单片机原理与应用

第一章 单片机概述

1.1 除了单片机这一名称之外，单片机还可称为（微控制器）和（嵌入式控制器）。

1.2 单片机与普通计算机的不同之处在于其将（微处理器）、（存储器）和（各种输入输出接口）三部分集成于一块芯片上。

1.3单片机根据其基本操作处理的位数可分为哪几种类型？

答：单片机根据其基本操作处理的位数可分为：1位单片机、4位单片机、8位单片机、16位单片机和32位单片机。

1.5 MCS-51系列单片机的基本芯片分别为哪几种？它们的差别是什么？

答：基本芯片为8031、8051、8751。

8031内部包括1个8位cpu、128BRAM，21个特殊功能寄存器（SFR）、4个8位并行I/O口、1个全双工串行口，2个16位定时器/计数器，但片内无程序存储器，需外扩EPROM芯片。

8051是在8031的基础上，片内又集成有4KBROM，作为程序存储器，是1个程序不超过4KB的小系统。

8751是在8031的基础上，增加了4KB的EPROM，它构成了1个程序小于4KB的小系统。用户可以将程序固化在EPROM中，可以反复修改程序。

第二章 MCS-51单片机的硬件结构

2.1 MCS-51单片机的片内都集成了哪些功能部件？各个功能部件的最主要的功能是什么？

答：功能部件如下：微处理器（CPU）；数据存储器（RAM）；程序存储器（ROM/EPROM，8031没有此部件），4个8位并行I/O口（P0口、P1口、P2口、P3口）；1个全双工的串行口；2个16位定时器/计数器；中断系统；21个特殊功能寄存器（SFR）。

各部件功能：CPU（微处理器）包括了运算器和控制器两大部分，还增加了面向控制的处理功能，不仅可处理字节数据，还可以进行位变量的处理；数据存储器（RAM）片内为128B（52系列的为256B），片外最多可外扩64KB。数据存储器来存储单片机运行期间的工作变量、运算的中间结果、数据暂存和缓冲、标志位等；程序存储器（ROM/EPROM）用来存储程序；中断系统具有5个中断源，2级中断优先权；定时器/计数器用作精确的定时，或对外部事件进行计数；串行口可用来进行串行通信，扩展并行I/O口，还可以与多个单片机相连构成多机系统，从而使单片机的功能更强且应用更广；特殊功能寄存器用于CPU对片内各功能部件进行管理、控制、监视。

2.2 说明MCS-51单片机的引脚 EA的作用，该引脚接高电平和接低电平时各有何种功能？

答：当该引脚为高电平时，单片机访问片内程序存储器，但在PC（程序计数器）值超过0FFFH（对于8051、8751）时，即超出片内程序存储器的4KB地址范围时，将自动转向执行外部程序存储器内的程序。

当该引脚为低电平时，单片机则只访问外部程序存储器，不论是否有内部程序存储器。对于8031来说，因其无内部程序存储器，所以该引脚必须接地，这样只能选择外部程序存储器。

2.3 MCS-51的时钟振荡周期和机器周期之间有何关系？

答：每12个时钟周期为1个机器周期。

2.4在MCS-51 单片机中，如果采用6 MHZ 晶振，1个机器周期为（2微秒）。

2.5程序存储器的空间里，有5个单元是特殊的，这5个单元对应MCS-51单片机5个中断源的中断入口地址，请写出这些单元的地址以及对应的中断源。

答： 中断源 入口地址

外部中断0 0003H

定时器0（T0） 000BH

外部中断1 0013H

定时器1（T1） 001BH

串行口 0023H

2.6 内部RAM中，位地址为30H的位，该位所在字节的字节地址为（26H）。

2.7 若A中的内容为63H，那么，P标志位的值为（0）。

2.8 判断下列说法是否正确：

（A）8031的CPU是由RAM和EPROM所组成。 （错）

（B）区分片外程序存储器和片外数据存储器的最可靠的方法是看其位于地址范围的低端还是高端。（错）

（C）在MCS-51中，为使准双向的I/O口工作在输入方式，必须保证它被事先预置为1。（对）

（D）PC可以看成是程序存储器的地址指针。 （对）

2.9 8031单片机复位后，R4所对应的存储单元的地址为（04H），因上电时PSW=（00H）。这时当前的工作寄存器区是（0）组工作寄存器区。

2.10 什么是机器周期？1个机器周期的时序是如何来划分的？如果采用12MHZ晶振，1个机器周期为多长时间？

答：CPU完成一个基本操作所需要的时间称为机器周期。时序划分：一个机器周期包括12个时钟周期，分为6个状态；S1-S6。每个状态又分为2拍；P1和P2。因此，1个机器周期中的12个时钟周期表示为：S1P1、S1P2、S2P1、S2P2、…、S6P2。如果采用12MHZ晶振，1个机器周期为1μs。

2.11判断以下有关PC和DPTR的结论是否正确？

（A）DPTR是可以访问的，而PC不能访问。 （错）

（B）它们都是16位的寄存器。 （对）

（C）它们都具有加1的功能。 （对）

（D）DPTR可以分为2个8位寄存器使用，但PC不能。 （对）

2.12、内部RAM中，哪些单元可作为工作寄存器区，哪些单元可以进行位寻址？写出它们的字节地址。

答：地址为00H-1FH的32个单元是4组通用工作寄存器区，每个区包括8个8位工作寄存器，编号为R0-R7。字节地址为20H-2FH的16个单元可进行128位的位寻址，这些单元构成了1位处理机的存储器空间。位地址范围是00H-7FH。

2.13使用8031单片机时，需将EA引脚接（低）电平，因为其片内无（程序）存储器。

2.14片内RAM低128个单元划分为哪3个主要部分？各部分的主要功能是什么？

答：字节地址为00H-1FH的32个单元是4组通用工作寄存器区，每个区包括8个8位工作寄存器，编号为R0-R7。可以通过改变PSW中的RS1、RS0来切换当前的工作寄存器区，这种功能给软件设计带来极大的方便，特别是在中断嵌套时，为实现工作寄存器现场内容保护提供了方便；字节地址为20H-2FH的16个单元可进行工128位的位寻址，这些单元构成了1位处理机的存储器空间；字节地址为30H-7FH的单元为用户RAM区，只能进行字节寻址。用于作为数据缓冲区以及堆栈区。

2.15 判断下列说法是否正确

（A）程序计数器PC不能为用户编程时直接使用，因为它没有地址。 （对）

（B）内部RAM的位寻址区，只能供位寻址使用，而不能供字节寻址使用。 （错）

（C）8031共有21个特殊功能寄存器，它们的位都是可用软件设置的，因此，是可以进行位寻址的。 （错）

2.16 PC的值是：（C）

（A）当前正在执行指令的前一条指令的地址

（B）当前正在执行指令的地址

（C）当前正在执行指令的下一条指令的地址

（D）控制器中指令寄存器的地址

2.18 写出P3口各引脚的第二功能。

答： 口引脚 第二功能定义

P3.0 串行输入口

P3.1 串行输出口

P3.2 外部中断0

P3.3 外部中断1

P3.4 定时器0外部计数输入

P3.5 定时器1外部计数输入

P3.6 外部数据存储器写选通

P3.7 外部数据存储器读选通

2.19 MCS-51单片机程序存储器的寻址范围是由程序计数器PC的位数所决定的，因为MCS-51的PC是16位的，因此其寻址的范围为（64）KB。

2.20 当MCS----51单片机运行出错或程序陷入死循环时，如何来摆脱困境？

答：可通过复位来解决。

2.21 判断下列说法是否正确？

（A）PC是1个不可寻址的特殊功能寄存器 （对）

（B）单片机的主频越高，其运算速度越快 （对）

（C）在MCS----51单片机中，1个机器周期等于1微秒 （错）

（D）特殊功能寄存器SP内装的是栈顶首地址单元的内容（错）

2.22 如果手中仅有一台示波器，可通过观察哪个引脚的状态，来大致判断MCS----51单片机正在工作？

答：ALE 引脚。

2.23 MCS-51 单片机内部有几个定时/计数器？它们由哪些寄存器组成？

答：MCS-51单片机内部有两个16位可编程的定时/计数器，简称定时器0（T0）和定时器1（T1）。它们分别由方式寄存器TMOD、控制寄存器TCON和数据寄存器TH0、TL0，TH1、TL1组成。

第五章 MCS-51的中断系统

5.1 什么是中断系统？

答：能够实现中断处理功能的部件称为中断系统。

5.2 什么是中断源？MCS-51有哪些中断源？各有什么特点？

答：产生中断的请求源称为中断源。MCS----51中断系统共有5个中断请求源：（1）外部中断请求0，中断请求标志为IE0。（2）外部中断请求1，中断请求标志为IE1。（3）定时器/计数器T0溢出中断请求，中断请求标志为TF0。（4）定时器/计数器T1溢出中断请求，中断请求标志为TF1。（5）串行口中断请求，中断请求标志为TI或RI。特点：2个外部中断源，

3个内部中断源。

5.3外部中断1所对应的中断入口地址为（0013H）。

5.4下列说法错误的是：（A，B，C）

（A）各中断源发出的中断请求信号，都会标记在MCS-51系统的IE寄存器中。

（B）各中断源发出的中断请求信号，都会标记在MCS-51系统的TMOD寄存器中。

（C）各中断源发出的中断请求信号，都会标记在MCS-51系统的IP寄存器中。

（D）各中断源发出的中断请求信号，都会标记在MCS-51系统的TCON和SCON寄存器中。

5.9 在MCS-51中，需要外加电路实现中断撤除的是（D）

（A）定时中断

（B）脉冲方式的外部中断

（C）外部串行中断

（D）电平方式的外部中断

5.11 下列说法正确的是（C D）

（A）同一级别的中断请求按时间的先后顺序顺序响应。

（B）同一时间同一级别的多中断请求，将形成阻塞，系统无法响应。

（C）低优先级中断请求不能中断高优先级中断请求，但是高优先级中断请求能中断低优先级中断请求

（D）同级中断不能嵌套。

5.12 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。

5.13. 如何消除键盘的抖动？

答：对于键抖动最方便的解决方法就是当发现有键按下后，不是立即进行扫描，而是延时大约10ms后再进行。由于一个键按下的时间 一般会持续上百毫秒，所以延迟10ms后再扫描处理，便可以消除键盘的抖动。

第六章 MCS-51的定时器/计数器

6.1 如果采用的晶振的频率为3MHZ，定时器/计数器工作在方式1下，其最大的定时时间各为多少？

答：方式1 216*4us=262.144

6.2 定时器/计数器用作定时器时，其计数脉冲由谁提供？定时时间与哪些因素有关？

答：定时器/计数器被选定为定时器工作模式时，计数输入信号是内部时钟脉冲，每个机器周期产生1个脉冲使计数器增1，因此，定时器/计数器的输入脉冲的周期与机器周期一样，为时钟振荡频率的1/2。

6.5 定时器/计数器的工作方式2有什么特点？适用于哪些应用场合？

答：工作方式2为自动恢复初值的（初值自动装入）8位定时器/计数器，TLX作为常数缓冲器，当TLX计数溢出时，在置1溢出标志TFX的同时，还自动的将THX中的初值送至TLX，使TLX从初值开始重新计数（X=0，1）。

6.6已知单片机系统晶振频率为6MHz，若要求定时值为10ms时，定时器T0工作在方式1

时，定时器T0对应的初值是多少？TMOD的值是多少？TH0=？TL0=？

答：定时值为10ms时，定时器T0工作在方式1时，定时器T0对应的初值是1388H

TMOD的值是00000001B，TH0=13H；TL0=88H。

6.12 判断下列说法是否正确？

（1）特殊功能寄存器SCON，与定时器/计数器的控制无关。 （√）

（2）特殊功能寄存器TCON，与定时器/计数器的控制无关 （×）

（3）特殊功能寄存器IE，与定时器/计数器的控制无关 （×）

（4）特殊功能寄存器TMOD，与定时器/计数器的控制无关 （×）

第七章 MCS-51的串行口

7.1 帧格式为1个起始位，8个数据位和1个停止位的异步串行通信方式是方式（1）。

7.2 串行口有几种工作方式？各种工作方式的波特率如何确定？

答：串行口有四种工作方式：方式0、方式1、方式2、方式3

有三种帧格式：

方式0波特率=Fosc/12

方式1波特率=2定时器T1的溢出率/32 SMOD

方式2的波特率=2*Fosc/64 MOD

方式3的波特率=2 *定时器T1的溢出率/32 MOD

7.4 判断下列说法是否正确

(A)串行口通行的第9数据位的功能可由用户定义。（T）

(B)发送数据的第9数据位的内容在SCON寄存器的TB8位中预先准备好的。（T）

(C)串行通讯帧发送时，指令把TB8位的状态送入发送SBUF中。（F）

（D）串行通讯接收到的第9位数据送SCON寄存器的RB8中保存。（T）

（E）串行口方式1的波特率是口变的，通过定时器/计数器T1的溢出率设定。（T）

7.6 为什么定时器/计数器T1用作串行口波特率发生器时，常采用方式2？若已知时钟频率，通讯波特率，如何计算器初值?

答：定时器T1工作方式2是一种自动重装方式，无需在中断服务程序中送数，没有由于中断引起的误差。定时器工作在方式2是一种既省事又精确的产生串行口波特率的方法。设定时器T1方式2的初值为X,则有：

定时器T1的溢出率=计数速率/(256-X)=Fosc/(256-X)*12

则方式2的波特率=2 *F/(256-X)*12*32 MODosc

故计数器初值为X=2 *F/12*32*波特率MODosc

7.7 串行口工作方式1的波特率是：C

（A）固定的，为Fosc/32。

（B）固定的，为Fosc/16。

（C）可变得，通过定时器/计数器T1的溢出率设定。

（D）固定的，为Fosc/64。

7.8 在串行通讯中，收发双方对波特率的设定应该是相同的。

7.9 若晶体振荡器为11.0592MHZ,串行口工作于方式1，波特率为9600b/s，写出用T1作为波特率发生器的方式控制字和计数初值。

7.10 已知串口通信在串口方式1下，SMOD=0, 波特率为9600bps,系统晶振频率为11.0592MHz，采用查询式编程实现：单片机往串口发送字符串 I get A；

答：

```
#include<reg52.h>
```

```
#define uchar unsigned char
uchar code table[]=”I get A”;
void main()
{
    uchar i,dat;
    TMOD=0x20;//设置定时器 1 为工作方式 2
    TH1=0xfd;//装入初值
    TL1=0xfd;//装入初值
    TR1=1;//开定时器 1

    SM0=0;
    SM1=1;//串口工作方式为方式 1
        REN=1;//允许串口接收数据

    for(i=0;i<6;i++)
    {
        SBUF=table[i]; //发送字符
        while(!TI);//等待字符发送完毕
        TI=0;//发送完毕标志清零
    }
        SBUF='A'; //发送字符
        while(!TI);//等待字符发送完毕
        TI=0;//发送完毕标志清零
    while(1);
}
```

7.11 已知串口通信在串口方式1下，SMOD=0, 波特率为9600bps,系统晶振频率为11.0592MHz，采用查询式和printf函数编程实现：单片机往串口发送字符串i get A；

```
#include<reg52.h>
#define uchar unsigned char
void main()
{
    uchar i,dat;
    TMOD=0x20;//设置定时器 1 为工作方式 2
    TH1=0xfd;//装入初值
    TL1=0xfd;//装入初值
    TR1=1;//开定时器 1

    SM0=0;
    SM1=1;//串口工作方式为方式 1
        REN=1;//允许串口接收数据
    dat='A';
    TI=1;//发送完毕标志置 1
    printf("I get %c\n" ,dat);//发送字符
    while(!T1);//等待字符发送完毕
```

```
    TI=0;//发送完毕标志清零

    while(1);
}
```

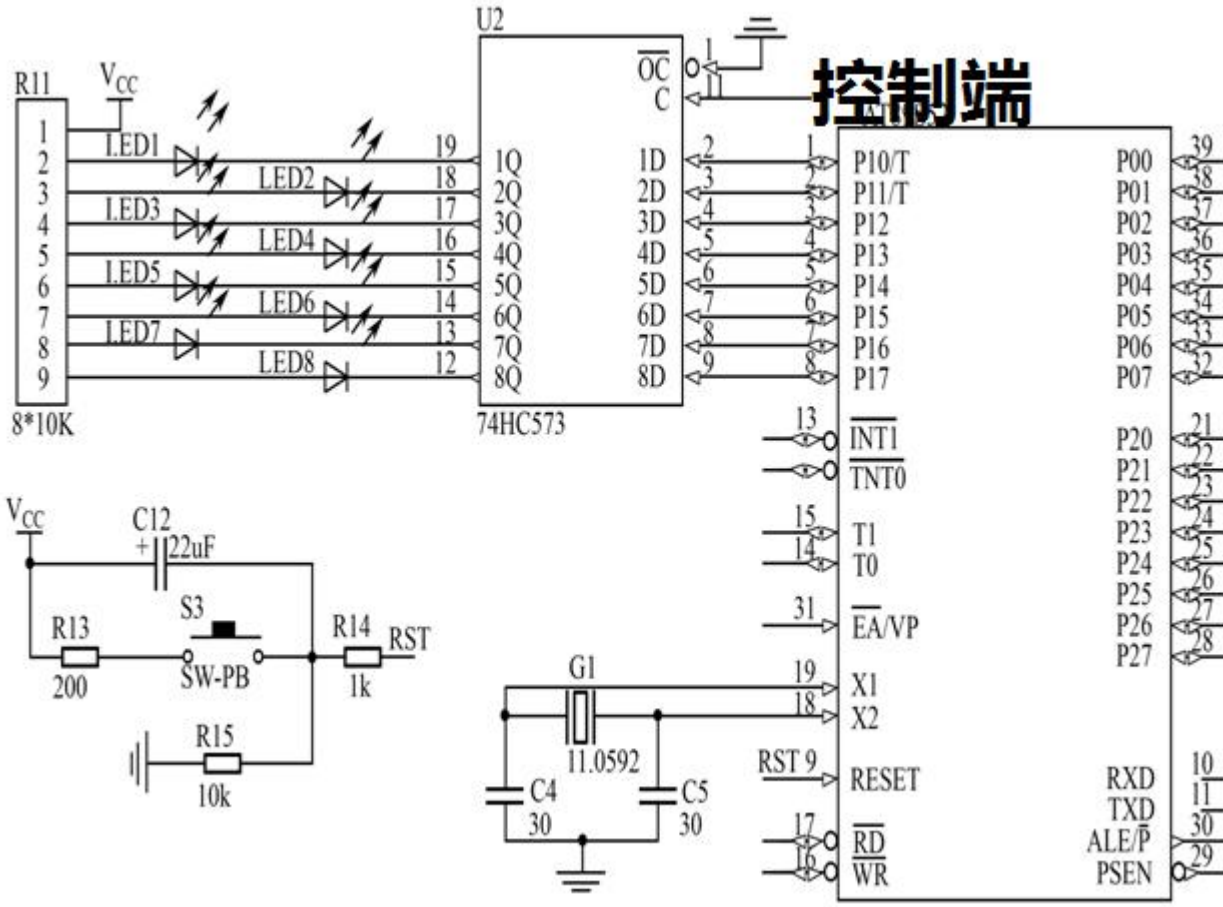


图 4-3　流水灯硬件原理图

对照图 4-3 发光二极管的电路原理图，回答以下问题：

1．74HC73 的功能是什么，进行发光二极管实验时，控制端（锁存允许信号）如何接线？

P1 口与发光二极管之间连接的通、断。　D 是数据输入端，Q 数据输出端，实验时，锁存器输入和输出连通，控制端接电源。

2、多个 LED 的连接方式是共阴方式还是共阳方式？

共阳。

3、 要使第一个 LED 点亮，单片机应输出什么信号？单片机复位后，8 个 LED 的状态如何？

0 低电平　FFH　全灭。

4.写程序让第二、四、五、六发光管闪烁**闪烁；**

```
#include <reg52.h>
#define uint unsigned int
uint a;
void main()
{
    while(1)
    {
    P1=0xc5;
    a=50000;
    while(a--);
    P1=0xff;
    a=50000;
    while(a--);
    }
}
```

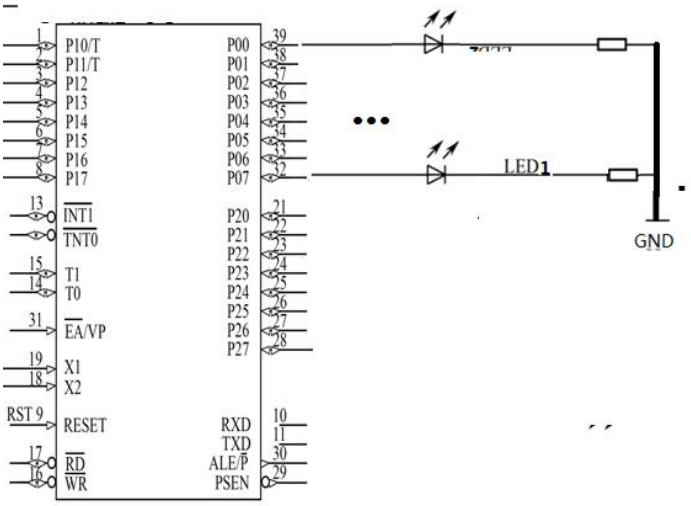


图 2 发光二极管的硬件原理图

对照上图发光二极管的电路原理图，回答以下问题：

1．进行发光二极管实验时，该接线方式正确与否，不正确的，指出错误并进行更正？

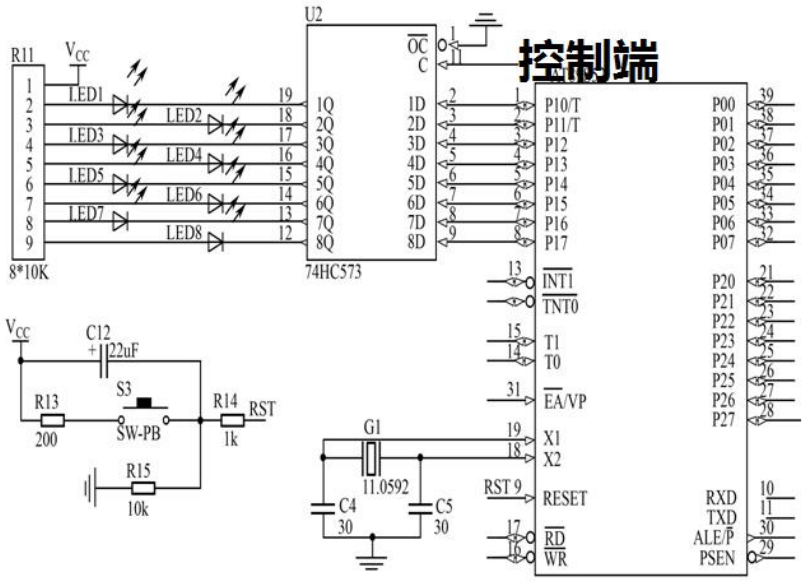
不正确， 缺少锁存器。

2、多个 LED 的连接方式是共阴方式还是共阳方式？
共阴。

3、 要使第一个 LED 点亮，单片机应输出什么信号？单片机复位后，8 个 LED 的状态如何？
1 高电平　　全亮

4.写程序让第 1 个发光管闪烁**闪烁；**

```
#include <reg52.h>
#define uint unsigned int
uint a;
void main()
{
    while(1)
    {
    P0=0x80;
    a=50000;
    while(a--);
    P0=0;
    a=50000;
    while(a--);
    }
}
```



按照以上电路连接原理图，编写以下程序(延时程序按带参数子函数调用方式)：

1、 8 个发光管由上至下间隔 1s 流动，其中每个管亮 500ms,灭 500ms,一直重复下去。

```
#include <reg52.h>
#include <intrins.h>
#define uchar unsigned char
```

```
uchar temp;
void delay(uint z)
 {
      uint x,y;
      for(x=z;x>0;x--)
            for(y=110;y>0;y--);
 }
void main()
{
      temp=0xfe;
      while(1)
      {
      P1=temp;
      delay(500);
      P1=0xff;
      delay(500);
      temp=_crol_(temp,1);
      }
}
```

2、 用 8 个发光管演示出 8 位二进制数累加过程。

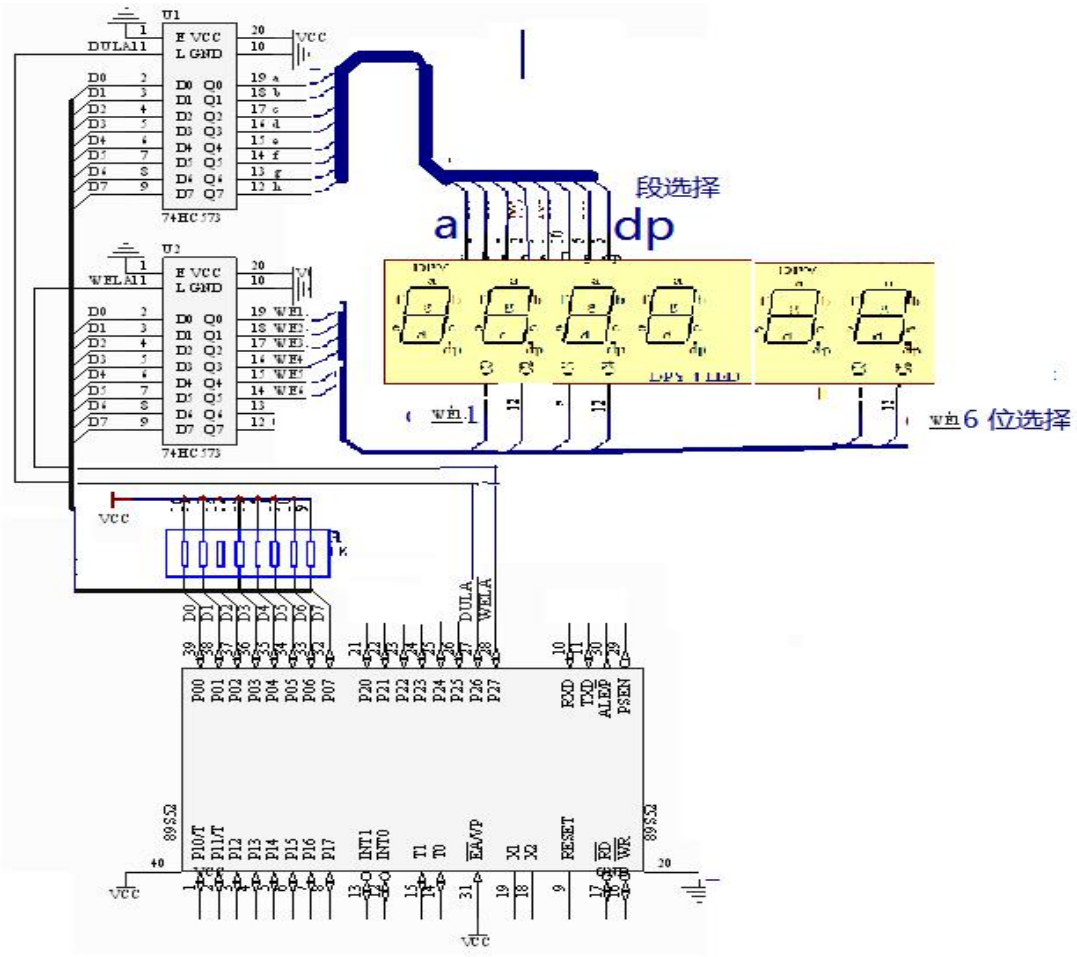
```
#include <reg52.h>
#define uint unsigned int
#define uchar unsigned char
uchar temp;
uint i;
void delay(uint z)
 {
      uint x,y;
      for(x=z;x>0;x--)
            for(y=110;y>0;y--);
 }
void main()
{
      temp=0xfe;
      for(i=0;i<256;i++)
    {
      P1=temp;
      delay(500);
      temp--;
    }
}
```

3、 间隔 300ms 第一次一个管亮流动一次，第二次两个管亮流动，依次到 8 个管亮，然后重

复整个过程。

```
#include <reg52.h>
#define uint unsigned int
uint a;
void delay(uint z)
 {
      uint x,y;
      for(x=z;x>0;x--)
            for(y=110;y>0;y--);
 }
void main()
{
      P1=0xff;
      for(a=0;a<8;a++)
{
      P1=P1<<1;
      delay(300);
      }

}
```



数码管与单片机引脚连接如上图所示，其中单片机的 P0 口分别连接两个锁存器 74HC573 的 D0~D7 端；单片机的 P2.6 连接段选锁存器的锁存允许信号端 L，用于控制输出段码；单片机的 P2.7 连接位选锁存器的锁存允许信号端 L，用于控制输出位码。

按照以上电路连接原理图，编写以下程序：

1、 第一个数码管显示 1，时间为 0.5s，然后关闭它，立即让第二个数码管显示 2.时间为 0.5s，再关闭它。。。。。，一直到最后一个数码管显示 6，时间为 0.5s，关闭它后再回来显示第一个数码管，一直循环下去；

```
#include <reg52.h>
#include <intrins.h>
#define uchar unsigned char
#define uint    unsigned int
sbit wela = P2^7;
sbit dula = P2^6;
uchar num,temp;
uchar code leddata[]={0x06,0x5b,0x4f,0x66,0x6d,0x7d};
void delay(uint z)
{     uint x,y;
      for(x = z; x > 0; x--)
            for(y = 110; y > 0 ; y--);       }
void main()
```

```
{
    while(1)
    {   temp=0xfe;
        for(num=1;num<=6;num++)
        {
              dula=1;
              P0=leddata[num];
              dula=0;

              wela=1;
              P0=temp;
              temp=_crol_(temp,1);
              wela=0;
              delay(500);
        }
    }
}
```

2、用定时器 0 的方式 1 实现用最后两位数码管，以 1s 间隔，进行 59s 循环计时；

```
#include<reg52.h>
#include<intrins.h>
#define uint unsigned int
#define uchar unsigned char
uchar num=0,aa,ge,shi;
sbit dula=P2^6;
sbit wela=P2^7;
uchar code table[]={
0x3f,0x06,0x5b,0x4f,
0x66,0x6d,0x7d,0x07,
0x7f,0x6f,0x77,0x7c,
0x39,0x5e,0x79,0x71};
uchar code tablewe[]={
0xfe,0xfd,0xfb,0xf7,0xef,0xdf};
void delay(uint z);
void main()
{

  aa=0;
  TMOD=0x01;
  TH0=(65536-46080)/256;
  TL0=(65536-46080)%256;
```

```
  EA=1;
  ET0=1;
  TR0=1;
  while(1)
  {
                    shi=num/10;
                    ge=num%10;
                    dula=1;
                    P0=table[shi];
                    dula=0;
                    P0=0xff;

                    wela=1;
                    P0=0xef;
                    wela=0;
                        delay(1);


                    dula=1;
                    P0=table[ge];
                    dula=0;
                    P0=0xff;

                    wela=1;
                    P0=0xdf;
                    wela=0;
                        delay(1);
  }
}
void delay(uint z)
{
  uint x,y;
  for(x=z;x>0;x--)
        for(y=110;y>0;y--);
}
void timer0() interrupt 1
{
  TH0=(65536-46080)/256;
  TL0=(65536-46080)%256;
  aa++;
         if(aa==20)
            {
                aa=0;
                num++;
```

```
                if(num==59)
                    num=0;
            }
    }
```

1、已知 P0.0~P0.5 为位选，其经过反相器 74LS06 输出至共阴数码管的位选端； P1 为段选，直接与数码管段选端 a~dp 连接，按照如上硬件连接方式，在数码管上动态显示“168168”；

```
#include <reg52.h>
#include <intrins.h>
#define uchar unsigned char
#define uint    unsigned int

void delay(uint z)
{
    uint x,y;
    for(x = z; x > 0; x--)
        for(y = 114; y > 0 ; y--);
}
void display(uchar a,uchar b)
{
    P0=a;
    P1=b;
    delay(1);
    P1=0xff;
}

void main()
{
    display(0x20,0x06);
    display(0x10,0x7d);
    display(0x08,0x7f);
    display(0x04,0x06);
    display(0x02,0x7d);
    display(0x01,0x7f);
}
```

2、已知小键盘与单片机的硬件连接线方式如图 2：行信号：P3.4~P3.7，列信号：P3.0~P3.3；数码管采用 P0.0~P0.5 为位选，其经过反相器 74LS06 输出至共阴数码管的位选端； P1 为段选，直接与数码管段选端 a~dp，编程实现扫描按下的键值，并用数码管显示键值。

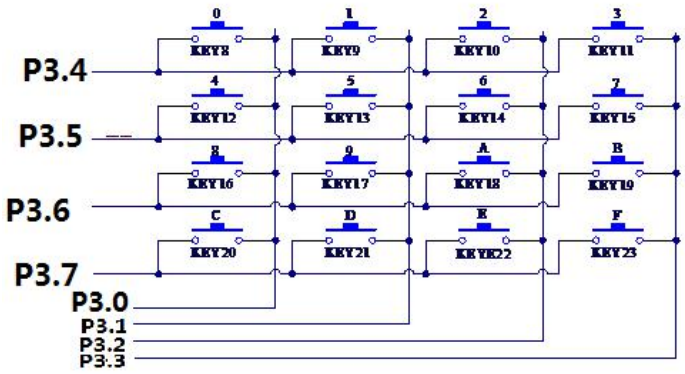


图 2　4×4 阵列式建盘电路

```
#include <reg52.h>
#include <intrins.h>
#define uchar unsigned char
#define uint    unsigned int
uchar num,temp;
uchar code number[]={0x3f,0x06,0x5b,0x4f,
0x66,0x6d,0x7d,0x07,
0x7f,0x6f,0x77,0x7c,
0x39,0x5e,0x79,0x71,0};

void delay(uint z)
{
      uint x,y;
      for(x = z; x > 0; x--)
            for(y = 112; y > 0 ; y--);
}
void display(uint i)
{
      P1=number[i];
}

uchar keyscan()
{
                  P3=0xfe;
                  temp=P3;
                  temp=temp&0xf0;
                  while(temp!=0xf0)
                        {
                              delay(5);
                              temp=P3;
                              temp=temp&0xf0;
```

```
            while(temp!=0xf0)
            {
                temp=P3;
            switch(temp)
                {
                    case 0xee:num=0;
                        break;
                    case 0xde:num=4;
                        break;
                    case 0xbe:num=8;
                        break;
                    case 0x7e:num=12;
                        break;
                }
                    while(temp!=0xf0)
                {
                    temp=P3;
                    temp=temp&0xf0;
                }
            }
      }
P3=0xfd;
temp=P3;
temp=temp&0xf0;
while(temp!=0xf0)
      {
            delay(5);
            temp=P3;
            temp=temp&0xf0;
            while(temp!=0xf0)
            {
                temp=P3;
            switch(temp)
                {
                    case 0xed:num=1;
                        break;
                    case 0xdd:num=5;
                        break;
                    case 0xbd:num=9;
                        break;
                    case 0x7d:num=13;
                        break;
                }
                    while(temp!=0xf0)
```

```
                {
                    temp=P3;
                    temp=temp&0xf0;
                }
            }
        }
P3=0xfb;
temp=P3;
temp=temp&0xf0;
while(temp!=0xf0)
    {
        delay(5);
        temp=P3;
        temp=temp&0xf0;
        while(temp!=0xf0)
        {
            temp=P3;
        switch(temp)
            {
                case 0xeb:num=2;
                    break;
                case 0xdb:num=6;
                    break;
                case 0xbb:num=10;
                    break;
                case 0x7b:num=14;
                    break;
            }
                while(temp!=0xf0)
            {
                temp=P3;
                temp=temp&0xf0;
            }
        }
    }
P3=0xf7;
temp=P3;
temp=temp&0xf0;
while(temp!=0xf0)
    {
        delay(5);
        temp=P3;
        temp=temp&0xf0;
        while(temp!=0xf0)
```

```
                    {
                        temp=P3;
                    switch(temp)
                        {
                            case 0xe7:num=3;
                                break;
                            case 0xd7:num=7;
                                break;
                            case 0xb7:num=11;
                                break;
                            case 0x77:num=15;
                                break;
                        }
                        while(temp!=0xf0)
                        {
                            temp=P3;
                            temp=temp&0xf0;
                        }
                    }
                }
return num;
}

void main()
{
    num=17;
    P1=0;
    P0=0x01;
    while(1)
    {
    display(keyscan());
    }
```