定义来对51系列单片机的code、data、pdata和xdata空间进行绝对寻址。在程序中，用"#include<absacc.h>"即可使用其中声明的宏来访问绝对地址，包括CBYTE、XBYTE、PWORD、DBYTE、CWORD、XWORD、PBYTE、DWORD，具体使用方法参考absacc.h头文件。其中：

CBYTE以字节形式对code区寻址；CWORD以字形式对code区寻址；

DBYTE以字节形式对data区寻址；DWORD以字形式对data区寻址；

XBYTE以字节形式对xdata区寻址；XWORD以字形式对xdata区寻址；

PBYTE以字节形式对pdata区寻址；PWORD以字形式对pdata区寻址；

如：

```
#include<absacc.h>
#define PORTA XBYTE[0xFFC0]/*将PORT定义为外部I/O口，地址为0xFFC0，长度为8位*/
#define NRAM DBYTE[0x40]   /*将NRAM定义为片内RAM，地址为40H，长度为8位*/
```

（2）_at_关键字

可以使用关键字_at_对指定的存储器空间的绝对地址进行访问，一般格式如下：

[存储器类型] 数据类型说明符 变量名 _at_地址常数;

其中，存储器类型为C51能识别的数据类型，如省略则按存储器模式规定的默认存储器类型确定变量的存储器区域；数据类型为C51支持的数据类型；地址常数用于指定变量的绝对地址，必须位于有效的存储器空间之内；使用_at_定义的变量必须为全局变量。

10．按照给定的数据类型和存储类型，写出下列变量的说明形式

（1）在data区定义字符变量val1

```
char  data  vall;
```

（2）在idata区定义整型变量val2

```
int  idata  val2;
```

（3）在xdata区定义无符号字符型数组val3[4]。

```
unsigned  xdata  val3[4];
```

（4）在xdata区定义一个指向char类型的指针px。

```
char  xdata  *px;
```

（5）定义可位寻址变量flag。

```
bit  flag;
```

（6）定义特殊功能寄存器变量P3。

```
Sfr  P3=0xB0;
```

11．break和continue语句的区别是什么？

答：break语句用于从循环代码中退出，然后执行循环语句之后的语句，不再进入循环。

Continue语句用于退出当前循环，不再执行本轮循环，程序代码从下一轮循环开始执行，直到判断条件不满足为止。

和break的区别是该语句不是退出整个循环。

12．C语言的基本运算、数组、指针、函数、流程控制语句

第 3 章　单片机内部资源的 C51 编程

1．在 8051 系统中，已知振荡频率是 12MHz，用定时器/计数器 T0 实现从 P1.1 产生周期是 2s 的方波，试编程。

```
#include <reg51.h>
sbit P1_1=P1^1;              //定义位变量
unsigned char i;             //定时次数
void timer0over(void);       //函数声明（定时 50ms 函数）
void main( )                 //主函数
{
    i=0;
    TMOD=0x01;                    //定时器 T0 定时 50ms，方式 1
    TH0=(65536-50000)/256;
    TL0=(65536-50000)%256;
    TR0=1;                        //开 T0
    for(;;)
    {
        if(TF0)
        timer0over( );
    }
}

void timer0over(void)
{
    TH0=(65536-50000)/256;
    TL0=(65536-50000)%256;
    TF0=0;
```

```
    i++;
    if(i==20)                    //20 次到了吗?
    {
        i=0;
        P1_1=!P1_1;
    }
}
```

2．在 8051 系统中，已知振荡频率是 12MHz，用定时器/计数器 T1 实现从 P1.1 产生高电平宽度是 10ms，低电平宽度是 20ms 的矩形波，试编程。

```
#include <reg51.h>
unsigned char i;
sbit  P1_1=P1^1;

void main( )                //主函数
{
    i=0;
    TMOD=0x10;                  //T1 定时方式 1，定时时间 10ms
    TH0=(65536-10000)/256;
    TL0=(65536-10000)%256;
    EA=1;
    ET1=1;
    TR1=1;
    while(1);
}

void timer1_int(void) interrupt 3
{
```

```
    TH0=(65536-10000)/256;
    TL0=(65536-10000)%256;
    i++;
    if(i==1) P1_1=0;
    else if(i==3)
   {
       i=0;
       P1_1=1;
   }
}
```

3．用 8051 单片机的串行口扩展并行 I/O 口，控制 16 个发光二极管依次发光，试编程。

```
#include <reg51.h>
sbit P1_0=P1^0;            //定义位变量
sbit P1_1=P1^1;

void main()                //主函数
{
    unsigned char i,j;
    bit flag=1;
    SCON=0x00;
    j=0x01;
    for(;;)
    {
        P1_0=0;
        P1_1=0;
        while(flag)
```

```
      {
          P1_0=0;
          SBUF=j;
          while(!TI){;}
          P1_0=1;TI=0;
          for(i=0;i++;i<=254){;}
          j=j*2;
          if(j==0x00)
      {
      j=0x01;
      flag=0;
      }
   }
  while(!flag)
  {
      P1_1=0;
      SBUF=j;
      while(!TI){;}
      P1_1=1;TI=0;
      for(i=0;i++;i<=254){;}
      j=j*2;
      if(j==0x00)
    {
    j=0x01;
    flag=1;
    }
  }
}
```

```
}
```

4．用 8751 单片机制作一个模拟航标灯，灯接在 P1.7 上，$\overline{\text{INT0}}$ 接光敏器件，使它具有如下功能：

（1）白天航标灯熄灭；夜间间歇发光，亮 2s，灭 2s ，周而复始。

（2）将 $\overline{\text{INT0}}$ 信号作为门控信号，启动定时器定时。

按以上要求编写控制主程序和中断服务程序。

```
#include <reg51.h>
unsigned char i;
sbit  P1_1=P1^1;

void main()
{
    i=0;
    TMOD=0x09;
    TH0=(65536-50000)/256;
    TL0=(65536-50000)%256;
    EA=1;
    ET0=1;
    TR0=1;
    while(1)
    {
        if(!P3^2)                    //int0=0,白天
        P1|=0x80;                    //P1.7 置 1,灯灭
    }
}
//int0=1 时启动定时器 0
void timer0_int(void) interrupt 1
```

```
{
    TH0=(65536-50000)/256;
    TL0=(65536-50000)%256;
    i++;
    if(i==40)
   {
        i=0;
        P1_1=!P1_1;
   }
}
```

5．外部 RAM 以 DATA1 开始的数据区中有 100 个数据，现在要求每隔 150ms 向内部 RAM 以 DATA2 开始的数据区传送 10 个数据，通过 10 次传送把数据全部传送完，以定时器 1 作为定时，编写有关程序。单片机的时钟频率是 6MHz。

```
#include <reg51.h>
unsigned char i,j,k;
unsigned char xdata data1[100];
unsigned char data data2[100];
void main()
{
    i=0;
    j=0;
    TMOD=0x10;
    TH1=(65536-25000)/256;
    TL1=(65536-25000)%256;
    EA=1;
    ET1=1;
```

```
    TR1=1;
    while(j<10);
    EA=0;
    while(1);
}

void timer1_int(void) interrupt 3
{
    TH1=(65536-25000)/256;
    TL1=(65536-25000)%256;
    i++;
    if(i==3)                        //150ms 时间到，传送一个数据
   {
        i=0;
        for(k=0;k++;k<10)
        data2[j*10+k]=data1[j*10+k];
   }
    j++;
}
```

6．用单片机和内部定时器来产生矩形波，要求频率为 100Hz，占空比为 2：1，设单片机的时钟频率为 12MHz，写出有关程序。

```
#include <reg51.h>
unsigned char i;
sbit  P1_1=P1^1;

void main()
{
```

```
    i=0;
    TMOD=0x10;
    TH0=(65536-10000)/256;
    TL0=(65536-10000)%256;
    EA=1;
    ET1=1;
    TR1=1;
    while(1);
}

void timer1_int(void) interrupt 3
{
    TH0=(65536-10000)/256;
    TL0=(65536-10000)%256;
    i++;
    if(i==2) P1_1=0;
    else if(i==3)
   {
    i=0;
    P1_1=1;
   }
}
```

第 4 章　单片机外部资源的 C51 编程

1．某单片机系统应用 8255 开展 I/O 口，设其 A 口为方式 1 输入，B 口为方式 1 输出，C 口余下的引脚用于输出，试写出其初始化程序。

```
#include <reg51.h>
#include <absacc.h>
#define COM8255 XBYTE[0xe003]  /*定义 8255 控制寄存器地址*/
void init8255(void)
{
 COM8255=0xb4;    /*定义 8255 控制寄存器地址*/
}
```

2．编写出 8×4 矩阵键盘的 C51 程序。

```
#include <reg51.h>
#include <absacc.h>
#define H_PORT XBYTE[0xport1]
#define L_PORT XBYTE[0xport2]
#define ulint unsigned long int
#define uint unsigned int
#define uchar unsigned char
ulint Exp(uint m,uint x); /*定义一个指数函数*/

void keyscan(void)
{
 uchar h_code,l_code,key_code;
 uint i,x,m;
 H_PORT=0x00;          /*所有行输出 0*/
 l_code=L_PORT;
```

```
l_code=l_code&0x0f;
if(l_code!=0xef)
{
 h_code=0xfe;          /*逐行输出 0*/
 while(h_code!=0xff)
 {
  H_PORT=h_code;
  l_code=L_PORT;
  l_code=l_code&0x0f;
  if(l_code==0x0f)
  {
    h_code=h_code<<1;
    h_code=h_code|0x01;
  }
 }
 h_code^=0xff;           /*计算键值,方法是先把行值与 0xff 相异或，使对应的 0 的
那一位变为 1*/
 for(i=0;i<=x;i++)       /*  然后把行值与 2 的指数幂作比较，若想等，则把其对应
的幂值赋给行值*/
 {                       /*最后，把得到的行值放在高四位，与列值相或可得到键值*/
  if(h_code==Exp(2,x))
  {
   h_code=x;
   break;
  }
 }
 h_code=h_code<<4;
 key_code=h_code|l_code;
```

```
  switch(key_code)
  {
   case 0x0e:    goto  k0;
   case 0x1e:    goto  k1
   ……
   case 0x67:    goto  k14
   case 0x77:    goto  k15
   default:break;
  }
 }
}

ulint Exp(uint m,uint x)   /*指数函数*/
{uint i;
 ulint temp=1;
 for(i=0;i<=x;i++)
 {
  temp*=m;
 }
 return temp;
}
```

3．编写出 8 位 LED 共阴极显示器的动态显示的 C51 程序。

```
#include <reg51.h>
#include <absacc.h>
#define uchar unsigned char
#define COM8255 XBYTE[0xbfff]
#define PA8255 XBYTE[0xbcff]
```

```
#define PB8255 XBYTE[0xbdff]
#define PC8255 XBYTE[0xbeff]
uchar idata dis_buf[8]={1,2,3,4,5,6,7,8}; /*显示缓冲区*/
uchar code table[18]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,
                      0x7f,0x6f,0x77,0x7c,0x39,0x5e,0x79,0x71,
                      0x40,0x00};

void dl_ms(uchar d);
void display(void)
{ uchar esgcode,bitcode,i;
  bitcode=0xfe;
  for(i=0;i<=7;i++)
  { esgcode=dis_buf[i];
    PB8255=table[esgcode];
    PA8255=bitcode;
    dl_ms(1);
    bitcode=bitcode<<1;
    bitcode=bitcode|0x01;
  }
}

void main(void)
{ COM8255=0x80;
  while(1)
  { display();
  }
}
```

4．与 8051 单片机接口的 8255 的 4 个端口地址分别为：0DFFCH、0DFFDH、0DFFEH、0DFFFH。对 8255 编程：口 A 输出数据 0AAH，口 B 输入 10 个数据到片内 RAM 区，由 PC4 位产生一个负脉冲，低电平宽度为 10μs。

```
#include <reg51.h>
#include <absacc.h>
#define uchar unsigned char
#define COM8255 XBYTE[0xdfff]
#define PA8255 XBYTE[0xdffc]
#define PB8255 XBYTE[0xdffd]
#define PC8255 XBYTE[0xdffe]
data uchar Buffer[10] _at_0x30;

void main(void)
{ uchar index;
  int j;
  COM8255=0x82;
  PA8255=0XAA;
  for(index=0;index<10;index++)
  { Buffer[index]=PB8255;
  }
  PC8255=0xef;
  for(j=0;j<1;j++);
}
```

5．设某个生产过程有 6 道工序，每道工序的时间分别为 10s、8s、12s、15s、9s 和 6s。设延迟程序 DYLA 的延时为 1s。用单片机通过 8255 的口 A 来进行控制。口 A 中的每一位可以控制某一位的起停，试编写有关程序。

```
#include <reg51.h>
#include <absacc.h>
#define uint unsigned int
#define COM8255 XBYTE[0xdfff]
#define PA8255 XBYTE[0xdffc]
#define PB8255 XBYTE[0xdffd]
#define PC8255 XBYTE[0xdffe]

void DYLA(uint i);
void main(void)
{ COM8255=0x80;
  while(1)
  { PA8255=0x3f;  /*PA 口的第 1 到第 6 位分别控制 6S,8S,9S,10S,12S,15S 这 6 道
工具的启停*/
    DYLA(6);
    PA8255=0x3e;
    DYLA(2);
    PA8255=0x3c;
    DYLA(1);
    PA8255=0x3a;
    DYLA(1);
    PA8255=0x30;
    DYLA(2);
    PA8255=0x20;
    DYLA(3);
    PA8255=0x00;
   }
}
```

6．编程实现由 DAC0832 输出的幅度和频率都可以控制的三角波，即从 0 上升到最大值，再从最大值下降到 0，并不断重复。

```
/*产生幅值为 125，频率为 2Hz 的三角波*/
#include <reg51.h>
#define uchar unsigned char
#define CYCLE 2000          //此波形的得到是通过定时 2ms 中断，在中断中当数据加到 125 个数时，再递减到 0.如此循环。可得三角波。其 T=125*2*2ms=500ms.所以 f=2Hz.
uchar xdata DA_data;  /*定义变量指向 0832 的数据输出地址*/

void main(void)
{ TMOD=0x10;  /*定时器 1 工作于方式 1*/
  TH1=-CYCLE/256;
  TL1=-CYCLE%256;
  TR1=1;
  IE=0x88;
  while(1);
}

void timer1(void) interrupt 2 using 1
{ static uchar s_Counter;
  static bit flag;          //定义递减标志位
  if(flag==0)
  { if(s_Counter++>=124)   //当增到 125 时开始递减
    { flag=1;
    }
  }
  else
```

```
  { if(s_Counter--<=0)      //当减到 0 时开始递增
    { flag=0;
    }
  }
  DA_data=s_Counter;         //输出需要转换的数据
  TH1=-CYCLE/256;
  TL1=-CYCLE%256;
}
```