项目二 接口与显示技术实训

本项目通过五个任务分别让学生掌握单片机与数码管的接口技术，能完成单片机的数码管静态及动态显示电路和程序设计；能完成点阵显示电路设计，能够用C语言实现对点阵屏的控制，最终实现点阵屏汉字、日期和温度显示等功能；能独立完成单片机键盘电路的设计，能使用C语言实现对键盘的扫描和按键识别控制程序的设计、运行及调试；能利用AT89S52单片机及1602液晶屏，通过按键设置液晶电子钟电路设计、C语言程序设计，完成按键设置液晶电子钟的设计、运行及调试。项目二作为单片机的接口显示技术，对后续项目的学习有很大的帮助和指导作用。

【知识目标】

1. 掌握LED数码管的结构、工作原理和显示方式；

2. 掌握数码管静态、动态显示的原理；

3. LED点阵显示系统结构与原理；

4. 键盘的防抖动措施；

5. 键盘的接口方法和编程方法；

6. 了解1602液晶屏结构；

7. 掌握1602液晶屏工作原理；

8. 掌握1602液晶屏与单片机的接口方法。

【技能目标】

1. 掌握数码管静态、动态显示的电路设计及程序设计；

2. 8*8LED点阵汉字显示电路、程序设计；

3. 8*8LED 点阵屏整体测试；

4. 矩阵式键盘设计与实现；

5. 会利用 I/O 口进行键盘、液晶显示电路设计；

6. 掌握液晶屏显示程序的设计方法。

【情感目标】

1. 培养学生谦虚、好学的态度，能利用各种信息媒体，获取新知识、新技术；

2. 培养学生勤于思考、做事认真的良好作风，能立足专业规划自己未来的职业生涯；

3. 培养学生良好的职业道德；

4. 培养学生勇于创新、敬业乐业的工作作风。

任务一　数码管显示 0-9

【任务目标】

通过控制一个数码管静态显示 0-9，掌握单片机常用外部电路（数码管）的设计方法，综合、灵活运用 C 语言进行编程，为后期学习打好基础。

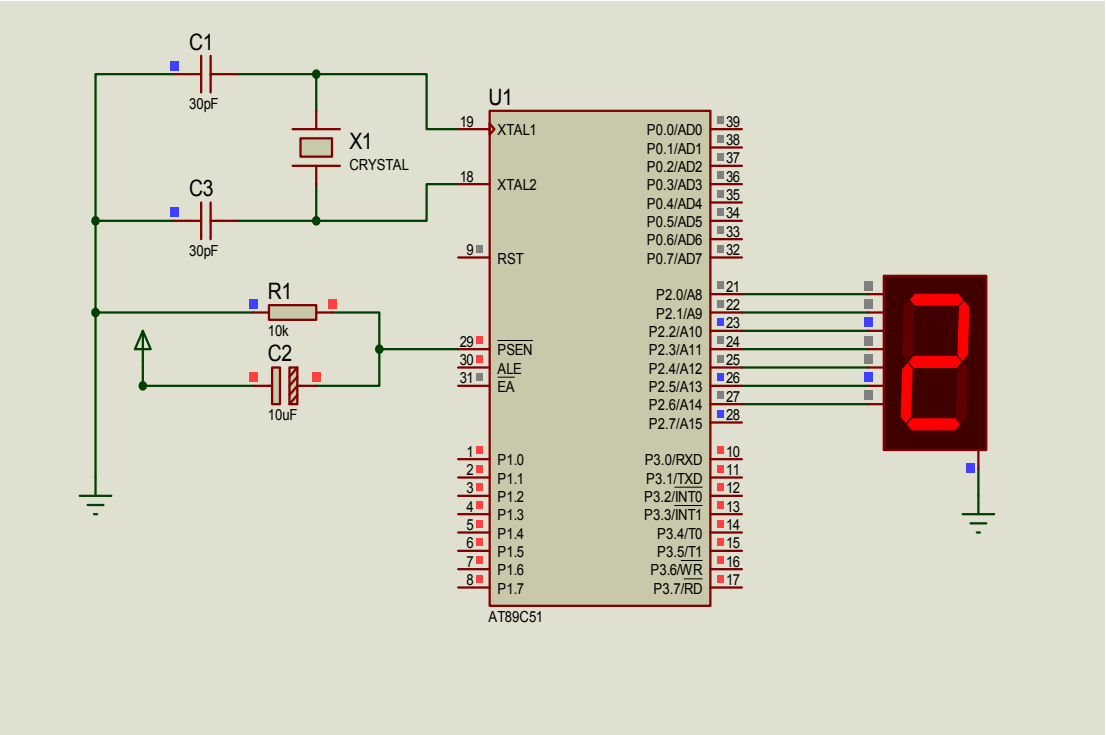
【任务要求】

1、完成仿真电路搭建；

2、建立工程 2-1-1，编写代码实现数码管 0-9 显示控制；

3、建立工程 2-1-2，编写代码实现数码管 0-99 显示控制。

【仿真电路图】



【任务咨询】

1. LED 数码管分类

按其内部结构可分为共阴型和共阳型；

按其外形尺寸有多种形式，使用较多的是 0.5"和 0.8"；

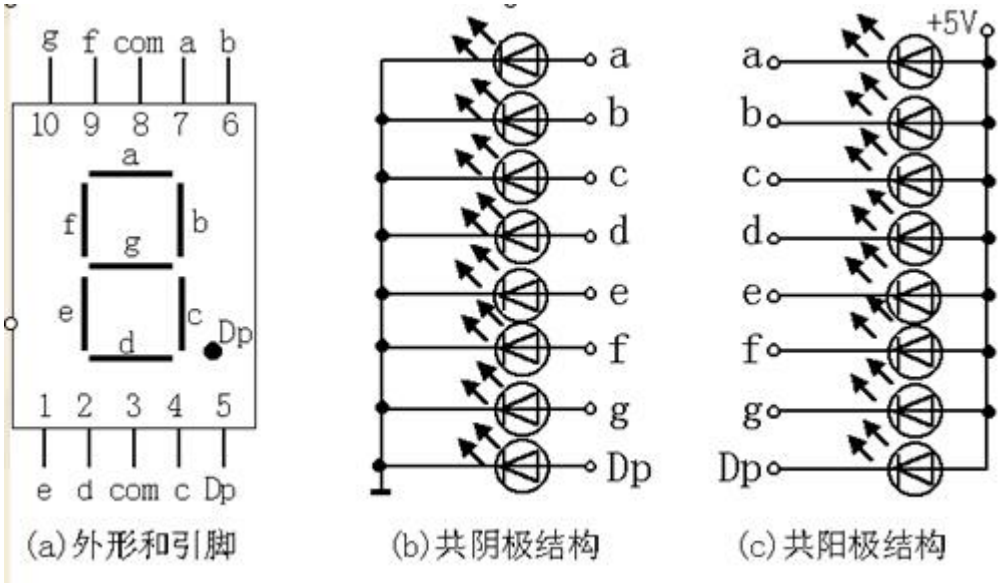
按显示颜色也有多种形式，主要有红色和绿色；

按亮度强弱可分为超亮、高亮和普亮。

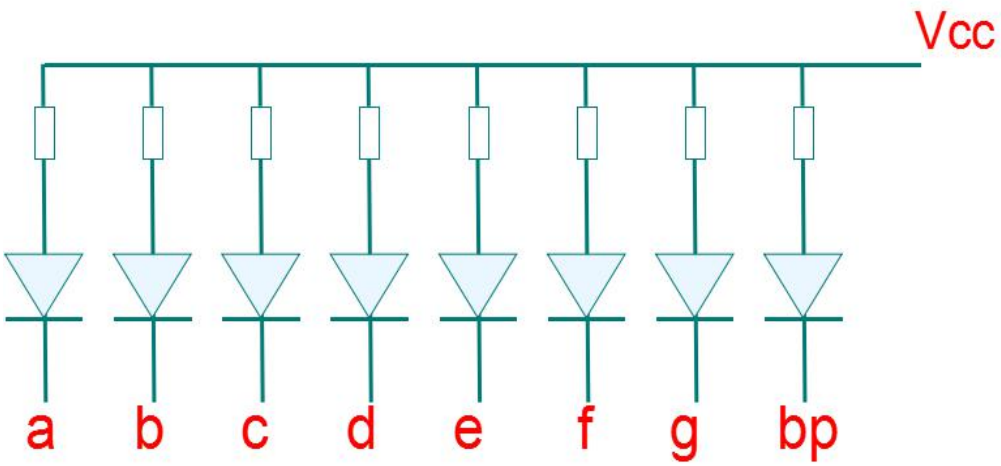
LED 数码管特性：正向压降一般为 1.5～2V，额定电流为 10mA，最大电流为 40mA。静态显示时取 10mA 为宜，动态扫描显示，可加大脉冲电流，但一般不超过 40mA。

2. LED 数码管结构

共阳数码管每个段笔画是用低电平(“0”)点亮的,要求驱动功率很小；而共阴数码管段笔画是用高电平(“0”)点亮的，要求驱动功率较大。



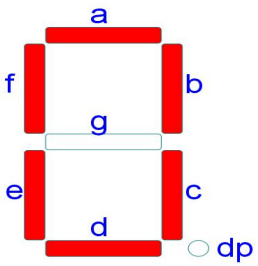
共阳极数码管：仅当段位接低电平，阳极接高电平时，相应位的LED才导通发光。



七段数码管的段位控制：

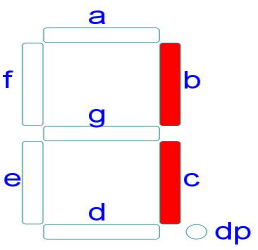
显示0编码：

dp	g	f	e	d	c	b	a
0	0	1	1	1	1	1	1



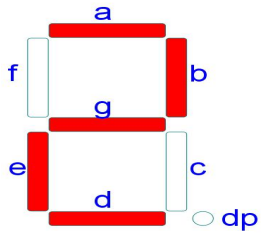
显示1编码：

dp	g	f	e	d	c	b	a
0	0	0	0	0	1	1	0



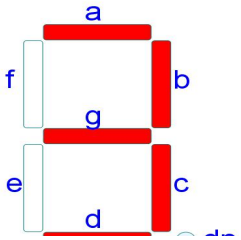
显示 2 编码：

dp	g	f	e	d	c	b	a
0	1	0	1	1	0	1	1



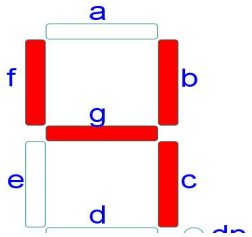
显示 3 编码：

dp	g	f	e	d	c	b	a
0	0	1	1	1	1	1	1



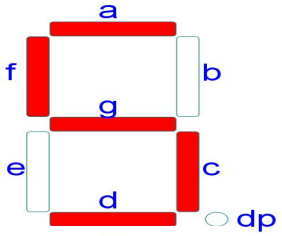
显示 4 编码：

dp	g	f	e	d	c	b	a
0	1	1	0	0	1	1	0



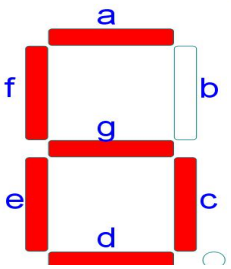
显示 5 编码：

dp	g	f	e	d	c	b	a
0	1	1	0	1	1	0	1



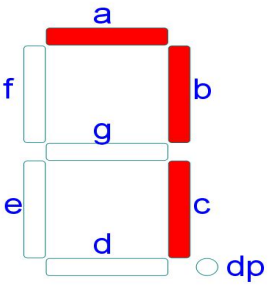
显示 6 编码：

dp	g	f	e	d	c	b	a
0	1	1	1	1	1	0	1



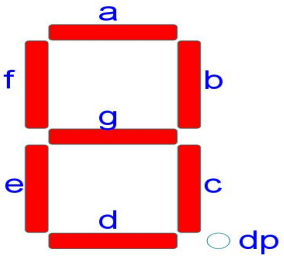
显示 7 编码：

dp	g	f	e	d	c	b	a
0	0	0	0	0	1	1	1



显示 8 编码：

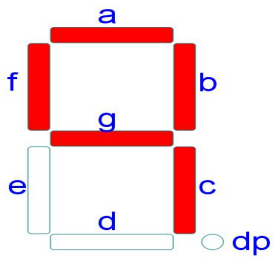
dp	g	f	e	d	c	b	a
0	1	1	1	1	1	1	1



显示 9 编码：

dp g f e d c b a

0 1 1 0 0 1 1 1



3. LED 数码管编码方式

表 1　共阴和共阳 LED 数码管几种八段编码表

显示数字	共阴顺序小数点暗									共阴逆序小数点暗									共阳顺序小数点亮	共阳顺序小数点暗
	Dp	g	f	e	d	c	b	a	16 进制	a	b	c	d	e	f	g	Dp	16 进制		
0	0	0	1	1	1	1	1	1	3FH	1	1	1	1	1	1	0	0	FCH	40H	C0H
1	0	0	0	0	0	1	1	0	06H	0	1	1	0	0	0	0	0	60H	79H	F9H
2	0	1	0	1	1	0	1	1	5BH	1	1	0	1	1	0	1	0	DAH	24H	A4H
3	0	1	0	0	1	1	1	1	4FH	1	1	1	1	0	0	1	0	F2H	30H	B0H
4	0	1	1	0	0	1	1	0	66H	0	1	1	0	0	1	1	0	66H	19H	99H
5	0	1	1	0	1	1	0	1	6DH	1	0	1	1	0	1	1	0	B6H	12H	92H
6	0	1	1	1	1	1	0	1	7DH	1	0	1	1	1	1	1	0	BEH	02H	82H
7	0	0	0	0	0	1	1	1	07H	1	1	1	0	0	0	0	0	E0H	78H	F8H
8	0	1	1	1	1	1	1	1	7FH	1	1	1	1	1	1	1	0	FEH	00H	80H
9	0	1	1	0	1	1	1	1	6FH	1	1	1	1	0	1	1	0	F6H	10H	90H

【任务实施】

1. 搭建仿真电路图。本任务使用 P2 口连接 1 个共阴极数码管；

2. 通过建立工程 2-1-1，把以下程序代码放到 Keil 编译软件工具中，生成 HEX 文件，加载到仿真电路图中，看显示效果。

```
#include"reg51.h"
#define uchar unsigned char
#define uint unsigned int
uchar code tab[10]=
{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
delay(uint t)
{
   uint i,j;
   for(j=t;j>0;j--)
   for(i=110;i>0;i--);
}
main()
{
   uint n;
   while(1)
   {
      for(n=0;n<10;n++)
      {
         P2=tab[n];
         delay(1000);
      }
   }
```

```
}
```

3.在已有的仿真图的基础上,在增加一个数码管连接在P3口上，实现0-99的计数效果。通过建立工程2-1-2，把以下程序代码放到Keil编译软件工具中，生成HEX文件，加载到仿真电路图中，看显示效果。

```
#include"reg51.h"
#define uchar unsigned char
#define uint unsigned int
uchar code tab[10]=
{0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
delay(uint t)
{
   uint i,j;
   for(j=t;j>0;j--)
      for(i=110;i>0;i--);
}
main()
{
    uint n,a,b;
     while(1)
      {
          for(n=0;n<99;n++)
```

```
            {
                a=n%10;
                 b=n/10;
                P2=tab[b];
                P1=tab[a];
                delay(1000);
            }
        }
    }
```

【任务评价】

任务检测		分值	学生互评（40%）	老师评估（60%）	任务总评
任务知识内容	数码管基本知识	20			
	电路图设计	20			
	程序代码编写	20			
	综合调试	20			
现场管理	出勤情况	5			
	课程纪律	5			
	团队协作精神	5			
	保持实验室卫生	5			

【任务练习】

1.要求在增加一个数码管连接到P0口，实现0-999的计数效果；

2.仿真成功后，将代码下载到试验箱继续调试。

【知识拓展】

1．LED数码管控制系统

（1）防水LED护栏管一般是在外罩接口处用热熔胶或硅胶密封，内部LED、电路板都是裸露的，由于昼夜温差大，外罩的端头与外罩

热胀冷缩不同，导致热熔胶或硅胶密封处出现缝隙，下雨后雨水渗进内部，造成电路短路而烧毁LED。要解决这个问题，一定要求对内部电路和LED进行灌胶处理。接头单用热熔胶或硅胶密封固然简单，但可靠性达不到在户外应用的要求。

（2）防紫外线LED护栏管由于要求混光防雨，外面都会有外罩，外罩的材料选择是很多不规范公司降低成本的又一个手段，质量好的产品都会使用增加了抗紫外线的材料。如GE、拜尔等材料，而质量不好的LED护栏管很多使用混合了水口料的材料，谈不上抗紫外线，太阳光比较大的地方，不到一个月，外罩就变成黄色的了，从而出光效果变差，透光率也大大减小。

（3）线损有色金属涨价很多，采用劣质材料和减小线径是一些厂家节省成本的方法，一般好的生产厂家都会在内部使用1mm以上的导线，而且导线材料使用的是符合国标的产品。如果导线面积不够或材料的杂质太多，电阻值就较大，前面的护栏管和后面的护栏管就有较大的电压差，为了让后面的LED也能正常工作，就需要增加输入电压，这样无形中就增加了功耗，很多电能不是用于驱动LED，而是浪费在导线和恒流芯片上。通用的恒流芯片都有功耗要求，电压高功耗就大，如果功耗太大，热量散不出去就会导致芯片烧毁。这就是为什么很多LED护栏管都是前面损坏得多的原因了。

（4）散热一般LED护栏管外罩和底座完全是一体的，都是塑胶材料。当LED排布很密时，在通电热平衡后，LED的结温已经很高了，就将造成LED的寿命急剧减少。实力强的LED护栏管公司肯定会有热

设计人员，在设计护栏管外罩时就会想法把LED的热量和恒流芯片的热量有效的传导到大气中去。底座使用铝材是比较好的方法。另外在设计时要尽量将PCB靠近铝底座。

（5）供电护栏管的供电有两种方式：220V的高压和48V以下的低压。220V直接供电是一种危险而又不经济的方式，低压的做法是用电容或电阻降压后供给。这种方式首先是安全性就存在很大的问题；其次是太耗电并对电网有极大的损害，供电部门是坚决反对的。由于突然在线路上加接了大量的容性负载，使供电线路失配形成了自激震荡，就可能出大事故。

2.应用领域

主要用于楼体亮化，广告招牌、高档的DISCO、酒吧、夜总会、会所的门头广告牌等。特别适合应用于广告牌背景、立交桥、河、湖护栏、建筑物轮廓等大型动感光带的夜景照明之中，可产生彩虹般绚丽的效果。 用护栏管装饰建筑物的轮廓，可以起到突出美彩亮化建筑物的效果。事实证明，它已经成为照明产品中的一只奇葩，绽放在动感都市。

3.LED驱动电流

（1）显示效果： 由于LED基本上属于电流敏感元件，其正向压降的分散性很大，并且还与温度有关，为了保证数码管具有良好的亮度均匀度，就需要使其具有恒定的工作电流，且不能受温度及其它因素的影响。另外，当温度变化时驱动晶片还要能够自动调节输出电流的大小以实现色差平衡温度补偿。

（2）安全性： 即使是短时间的电流超载也可能对发光管造成永久性的损坏，采用恒流驱动电路后可防止由于电流故障所引起的数码管的大面积损坏。 另外，我们所采用的超大型积体电路还具有级联延时开关特性，可防止反向尖峰电压对发光二极体的损害。超大型积体电路还具有热保护功能，当任何一片的温度超过一定值时可自动关断，并且可在控制室内看到故障显示。

【任务思考】

1.填空题。

（1）共阳LED数码管加反相器驱动显示字符“6”的段码是_____。

（2）在共阳极数码管使用中，若5要显示小数点，则其相应的字段码是_____。

（3）共阴LED数码管显示字符“A”的段码是_____。

（4）共阳LED数码管公共端接的是_____。

（5）共阴LED数码管公共端接的是_____。

2.思考题

（1）在实际中，数码管显示不是很明显，原因是什么，应该怎么改变，效果更好一点？

（2）在静态显示下，单片机最多可以接多少数码管？有没有别的方式可以增加数码管？

3.技能提高

训练任务1：利用仿真图，修改程序代码，是数码管显示从99开始，大约1S减1，减到0，设计方案如何修改？

评价标准：流程图绘制、硬件电路原理图修改、软件程序修改、软硬件联调、实物连接。

训练任务 2:实现 0-9999 的计数,电路如何连接？程序如何修改？

评价标准：硬件电路原理图修改、软件程序修改、软硬件联调、实物连接。

任务二　数码管动态显示 0-999999

【任务目标】

通过控制一个数码管动态显示 0-999999，掌握单片机常用外部电路（数码管动态显示）的设计方法，综合、灵活运用 C 语言进行编程，为后期学习打好基础。

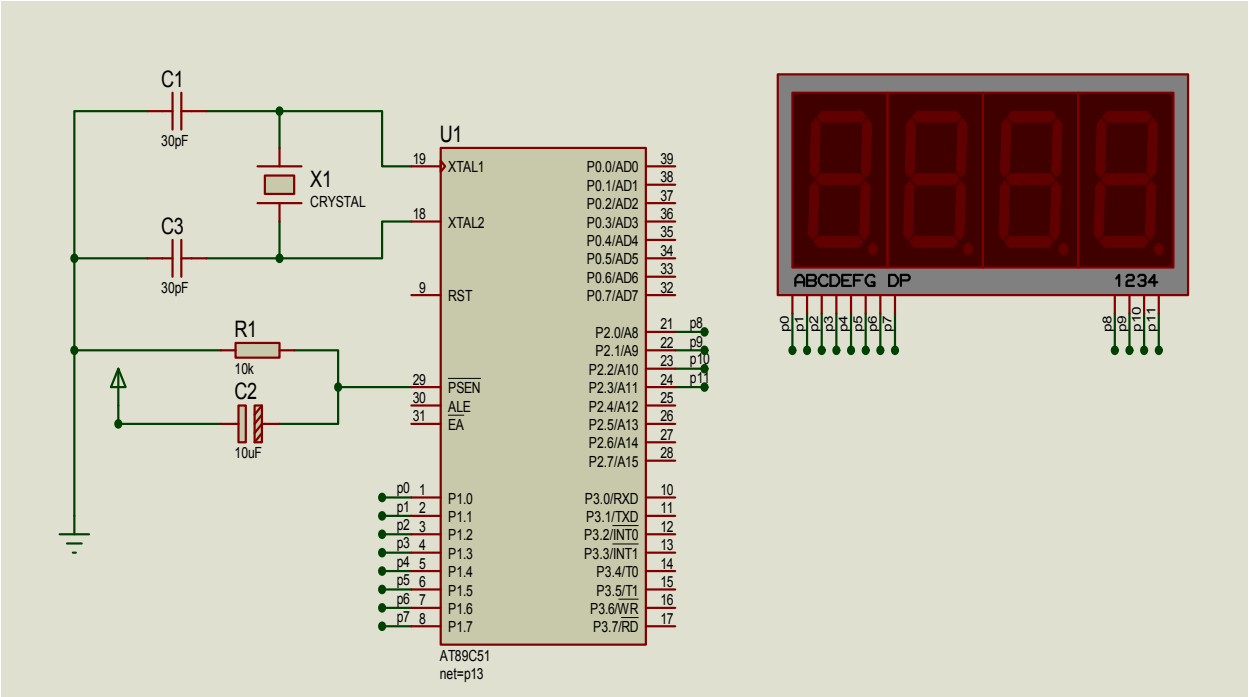
【任务要求】

1、根据任务提示的电路图，在 Proteus 中完成仿真电路搭建；

2、建立工程 2-2-1，根据任务咨询，在 keil 上编写代码实现数码管 0-9999 动态显示控制；

3、建立工程 2-2-2，在工程 2-2-1 的代码基础上，在 keil 上编写编写代码实现数码管 0-999999 动态显示控制。

【仿真电路图】



【任务咨询】

1. 动态显示方式

（1）动态显示电路连结形式：

① 显示各位的所有相同字段线连在一起，共 8 段，由一个 8 位 I/O 口控制；

② 每一位的公共端（共阳或共阴 COM）由另一个 I/O 口控制。

（2）工作原理：

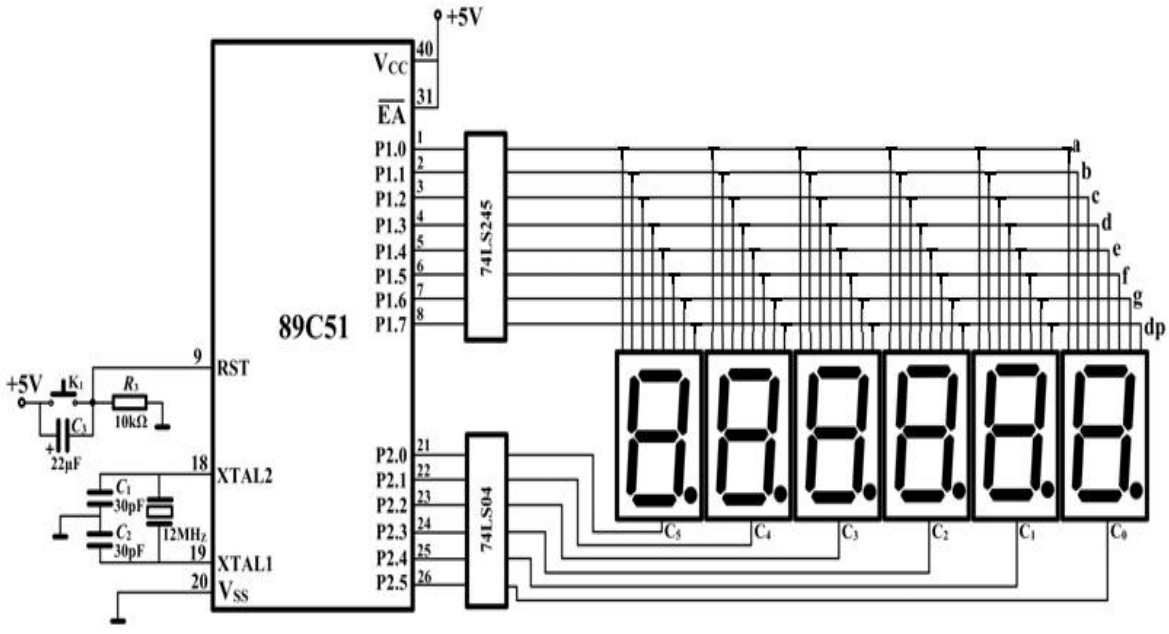
从 P0 口送段代码,P1 口送位选信号。段码虽同时到达 6 个 LED,但一次仅一个 LED 被选中。利用“视觉暂留”，每送一个字符并选中相应位线，延时一会儿,再送/选下一个……循环扫描即可。

要求：此处为共阴数码管，P0 口送段代码，P1 口送位选信号。通过查表实现动态显示。

条件：待显数据(00H—09H)已放在：7FH—7AH 单元中(分别对应十万位～个位)。

说明：由于用了反相驱动器 7406，要用共阳译码表。

2.典型电路



动态显示是一种按位轮流点亮各位数码管的显示方式，即在某一时段，只让其中一位数码管“位选端”有效，并送出相应的字型显示编码。此时，其它位的数码管因“位选端”无效而都处于熄灭状态；下一时段按顺序选通另外一位数码管，并送出相应的字型显示编码，依此规律循环下去，即可使各位数码管分别间断地显示出相应的字符。这一过程称为动态扫描显示。

【任务实施】

1.搭建仿真电路图。本任务使用P1口的8个引脚作为动态数码管的段选端，P2口的前4位作为的动态数码管的位选端；

2.把以下程序代码放到Keil编译软件工具中，生成HEX文件，加载到仿真电路图中，看显示效果。

```
#include"reg51.h"//库文件声明
#define uint unsigned int//宏定义无符号整型常量
#define uchar unsigned char//宏定义无符号字符型常量
uchar code tab[10]={0x3f,0x06,0x5b,0x4f,
0x66,0x6d,0x7d,0x07,0x7f,0x6f};//数码管显示0-9编码
delay(uint t)//延时函数
{
   uint i,j;
   for(j=t;j>0;j--)
      for(i=110;i>0;i--);//执行空语句
}
```

```
main()//主函数
{
uint n,m,a,b,c,d;
  while(1)
    {
       for(n=0;n<9999;n++)//确定显示范围 0-9999
         for(m=0;m<11;m++)//多次扫描显示状态
       {
           a=n%10;//分离四位数的个位
           b=n%100/10;//分离四位数的十位
           c=n%1000/100;//分离四位数的百位
           d=n/1000;//分离四位数的千位
           P2=0x01;//选通位选端的第 4 个数码管
           P1=~tab[d];//P1 口送入千位段选码
           delay(20);
           P2=0X00;//去阴影
           P2=0x02;//选通位选端的第 3 个数码管
           P1=~tab[c];//P1 口送入百位段选码
          delay(20);//视觉延时
           P2=0X00;//去阴影
          P2=0x04;//选通位选端的第 2 个数码管
          P1=~tab[b];//P1 口送入十位段选码
```

```
            delay(20);//视觉延时
            P2=0X00;//去阴影
             P2=0x08;//选通位选端的第 1 个数码管
            P1=~tab[a];//P1 口送入个位段选码
            delay(20);//视觉延时
            P2=0X00; //去阴影
            }
        }
    }
```

【任务评价】

任务检测		分值	学生互评（40%）	老师评估（60%）	任务总评
任务知识内容	数码管编码知识	20			
	电路图认知	20			
	任务程序设计	20			
	任务完成效果	20			
现场管理	出勤情况	5			
	实验纪律	5			
	团队协作精神	5			
	保持实验室卫生	5			

【任务练习】

1. 要求把四位数码管换成六位数码管，段选端接 P1 口不变，位选端接 P2 口不变，连接电路，修改程序代码完成任务，实现 0-999999；

2. 仿真成功后，将代码下载到试验箱继续调试。

3. 仿真电路图不变，修改代码实现 999999-0 的显示控制。

【知识拓展】

1.LED 数码管结构特点

led 数码管常用段数一般为 7 段有的另加一个小数点，还有一种是类似于 3 位“+1”型。位数有半位，1，2，3，4，5，6，8，10 位等等....，led 数码管根据 LED 的接法不同分为共阴和共阳两类，了解 LED 的这些特性，对编程是很重要的，因为不同类型的数码管，除了它们的硬件电路有差异外，编程方法也是不同的。共阴和共阳极数码管的发光原理是一样的，只是它们的电源极性不同而已。颜色有红，绿，蓝，黄等几种。led 数码管广泛用于仪表，时钟，车站，家电等场合。选用时要注意产品尺寸颜色，功耗，亮度，波长等。

透过分时轮流控制各个 LED 数码管的 COM 端，就使各个数码管轮流受控显示，这就是动态驱动。每位元数码管的点亮时间为 1～2ms，由于人的视觉暂留现象及发光二极体的余辉效应，尽管实际上各位数码管并非同时点亮，但只要扫描的速度足够快，给人的印象就是一组稳定的显示资料，不会有闪烁感，动态显示的效果和静态显示是一样的，能够节省大量的 I/O 口，而且功耗更低。

2.LED 数码管光源

总的来说，LED 光源的来源有两种做法： 一种是使用传统小功率 LED 作组合，一般多达上百颗甚至数百颗，电源设计复杂。另一种是使用大功率管作光源，价格比较贵。

两种方法都不可避免地要将散热设计和工作可靠性作为主要设计考虑因素，国内多应用于政府示范性工程，真正市场化运作

的工程很少，国外这方面的应用实例较多，但其最大的缺点依然是可靠性、出光流明数和价格，很多工程由于LED品质低劣，没有很好地表现出寿命长的优点。还有，从成本、市场的角度考虑，LED作为照明光源，其是否与太阳能结合使用，在设计上需要走不同的路线，并不是单独作为一种光源来开发就能完成的。

3. LED引脚测量

找公共共阴和公共共阳，首先，我们找个电源（3到5伏）和不同规格的电阻，VCC串接个电阻后和GND接在任意2个脚上，组合有很多，但总有一个LED会发光的找到一个就够了，然后用GND不动，VCC（串电阻）逐个碰剩下的脚，如果有多个LED（一般是8个），那它就共阴的了。相反用VCC不动，GND逐个碰剩下的脚，如果有多个LED（一般是8个），那它就共阳的。也可以直接用数位万用表，红表笔是电源的正极，黑表笔是电源的负极。

4.LED常见故障检修

（1）不亮：

产生此故障的原因可能有：

1）变压器损坏、引线断开或虚焊。若变压器损坏，则予以更换或重新绕制；若引线断开或虚焊，则应重新连接或重新焊接。

2）+5V电源故障。应检查显示电路电源电压及数码管供电是否正常（正常时为5V直流电压）。若两者不正常则检查7805三端稳压器是否有5V电压输出。若三端稳压器有5V电压输出，则为电路板的连接线损坏，或未开炉或短路，若开路则应重新连接；若短路应清理

短路点。若三端稳压器没有5V电压输出，则检查 7805输人端有无8V左右的电压。若7805输入端有8V电压,则说明7805损坏。若7805输入端没有8V电压，则检查桥式整流电路有无8V左右电压输出。若整端电路没有8V电压输出，则检查220V交流电源是否正常。若交流电源不正常，则应对交流电源进行检查。若交流电源220V电压正常，则在连接线正常的情况下应为变压器损坏，予以更换。

3）LED数码管公共阳极（或公共阴极）上无+5V电压。这种故障常常是由限流电阻发生虚焊或印制导线不通引起的。若无5V电压，则按照前面所述检查电源电压的方法进行检查；若限流电阻开路，则应予以更换；若为电路板虚焊或印制导线不通，则必须重新连接或焊接。

4）数码管内部损坏，则应予以更换。

（2）仅小数点亮

这一现象表明±5V电源正常，产生这种故障的原因可能有：

1）若A/D转换器工作不正常，则应更换A/D转换器。

2）若数码管内部有故障，则应更换数码管。

3）若A/D转换器与数码管之间的连线开路，则应重新连接或新焊接。

4）若A/D转换器集成元器件损坏，则应更换集成元器件。

5）若A/D转换器管脚接触不良，则应必须重新焊接。

（3）缺笔段

对于A/D转换器为7107的仪表，可检查7107用于显示输出的各

对应输出脚与V+（正电源脚）闸的电压（即用万用表的黑表笔接所对应的输出脚，红表笔接电源供电正端电压点），若始终为几毫伏，则表明7107坏，应予以更换；如有4V左右（数码管管脚端为负），LED却不亮，则表时7107输出正常，接着可检查数码管的各自对应管脚与V+之间的电压（万用表的黑表笔接所对应数码管的管脚，红表笔接电源供电正端电压点），若也有4V左右，则表明LED数码管坏∵应更换该数码管；若无电压，则可能是7107管脚接触不良，或从7107到数码管管脚之间的连接线路不通，应重新连接和重新焊接。若两处电压不同，则检查电路板从7107到数码管之间有没有腐蚀或油污，若有，则必须清洗干净并作好绝缘处理工作；若没有，则数码管内部短路，必须予以更换。

（4）亮度不足

产生这种故障的原因可能是：

1）LED数码管使用时间太久，发光效率降低，可更换新的数码管。

2）R9阻值太大，可重新更换或重新调整。

3）电路板漏电，与将电路板上的电容放电或清洗电路板，并经干燥处理后，再接线使用。

4）若供电不足，检查供电电路的连接电阻是否变值，若变值，则应予以更换；若稳压器负载功率下降，则应予以更换；若滤波电容漏电，则应予以更换；若整流二极管损坏，则应予以更换；若变压器负载功率下降，则应予以更换；若常规检查后仍然供电不足，则检查

所有负载电路中有无对地短路元件，若检查出短路元件后，则应予以更换。

5）A/D 电路的输出电压不足，先检查外围电路有无对地短路元件，若有对地短路元件，则应予以更换；若检查正常，则为 A/D 转换集成电路损坏，应予以更换。

【任务思考】

1 思考题

（1）LED 数码管的静态与动态显示方式主要区别在那？

（2）一般在什么情况下，在设计电路或者程序时采用数码管动态显示方式？

2. 技能提高

训练任务 1：修改任务 2 的仿真图，用 P0 口控制段选端，用 P3 口控制位选端，修改工程 2-2-1 实现 0-9999 的计数，设计方案如何修改？

评价标准：流程图绘制、硬件电路原理图修改、软件程序修改、软硬件联调、实物连接。

训练任务 2：用任务 2 的仿真图，在此基础上增加 4 个按键，分别控制 4 个数码管，第一个按键按下第一个数码管显示数字 1，第二个按键按下第二个数码管显示数字 2，第三个按键按下第三个数码管显示数字 3，第四个按键按下第四个数码管显示数字 4，电路如何连接？程序如何修改？

评价标准：硬件电路原理图修改、软件程序修改、软硬件联调、

实物连接。

训练任务 3：用训练任务 2 的仿真图，第一个按键按下第一个数码管循环显示 0-9，第二个按键按下前 2 位数码管循环显示 0-99，第三个按键按下前三位数码管循环显示 0-999，第四个按键按下前四位数码管循环显示 0-9999，电路如何连接？程序如何修改？

评价标准：硬件电路原理图修改、软件程序修改、软硬件联调、实物连接。

任务三 点阵显示电子广告牌

【任务目标】

通过利用单片机制作一个 8×8 点阵的电子广告牌，显示出字符“大”的设计与仿真演示，掌握单片机常用外部电路（点阵显示）的设计方法，综合、灵活运用 C 语言进行编程，为后期学习打好基础。

【任务要求】

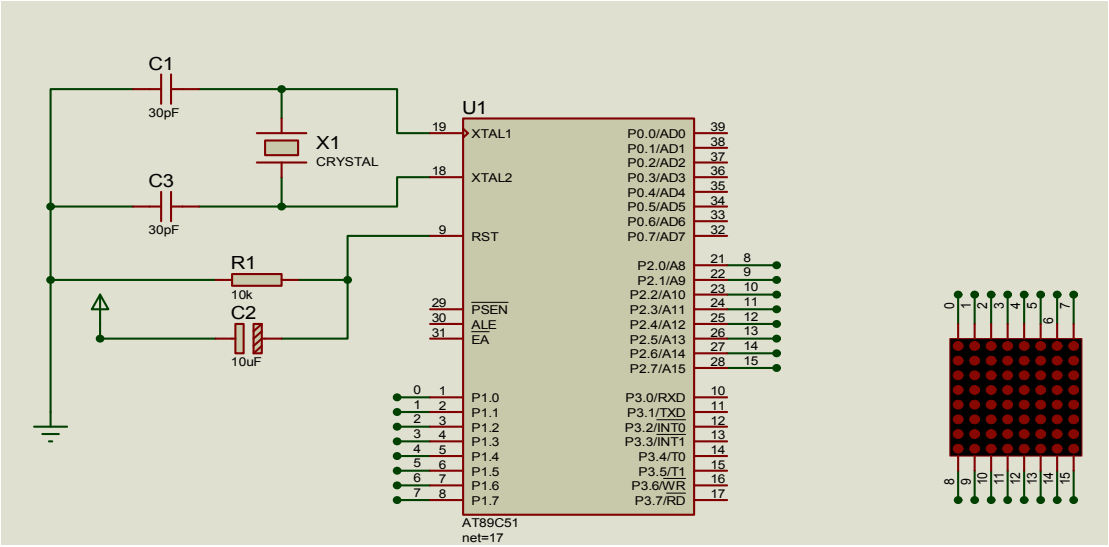
1、采用项目二任务 3 的仿真电路，根据任务咨询，建立工程 2-3-1，编写代码完成 8×8 点阵显示出字符“大”的效果；

2、采用项目二任务 3 的仿真电路，根据任务咨询，建立工程 2-3-2，编写代码完成 8×8 点阵轮流显示出字符“0-9”具有翻页的效果；

3、采用项目二任务 3 的仿真电路，根据任务咨询，建立工程 2-3-3，编写代码完成 8×8 点阵显示出字符“LOVE”具有上移的效果；

4、用项目二任务 3 的仿真电路，根据任务咨询，建立工程 2-3-4，编写代码完成 8×8 点阵显示出字符“LOVE”具有左移的效果。

【仿真电路图】

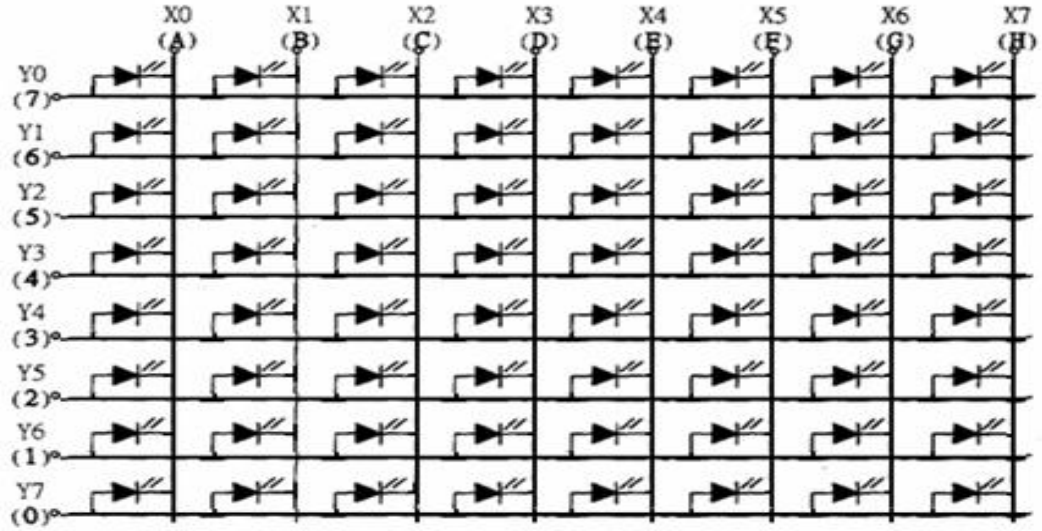


【任务咨询】

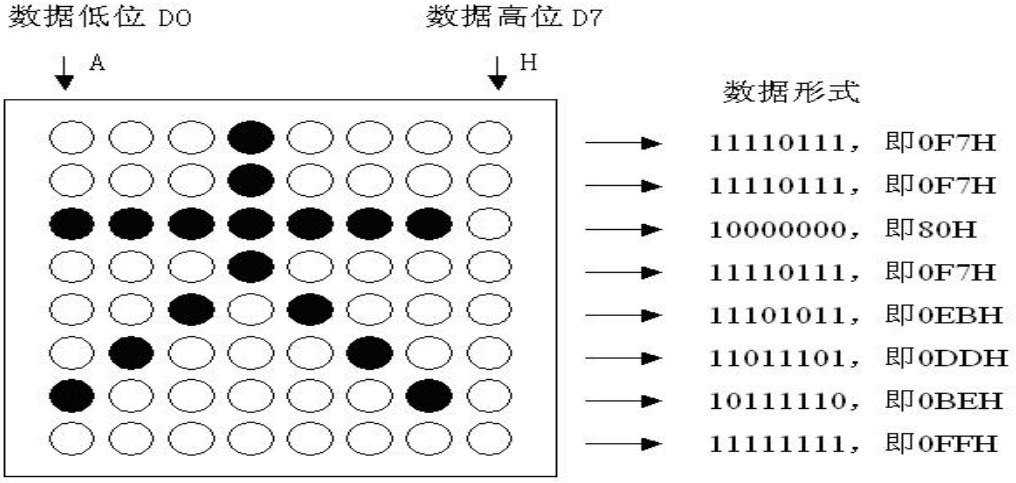
1. LED 大屏幕显示器原理

LED 点阵显示器是把很多 LED 发光二极管按矩阵方式排列在一起，通过对每个 LED 进行发光控制，完成各种字符或图形的显示。最常见的 LED 点阵显示模块有 5×7（5 列 7 行），7×9（7 列 9 行），8×8（8 列 8 行）结构。LED 点阵由一个一个的点（LED 发光二极管）组成，总点数为行数与列数之积，引脚数为行数与列数之和。

2. LED 大屏幕显示器结构

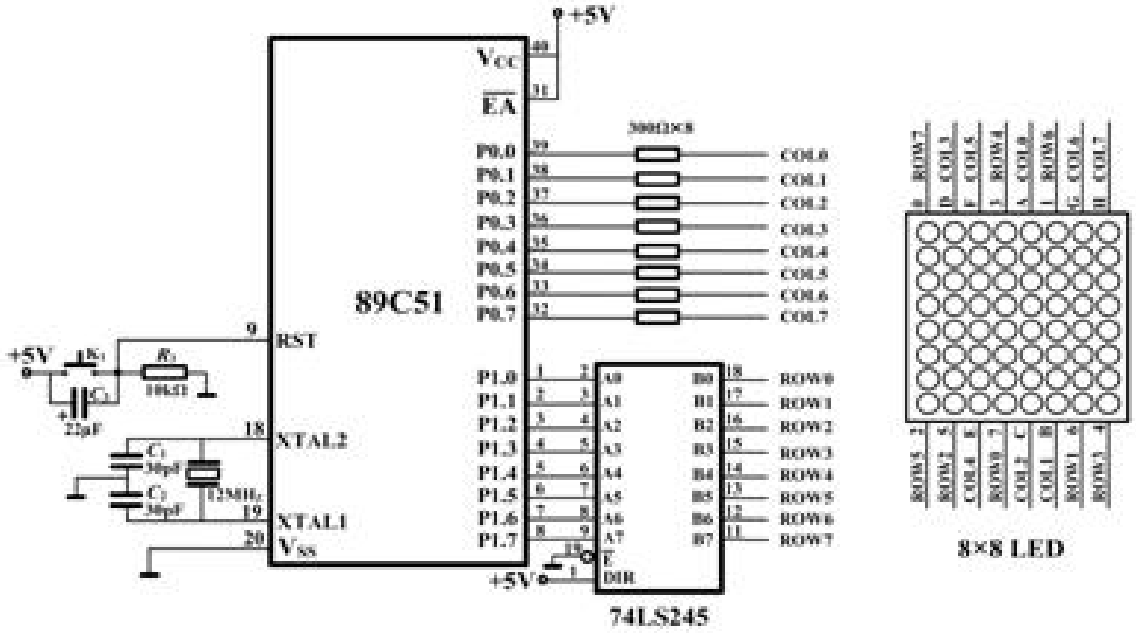


3. “大”字显示字型码示意图



显示字符“大”的过程如下：先给第一行送高电平（行高电平有效），同时给 8 列送 11110111（列低电平有效）；然后给第二行送高电平，同时给 8 列送 11110111，……最后给第八行送高电平，同时给 8 列送 11111111。每行点亮延时时间为 1ms，第八行结束后再从第一行开始循环显示。利用视觉驻留现象，人们看到的就是一个稳定的图形。

4.LED 大屏幕显示器接口



【任务实施】

1.搭建仿真电路图。本任务使用 P1 口控制 8×8 点阵的行选，用 P2 口控制 8×8 点阵的列选；

2.通过建立工程 2-3-1，把以下程序代码放到 Keil 编译软件工具中，生成 HEX 文件，加载到仿真电路图中，看显示效果。

```
#include  "reg51.h"
#define uint unsigned int
```

```
void delay1ms();      //延时约 1ms 函数声明
void main()
{
   unsigned char code led[]=
      {0xf7,0xf7,0x80,0xf7,0xeb,0xdd,0xbe,0xff};
    unsigned char w[8]=
      {0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
   uint  i,m;
   while(1) {
      for(m=0;m<400;m++)   //每个字符扫描显示 400 次,控制
每个字符显示时间
        {
          for(i=0;i<8;i++)
          {
             P2=w[i];    //行数据送 P1 口
             P1=~led[i]; //列数据送 P0 口
             delay1ms();
          }
        }
   }
}
//函数名: delay1ms
```

```
void delay1ms()
{
   uint i;
   for(i=0;i<200;i++);
}
```

3.通过建立工程2-3-2，把以下程序代码放到Keil编译软件工具中，生成HEX文件，加载到仿真电路图中，看显示效果，比较工程2-3-1效果，进行比较分析。

```
#include  "REG51.H"
void delay1ms();      //延时约1ms函数声明
void main()
{
    unsigned char code led[]=
       {0xf7,0xf7,0x80,0xf7,0xeb,0xdd,0xbe,0xff,      //0
        0x00,0x18,0x1c,0x18,0x18,0x18,0x18,0x18,      //1
       0x00,0x1e,0x30,0x30,0x1c,0x06,0x06,0x3e,       //2
       0x00,0x1e,0x30,0x30,0x1c,0x30,0x30,0x1e,       //3
       0x00,0x30,0x38,0x34,0x32,0x3e,0x30,0x30,       //4
       0x00,0x1e,0x02,0x1e,0x30,0x30,0x30,0x1e,       //5
       0x00,0x1c,0x06,0x1e,0x36,0x36,0x36,0x1c,       //6
       0x00,0x3f,0x30,0x18,0x18,0x0c,0x0c,0x0c,       //7
       0x00,0x1c,0x36,0x36,0x1c,0x36,0x36,0x1c,       //8
```

```
        0x00,0x1c,0x36,0x36,0x36,0x3c,0x30,0x1c};     //9
        unsigned char w[8]=
            {0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
        unsigned int i,j,k,m;
        while(1) {
            for(k=0;k<10;k++)     //字符个数控制变量
            {
                for(m=0;m<200;m++)    //每个字符扫描显示 400
次，控制每个字符显示时间
        {
            j=k*8;//指向数组 led的第 k个字符第一个显示码下标
            for(i=0;i<8;i++)
            {
                P2=~w[i];       //行数据送 P1 口
                P1=led[j];  //列数据送 P0 口
                delay1ms();
                j++;      //指向数组中下一个显示码
            }
        }
      }
    }
  }
```

```
//函数名：delay1ms
//函数功能：采用软件实现延时约1ms
//形式参数：无
//返回值：无
void delay1ms()
{
    unsigned int i;
        for(i=0;i<200;i++);
}
```

4.通过建立工程2-3-3，把以下程序代码放到Keil编译软件工具中，生成HEX文件，加载到仿真电路图中，看显示效果，比较工程2-3-1及2-3-2效果，进行比较分析。

```
#include<reg51.h>
#define uchar unsigned char
#define uint unsigned int
uchar code TB[]=
    {0x01,0x02,0x04,0x08,0x10,0x20,0x40,0x80};
uchar code TA[]=
    {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,//空屏
    0xFD,0xFD,0xFD,0xFD,0xFD,0xFD,0xC1,0xFF, //L
    0xE3,0xDD,0xDD,0xDD,0xDD,0xDD,0xE3,0xFF, //O
    0xDD,0xDD,0xDD,0xDD,0xDD,0xEB,0xF7,0xFF, //V
```

```
0xC1,0xFD,0xFD,0xC1,0xFD,0xFD,0xC1,0xFF, //E
0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF, //空屏
 };
uchar i,t;
delay(uchar t)
{
   while (t--)
{;}}
void main(void)
{
   uchar N,T;
   while(1)
       {
      for(N=0;N<40;N++)/*循环扫描一遍 40 帧（5 个字每次
   显示要 8 帧=40）*/
          for(T=0;T<60;T++) //移动速度
              {
                 for(i=0;i<8;i++)
                  {
                    P2=~TB[i];
                      P1=~TA[i+N];
                    delay(100);
```

```
            }
        }
    }
}
```

4.通过建立工程2-3-4，把以下程序代码放到Keil编译软件工具中，生成HEX文件，加载到仿真电路图中，看显示效果，比较工程2-3-1、2-3-2、2-3-2效果，进行比较分析。

```
/*8X8行扫描，左移显示*/
#include<reg51.h>
#define uchar unsigned char
#define uint unsigned int
uchar code TAB[]=
    {0xFF,0xF7,0xFB,0x81,0xFB,0xF7,0xFF,0xFF};
uchar i,t,j=0;
delay(uchar t)
{
   while (t--)
       {;}
}
void main(void)
   {
   uchar T,Y,Q;
```

```
        while(1)
        {
            for(Q=0;Q<8;Q++)
            for(T=0;T<100;T++)   //速度
            {
                P2=0x01;
                for(i=0;i<8;i++)
                {
                    Y=TAB[i+1]*256+TAB[i];
                      Y=Y<<(7-Q)|Y>>Q;          /*保证第 9 个数据
的最高位移到第二次数据的最低处，再输入到列端口*/
                      P1=Y%256;          /*P1=TAB[i]*/
                      delay(60);
                        P2=P2<<1|P2>>7;
                 }
             }
        }
    }
```

如果将扫描方式改为列扫描，那么左右移动的程序就容易写了，但当点阵比较巨大并且硬件已经定下时，改变扫描方式不是好方法，甚至不可能实现。这里是以行扫描为例（逐行取字模），第一次取字码数组中的第 1～8 个数据到点阵列输入端，行码扫描 1～8 行。第二

次将第一次的 1～8 个数据都循环左（右）移一位，并且将第 9 个数据的最高位移到第二次数据的最低处，再输入到列端口，行扫描 1～8 行。即每次扫描都要把前一次扫描的列码左移一位。

【任务评价】

任务检测		分值	学生互评（40%）	老师评估（60%）	任务总评
任务知识内容	点阵基本知识	20			
	电路图设计	20			
	程序设计	20			
	综合调试效果	20			
现场管理	出勤情况	5			
	实验室纪律	5			
	团队协作精神	5			
	保持实验室卫生	5			

【任务练习】

1. 利用工程 2-3-3，修改程序代码，实现字符“LOVE”在点阵下移的显示效果；

2. 利用工程 2-3-4，修改程序代码，实现字符“LOVE”在点阵右移的显示效果；

3. 仿真成功后，将代码下载到试验箱继续调试。

【知识拓展】

1．驱动

由 LED 点阵显示器的内部结构可知，器件宜采用动态扫描驱动方式工作，由于 LED 管芯大多为高型，因此某行或某列的单体 LED 驱动电流可选用窄脉冲，但其平均电流应限制在 20mA 内．多数点阵显示器的单体 LED 的正向压降约在 2V 左右．但大亮点∮10 的点阵显示器单体 LED 的正向压降约为 6V。

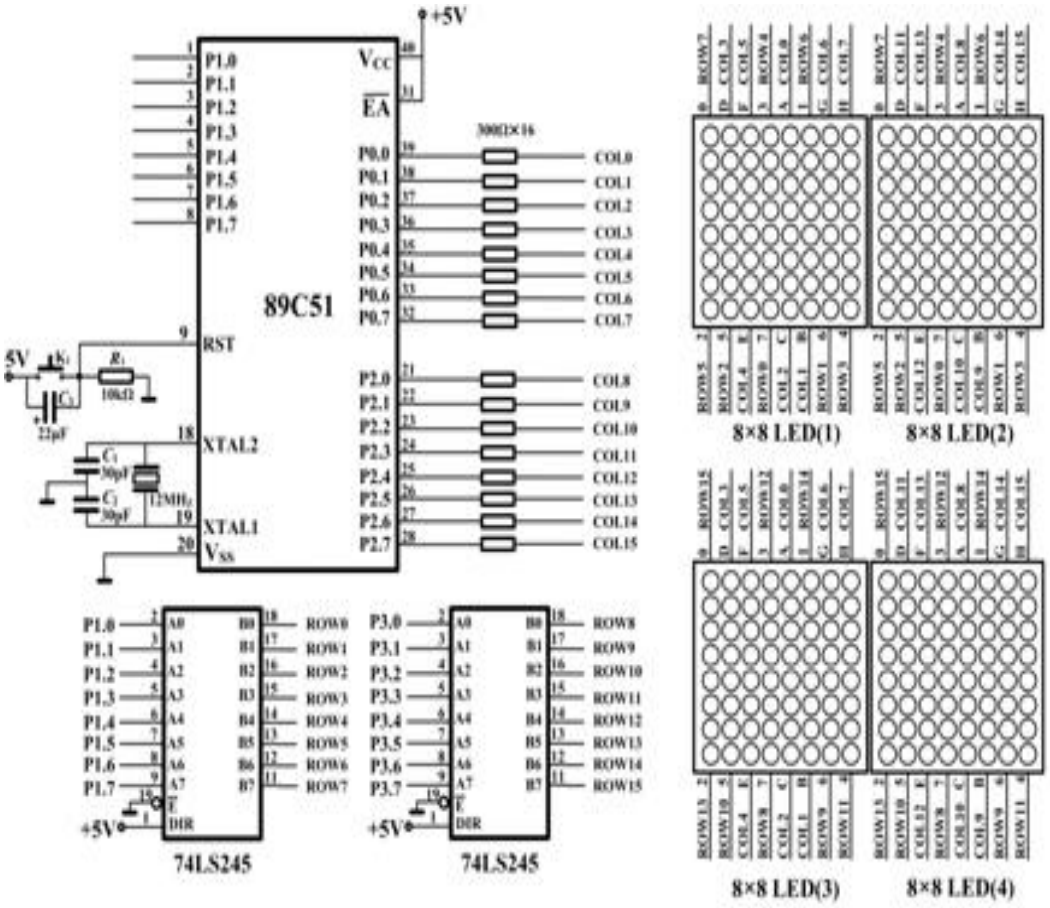
大屏幕显示系统一般是将由多个LED点阵组成的小模块以搭积木的方式组合而成的，每一个小模块都有自己的独立的控制系统，组合在一起后只要引入一个总控制器控制各模块的命令和数据即可，这种方法既简单而且具有易装、易维修的特点。

LED点阵显示系统中各模块的显示方式有静态和动态显示两种。静态显示原理简单、控制方便，但硬件接线复杂，在实际应用中一般采用动态显示方式，动态显示采用扫描的方式工作，由峰值较大的窄脉冲驱动，从上到下逐次不断地对显示屏的各行进行选通，同时又向各列送出表示图形或文字信息的脉冲信号，反复循环以上操作，就可显示各种图形或文字信息。

2.LED大屏幕显示器扩展接口

一个国标汉字是由16X16即256个点（像素）来构成的，显示一个汉字该亮哪些点这些复杂的工作都交给取模软件来完成，同时，取模软件也负责把要显示的汉字转化成程序中要用到的显示代码，代码以一定的规律表征了该亮的点（一般用“1”表示）与不该亮的点（一般用“0”表示），一共256位。单片机负责将这些代码一段一段有规律地送到LED屏，比如第一次输出表示第1列的16位代码点亮第1列、紧接着再输出16位代码去点亮第2列„„直到点亮第16列然后再重新点亮第1列„„，如此循环，就完成了这个汉字的显示。单片机输出的速度足够快时，由于视觉暂留现象使得人眼在同一时刻感受到了这16列输出的信息，也就是看到了这个汉字。由于AT89S52单片机是8位总线结构，一次不能输出16位代码以显示完整的一列，这

样我们把一个字拆分为上下两部分，一次送 8 位，一共送 32 次，这样同样完成了一个汉字的显示。事实上这个汉字区域也可以是在 256 象素范围内的任何图形。



【任务思考】

1.程序设计题

以下是 8*8 点阵下移显示字符“L0VE”的程序代码，部分代码缺失，根据提示完成代码。实现效果。

```
/*8X8 行扫描，下移显示*/
#include<reg51.h>
#define uchar unsigned char
#define uint unsigned int
```

```
uchar code TAB[]=
   {0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,     //空屏
    0xFD,0xFD,0xFD,0xFD,0xFD,0xFD,0xC1,0xFF,     //L
   0xE3,0xDD,0xDD,0xDD,0xDD,0xDD,0xE3,0xFF,     //O
    0xDD,0xDD,0xDD,0xDD,0xDD,0xEB,0xF7,0xFF,     //V
    0xC1,0xFD,0xFD,0xC1,0xFD,0xFD,0xC1,0xFF,     //E
    0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,     //空屏
    };
 uchar idata Buffer[48]={0};  //缓存显示单元
 uchar i,t;
 delay(uchar t)
{
 while (t--)
  {;}
 }
 void main(void)
{
  uchar N,T,m,n;
  for(m=0;m<6;m++)//循环扫描一遍 6 帧(6 帧能显示完整的 6
个字)
     for(n=0;n<8;n++)
      Buffer[8*m+n]=TAB[_________________];  //将 TAB 数组
```

中的数据重新排列 ，TAB 里面有 48 个数据/

```
        //使得下移字母顺序不变
     while(1)
    {
     for(N=0;N<_____________;N++)          //循环扫描一遍 6 帧
       for(T=0;T<70;T++)          //速度
      {
   P2=0x80;
       for(i=0;i<8;i++)
       {
        P1=Buffer[i+N];
       delay(100);
        P2=_________________;       //扫描起始行为第一行 /*显
```

示的结果可：P2=0X80，只显示第一行，在下面的 8 次小循环中，第一行一次显示了 L 从上到下的移动的每次第一行要显示的单元，当 P2=0X40，由于前面一行始终比当前一行早，所以总的显示 L 在向下移动*//

```
    }
            }
          }
        }
```

2.思考题

（1）LED 大屏幕点阵显示器一次能点亮多少行？显示的原理是怎样

的？

（2）LED 大屏幕点阵显示器的行和列是怎么区分的？

3. 技能提高

训练任务 1：独立设计一段代码，要求实现将字符“箭头”从左右滚动显示，字符“箭头”的字型编码如以下数组所示：{0xFF, 0xF7, 0xFB, 0x81, 0xFB, 0xF7, 0xFF, 0xFF}；方案如何设计？

评价标准：流程图绘制、硬件电路原理图修改、软件程序修改、软硬件联调、实物连接。

训练任务 2：在任务三仿真电路图的基础上，从 8*8 点阵扩展到 16*16 点阵电路，实现自己姓名的轮流显示效果，电路如何连接？程序如何设计？

评价标准：硬件电路原理图修改、软件程序修改、软硬件联调、实物连接。

任务四 数码管显示 4X4 矩阵键盘按键号

【任务目标】

通过 4 个独立式按键控制数码管数字显示及 4*4 矩阵式按键控制数码管显示对应数字的设计与仿真演示，熟悉独立式按键及矩阵式按键编程方法。

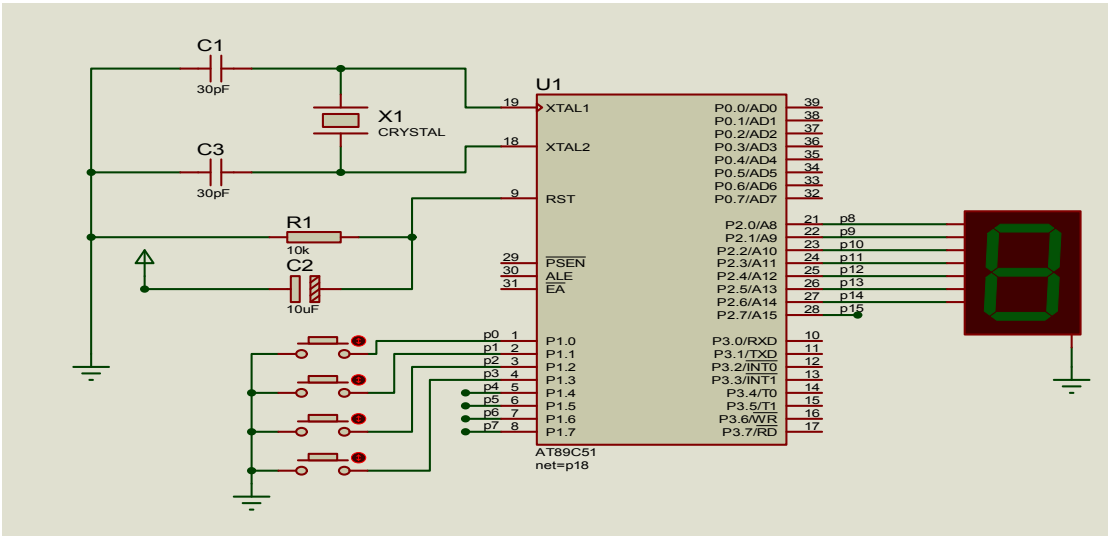
【任务要求】

1、采用项目二任务 4 的仿真电路 1，根据任务咨询，建立工程 2-4-1，编写代码完成 K1-K3 按键控制数码管显示数字加 1、减 1、归 0 的效果；

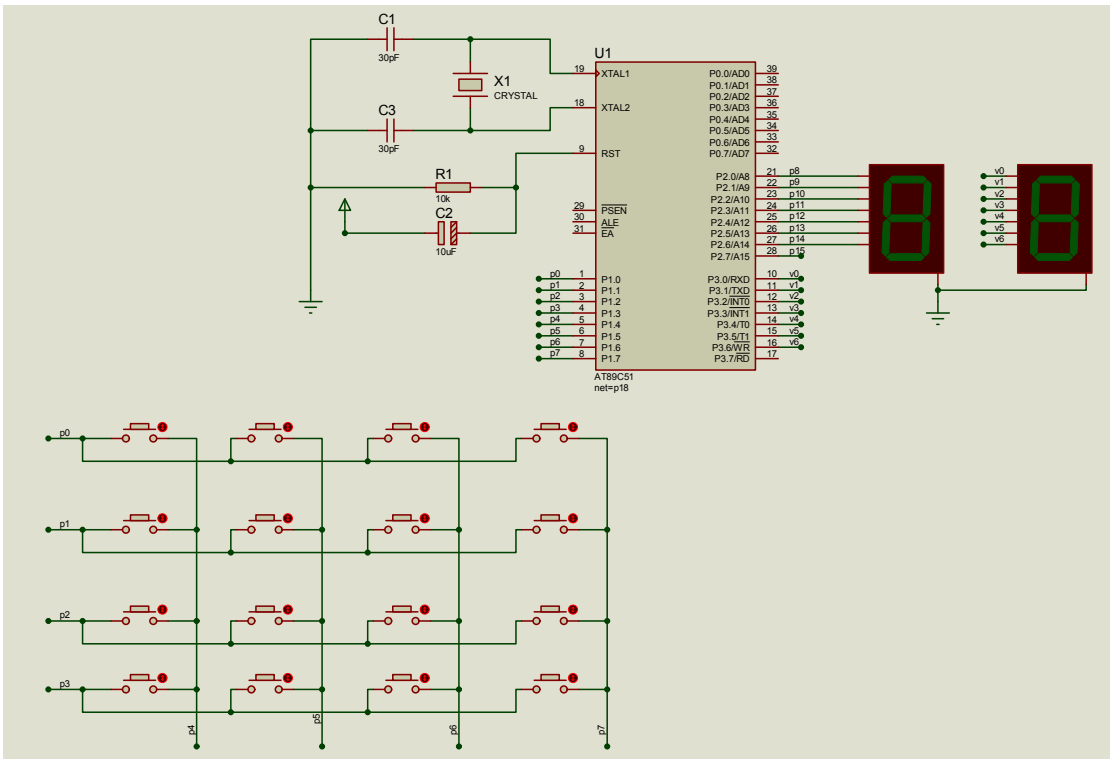
2、采用项目二任务 4 的仿真电路 2，根据任务咨询，建立工程 2-4-2，采用反转法检测形式，编写代码完成 4*4 按键控制数码管显示数字 1-16 效果；

3、建立工程 2-4-3，参照工程 2-4-2，编写代码完成 3*3 按键控制数码管显示数字 0-9 效果。

【仿真电路图】



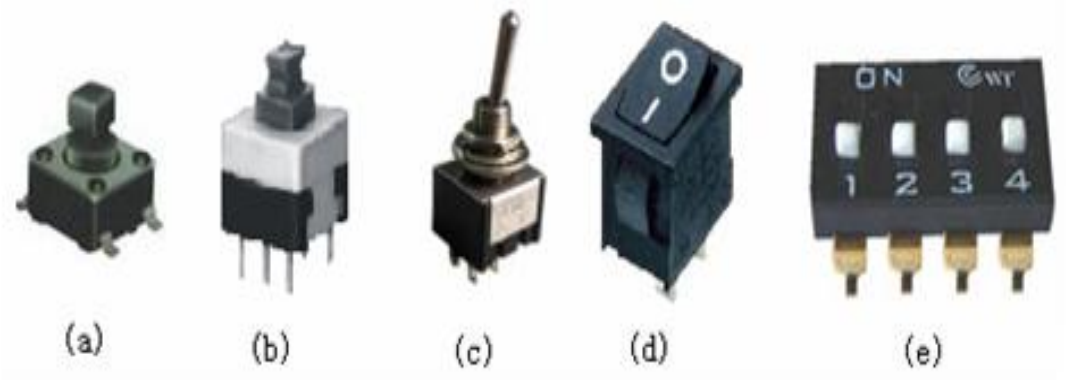
仿真电路 1



仿真电路 2

【任务咨询】

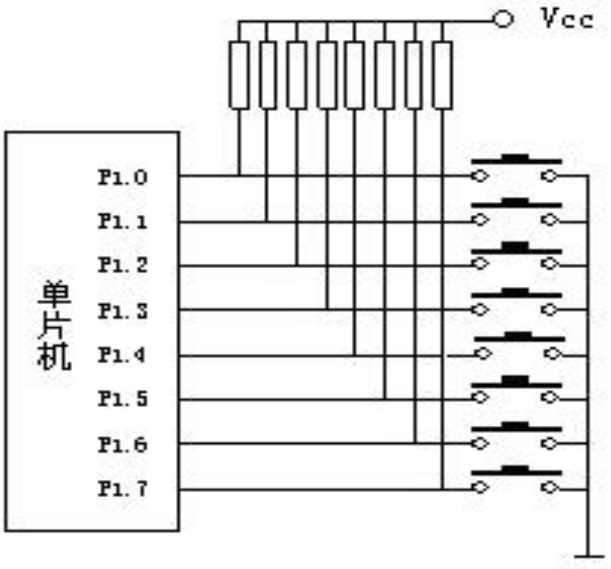
1. 认识常用按键开关



单片机应用系统中经常使用的按键开关

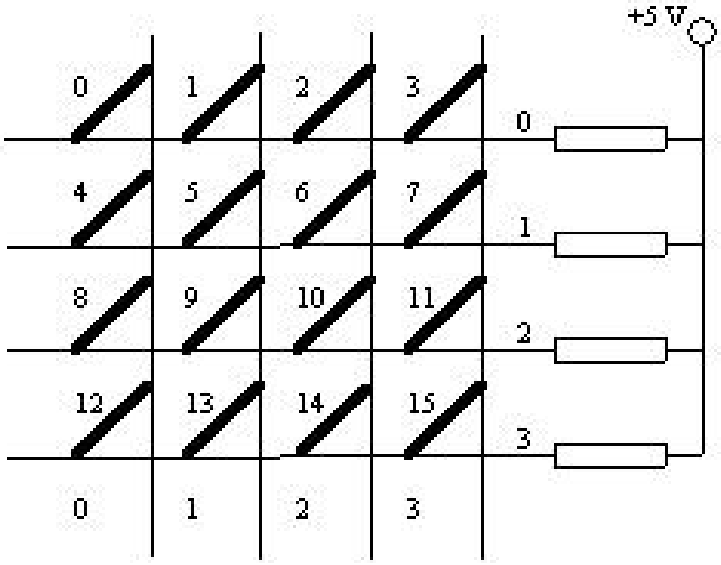
2. 独立式按键

独立式按键电路配置灵活，软件结构简单，但每个按键必须占用一根 I/O 口线，因此，在按键较多时，I/O 口线浪费较大，不宜采用。



3. 矩阵式按键

通常，矩阵式键盘的列线由单片机输出口控制，行线连接单片机的输入口。



4. 键盘编程扫描法识别按键一般应包括以下内容：

（1）判别有无键按下；

（2）键盘扫描取得闭合键的行、列号；

（3）用计算法或查表法得到键值；

（4）判断闭合键是否释放，如没释放则继续等待；

（5）将闭合键的键值保存，同时转去执行该闭合键的功能。

【任务实施】

1.搭建仿真电路图1。本任务使用P1口控制3个独立式按键，用P2口控制数码管显示；其中按键1（P1.0）控制数码管加一操作，按键2（P1.1）控制数码管减一操作，按键3（P1.2）控制数码管归0操作。

2.通过建立工程2-4-1，把以下程序代码放到Keil编译软件工具中，生成HEX文件，加载到仿真电路图中，看显示效果。

```
#include"reg51.h"
#define uint unsigned int
#define uchar unsigned char
sbit key2=P1^0;//独立式按键一
sbit key3=P1^1;//独立式按键二
sbit key4=P1^2;//独立式按键三
uchar code num[10]=
{0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};
delay(uint t)//延时函数
{
 uint i,j;
   for(i=0;i<t;i++)
       for(j=110;j>0;j--);//执行空语句
}
```

```
main()//主函数
{
    int n=0;
    P2=~num[0];//初始化显示 0
    while(1)
    {
        if(key1==0)//检测按键是否按下
        {
            while(key1==0);//去抖动
            {
            n++;//加一操作
            P2=~num[n];
            if(n==9)//加到 9 归 0
                n=0;
            }
        }
        if(key2==0)//检测按键是否按下
        {
            while(key2==0);//去抖动
            {
             n--;//减一操作
            P2=~num[n];
```

```
            if(n==0||n==-1)//减到 0 归 9
                n=9;
            }
        }
        if(key3==0)//检测按键是否按下
        {
            while(key3==0);//去抖动
            {
                n=0;//归 0 操作
                P2=~num[n];
            }
        }
    }
}
```

3.搭建仿真电路图 2。本任务使用 P1 口控制 4*4 矩阵式按键，用 P1 口控制矩阵式按键值的输入，其中前 4 位控制行选，后 4 位控制列；用 P2 口、P3 口各控制一个数码管的显示，显示对应按键值的键值，一次是 0-15。

4.通过建立工程 2-4-2，把以下程序代码放到 Keil 编译软件工具中，生成 HEX 文件，加载到仿真电路图中，看显示效果，比较工程 2-4-1 效果，进行比较分析。

```
#include"reg51.h"
```

```
#define uint unsigned int
#define uchar unsigned char
 uchar code num[10]=
  {0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};//
  uint keyget()//按键扫描并获取按键值
   {
   uint r,c;
   int n;
   n=r=c=0;
   P1=0xf0;
   switch(P1)//行按下检测
      {
      case 0xe0:r=1;break;//检测到第一行
      case 0xd0:r=2;break;//检测到第二行
      case 0xb0:r=3;break;//检测到第三行
      case 0x70:r=4;break;//检测到第四行
      default:break;
   }
   P1=0x0f;//行与列扫描值进行翻转
  switch(P1)//列按下检测
   {
      case 0x0e:c=1;break;//检测到第一列
```

```
    case 0x0d:c=2;break;//检测到第二列
    case 0x0b:c=3;break;//检测到第三列
     case 0x07:c=4;break;//检测到第四列
    default:break;
}
switch(r)//行和列交点的扫描检测
{
  case 1:if(c==1)n=1; if(c==2)n=2;
         if(c==3)n=3;if(c==4)n=4;break;
  case 2:if(c==1)n=5; if(c==2)n=6;
        if(c==3)n=7;if(c==4)n=8;break;
  case 3:if(c==1)n=9; if(c==2)n=10;
        if(c==3)n=11;if(c==4)n=12;break;
  case 4:if(c==1)n=13; if(c==2)n=14;
        if(c==3)n=15;if(c==4)n=16;break;
  default:break;
   }
return (n);//返回检测到的按键值
}
main()
{
uint a,b;
```

```
    while(1)
      {
      a=keyget()%10;//分离返回按键值的个位
      b=keyget()/10;//分离返回按键值的十位
      P2=~num[b];//显示十位
      P3=~num[a];//显示个位
      }
  }
```

【任务评价】

任务检测		分值	学生互评（40%）	老师评估（60%）	任务总评
任务知识内容	按键基本知识	20			
	电路图设计	20			
	程序设计	20			
	综合调试效果	20			
现场管理	出勤情况	5			
	实验室纪律	5			
	团队协作精神	5			
	保持实验室卫生	5			

【任务练习】

1.利用工程 2-4-1，修改程序代码，完成 3*3 矩阵式按键检测；

2.利用工程 2-4-2，修改程序代码，完成 4*3 矩阵式按键检测；

3.仿真成功后，将代码下载到试验箱继续调试。

【知识拓展】

1．按键分类

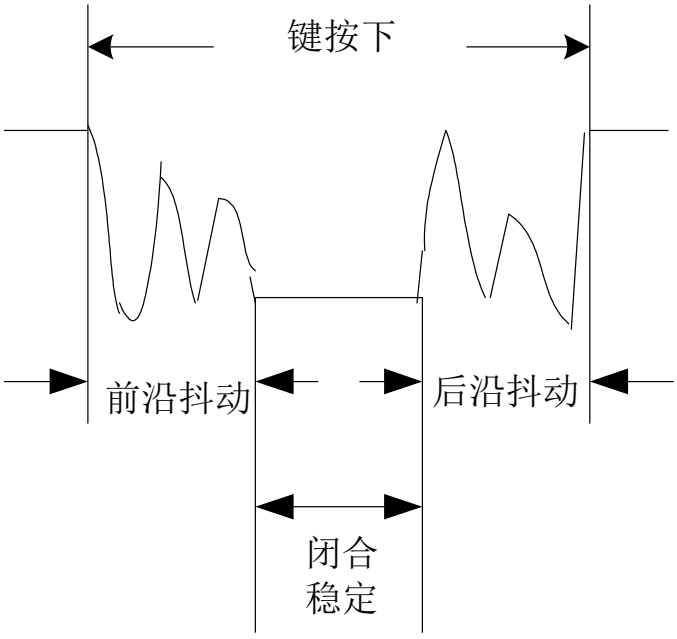
按键按照结构原理可分为两类，一类是触点式开关按键，如机械式开关、导电橡胶式开关等；另一类是无触点开关按键，如电气式按

键，磁感应按键等。前者造价低，后者寿命长。按键按照接口原理可分为编码键盘与非编码键盘两类。

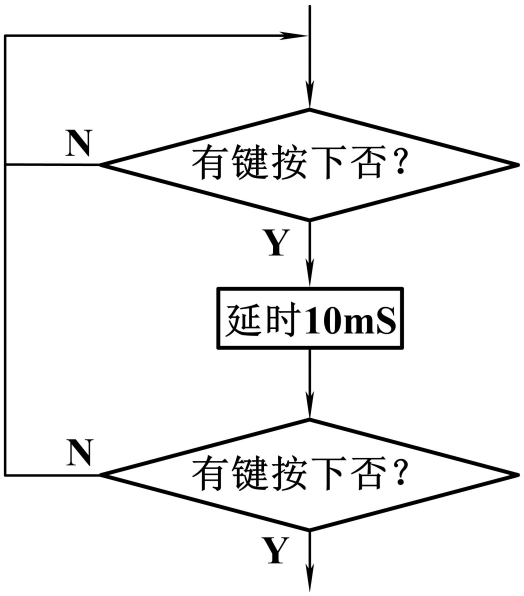
这两类键盘的主要区别是识别键符及给出相应键码的方法。编码键盘主要是用硬件来实现对按键的识别，硬件结构复杂；非编码键盘主要是由软件来实现按键的定义与识别，硬件结构简单，软件编程量大。这里将要介绍的独立式按键和矩阵式键盘都是非编码键盘。

2. 按键的去抖

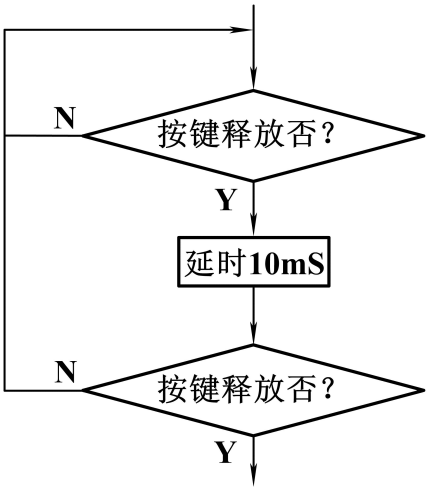
机械式按键在按下或释放时，由于机械弹性作用的影响，通常伴随有一定时间的触点机械抖动，然后其触点才稳定下来，抖动时间一般为 5-10ms，在触点抖动期间检测按键的通与断状态，可能导致判断出错。



3. 按键去抖流程图



（a）检测按键



（b）释放按键

【任务思考】

1.程序设计题

以下是 3*3 矩阵式按键检测按键值显示的程序代码，部分代码缺失，根据提示完成代码。实现效果。

```
/*3*3 矩阵式按键检测*/
#include"reg51.h"
#define uint unsigned int
#define uchar unsigned char
 uchar code num[10]=
   {0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};//
  uint keyget()//按键扫描并获取按键值
   {
   uint r,c;
   int n;
```

```
 n=r=c=0;
 P1=0xf8;//行检测初始化
 switch(P1)//行按下检测
    {
    case _____:r=1;break;//检测到第一行
    case _____:r=2;break;//检测到第二行
    case _____:r=3;break;//检测到第三行
    default:break;
 }
 P1=0x07;//列检测初始化
switch(P1)//列按下检测
 {
    case _____:c=1;break;//检测到第一列
    case _____:c=2;break;//检测到第二列
    case _____:c=3;break;//检测到第三列
       default:break;
 }
 switch(r)//行和列交点的扫描检测
 {
   case 1:if(c==1)n=1; if(c==2)n=2;if(c==3)n=3;break;
   case 2:if(c==1)n=4; if(c==2)n=5; if(c==3)n=6;break;
   case 3:if(c==1)n=7; if(c==2)n=8; if(c==3)n=9;break;
```

```
  default:break;
    }
return (n);//返回检测到的按键值
}
main()
{
 while(1)
    {
    P2=~num[____________];//显示按键值
    }
}
```

2.思考题

（1）按键去抖动，还有其它方式吗？

（2）如果按键多于 10 个时，应该怎么处理？

（3）采用矩阵法按键检测有什么好处，不利的地方在那？

3.技能提高

训练任务 1：独立设计一段代码，要求实现 3*5 矩阵式按键检测，电路如何连接？程序如何设计？

评价标准：流程图绘制、硬件电路原理图修改、软件程序修改、软硬件联调、实物连接。

训练任务 2：在任务四仿真电路图 2 的基础上，修改代码完成以

下功能，其中 1-9 按键按下显示按键值，第 10 个按键按下进行加一操作，第 11 个按键按下进行减一操作，第 12 个按键按下进行加 2 操作，第 13 个按键按下进行减 2 操作，第 14 个按键按下进行加 3 操作，第 15 个按键按下进行减 3 操作，第 16 个按键按下进行归 0 操作，电路如何连接？程序如何设计？

评价标准：硬件电路原理图修改、软件程序修改、软硬件联调、实物连接。

任务五 1602 液晶显示班级学号

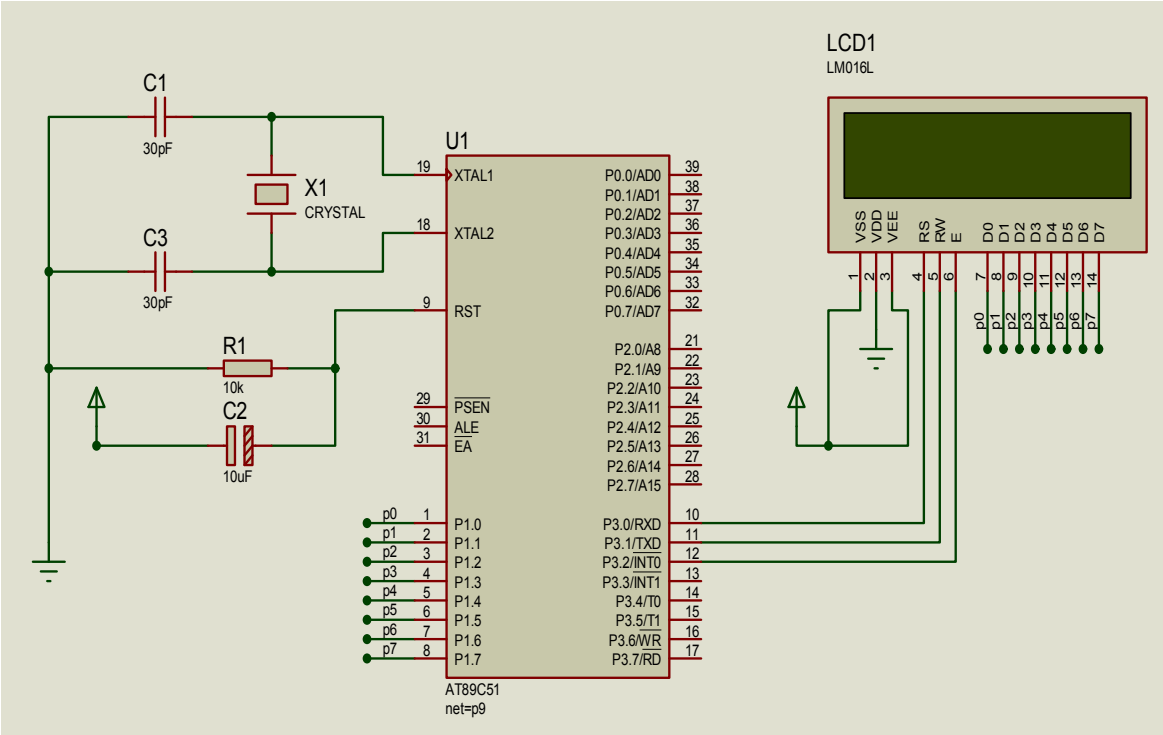
【任务目标】

通过将单片机与 1602 液晶连接，用液晶显示出”XIAO FEI ”“130100189”的设计与仿真演示，掌握单片机常用外部电路（LED 液晶显示）的设计方法，综合、灵活运用 C 语言进行编程。

【任务要求】

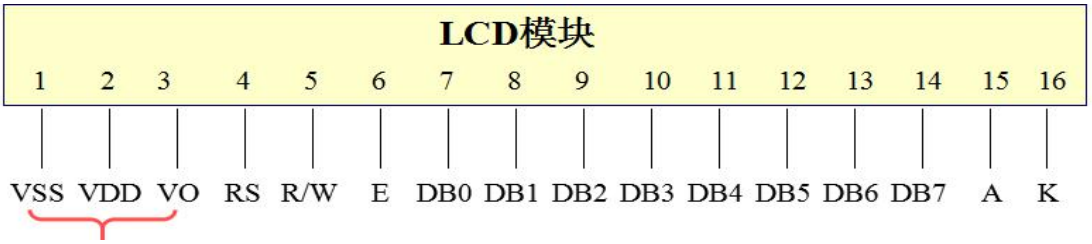
采用项目二任务 5 的仿真电路，根据任务咨询，建立工程 2-5，编写代码完成液晶显示出”XIAO FEI ”“130100189”的效果；

【仿真电路图】



【任务咨询】

1. 字符 LCD 液晶器件引脚



Vss：+5V 电源管脚(Vcc)

VDD：地管脚(GND)

Vo：液晶显示驱动电源(0V～5V)

DB0～DB7：数据线，可以用 8 位连接，也可以只用高 4 位连接，节约单片机资源，本实验中采用的是八位连接方法。

A：背光控制正电源

K：背光控制地

2.单片机与 LCD 模块之间的基本操作：

单片机与 LCD 模块之间有四种基本操作：写命令、读状态、写显示数据、读显示数据。

RS	R/W	操作
0	0	写命令操作（初始化、光标定位等）
0	1	读状态操作（读忙标志）
1	0	写数据操作（要显示的内容）
1	1	读数据操作（可以把显示存储区中的数据反读出来）

RS：数据和指令选择控制端，RS=0:命令/状态；RS=1:数据

R/W：读写控制线，R/W=0:写操作；R/W=1:读操作

E：数据读写操作控制位，E 线向 LCD 模块发送一个脉冲，LCD 模块与单片机之间将进行一次数据交换

3.写命令操作

LCD 上电时，都必须按照一定的时序对 LCD 进行初始化操作，主要任务是设置 LCD 的工作方式、显示状态、清屏、输入方式、光标位置等。

编号	指令名称	控制信号	命令字
			D7 D6 D5 D4 D3 D2 D1 D0
1	清屏	0 0	0 0 0 0 0 0 0 1
2	归home位	0 0	0 0 0 0 0 0 1 ×
3	输入方式设置	0 0	0 0 0 0 0 1 I/D S
4	显示状态设置	0 0	0 0 0 0 1 D C B
5	光标画面滚动	0 0	0 0 0 1 S/C R/L × ×
6	工作方式设置	0 0	0 0 1 DL N F × ×
7	CGRAM地址设置	0 0	0 1 A5 A4 A3 A2 A1 A0
8	DDRAM地址设置	0 0	1 A6 A5 A4 A3 A2 A1 A0
9	读BF和AC	0 1	BF AC6 AC5 AC4 AC3 AC2 AC1 AC0

4.初始化操作

LCD 液晶初始化操作主要有：工作方式设置、显示状态设置、清屏、输入方式设置等。

工作方式设置：001DL N F＊＊—设置单片机与 LCD 接口数据位数 DL、显示行数 N、字型 F。DL=1:8 位、DL=0:4 位； N=1:2 行、N=0:1 行；F=1:5×10、F=0:5×7

例：00111000B（38H）设置数据位数 8 位，2 行显示，5×7 点阵字符。

显示状态设置：00001DCB—设整体显示开关 D、光标开关 C、光标位的字符闪耀 B。

D=1:开显示；C=0:不显示光标；B=0:光标位字符不闪烁。

例：00001100B（0CH）打开 LCD 显示，光标不显示，光标位字符不闪烁。

清屏：清屏命令字 01H，将光标设置为第一行第一列。

输入方式设置：000001 I/D S—设光标移动方向并确定整体显示是否移动。

I/D=1:增量方式右移、I/D=0:减量方式左移；S=1:移位、S=0:不移位。

例：00000110B（06H）设置光标增量方式右移，显示字符不移动。

4.写数据操作

列 行	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
2	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF

注：表中命令字以十六进制形式给出，该命令字就是与 LCD 显示位置相对应的 DDRAM 地址。

【任务实施】

1.搭建仿真电路图。本任务使用 P1 口控制 LCD 的数据输入端，用 P3 口的前 3 位控制 LCD 液晶的命令控制端；

2.通过建立工程 2-5,把以下程序代码放到 Keil 编译软件工具中，

生成HEX文件，加载到仿真电路图中，看显示效果。

```
#include <reg51.h>    //包含52单片机头文件
#include <lcd1602.h> //包含LCD头文件
unsigned char x[] = "XIAO FEI";
unsigned char y[] = "130100189";
//-----------------------------------------//
#define uchar unsigned char
#define uint unsigned int
sbit LCD_RS  =P3^0;  //RS  1:DATA    0 :COMMAND
sbit LCD_RW  =P3^1; //R/W 1:READ  0 :WRITE
sbit LCD_E   =P3^2;    //E   1:ENABLE
#define LCD_ch   P1
//-----------------------------------------//
void delay(uint i)
{
    while(i--);
}
//**********写指令进入LCD1602**********//
void LCD_command()
{
   LCD_RS=0;
    LCD_RW=0;
```

```
 LCD_E=0;     delay(200);  //延时大约 2ms
 LCD_E=1;
}
//********把数据写入 LCD1602*****************//
void LCD_data()
{
    LCD_RS=1;
     LCD_RW=0;
    LCD_E=0;  delay(200);
    LCD_E=1;
}
//-----------------------------------------//
void Init_LCD(void)  /*初始化液晶*/
{
    LCD_ch=0x01;      //清屏
  LCD_command();
    LCD_ch=0x38;      //8 位数据，两行显示，5*7 点阵
  LCD_command();
  LCD_ch=0x0c;      //开显示，关光标，关闪烁
    LCD_command();
    LCD_ch=0x06;       //读写数据后 AC 自动增一，画面不动
  LCD_command();
```

```
}
/***********将数据 ch 显示在第 i 行第 j 列******/
void LCD_dis(uchar i,uchar j,uchar ch)
{
  uchar    addr;
 if(i==0) addr = 0x80+j;   //设置为第一行
  else      addr = 0xc0+j;    //设置为第二行
    LCD_ch=addr;
 LCD_command(); //先写地址
 LCD_ch=ch;
 LCD_data();           //后送数据
}
void main()                //主函数
{
 unsigned int  i,j;
 Init_LCD();         //首先初始化各数据
 while(1)
 {
    for(i=0;i<36;i++)
    {
     //LCD_dis(1, i, 0x30+i); //LCD_dis(0, i, i+’0’);
      LCD_dis(1, i, i+’A’);
```

```
        LCD_dis(1, i,x[i]); //显示数组内容
          delay(5000);
         }
       for(j=0;j<10;j++)
         {
       LCD_dis(0,j,y[j]);
       delay(5000);}
         }
   }
```

【任务评价】

任务检测		分值	学生互评（40%）	老师评估（60%）	任务总评
任务知识内容	LED 液晶基本知识	20			
	电路图设计	20			
	程序设计	20			
	综合调试效果	20			
现场管理	出勤情况	5			
	实验室纪律	5			
	团队协作精神	5			
	保持实验室卫生	5			

【任务练习】

1.利用工程 2-5，修改程序代码，实现字符“I LOVE CHANA”移动的显示效果；

2.仿真成功后，将代码下载到试验箱继续调试。

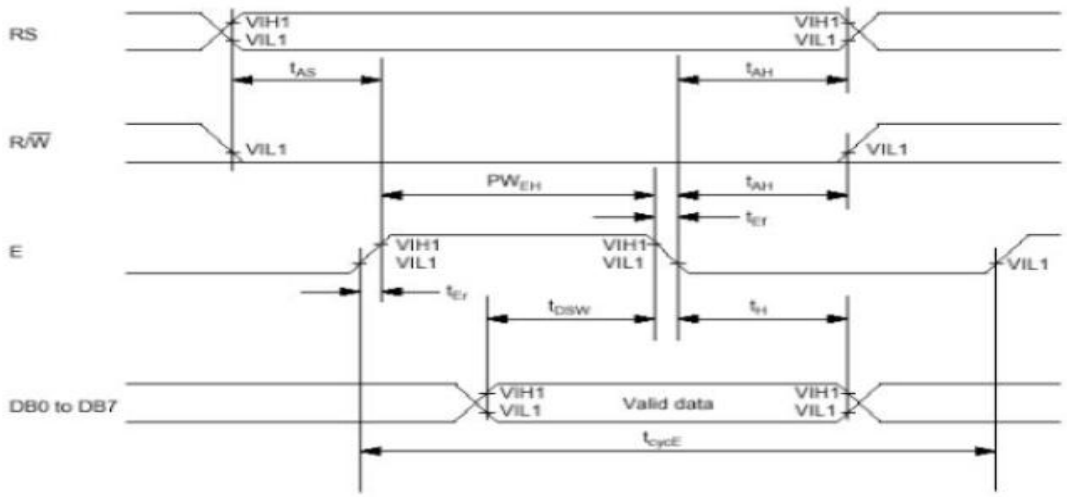
【知识拓展】

1．认识 162 液晶显示的原理

引脚号	符号	状态	功　能
1	Vss		电源地
2	Vdd		+5V逻辑电源
3	V0		液晶驱动电源
4	RS	输入	寄存器选择 1：数据；0：指令
5	R/W	输入	读、写操作选择 1：读；0：写
6	E	输入	使能信号（MDLS40466未用，符号NC）
7	DB0	三态	数据总线（LSB）
8	DB1	三态	数据总线
9	DB2	三态	数据总线
10	DB3	三态	数据总线
11	DB4	三态	数据总线
12	DB5	三态	数据总线
13	DB6	三态	数据总线
14	DB7	三态	数据总线（MSB）
*15	E1	输入	MDLS40466上两行使能信号
*16	E2	输入	MDLS40466下两行使能信号

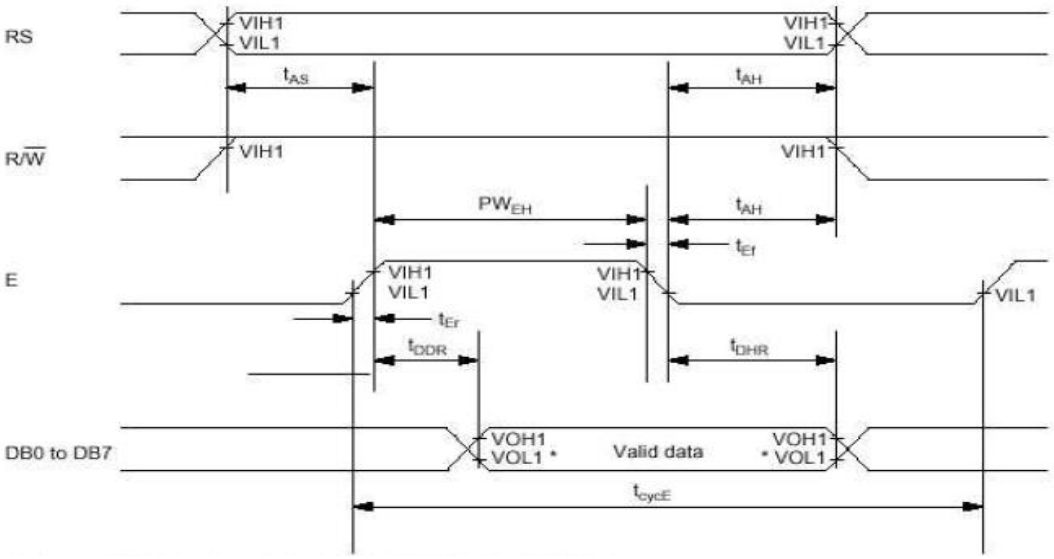
注；15、16两管脚仅用于MDLS40466，其余型号不用或为LED背光电源输入。

2. 写操作



项　目	符号	最小值	最大值	单位
使能周期	TcycE	1000	—	ns
使能脉冲宽度	Pweh	450	—	ns
使能升、降时间	Ter,Tef	—	25	ns
地址建立时间	Tas	140	—	ns
地址保持时间	Tah	10	—	ns
数据建立时间	Tdsw	195	—	ns
数据保持时间	Th	10	—	ns

3. 读操作



项　目	符号	最小值	最大值	单位
使能周期	TcycE	1000	—	ns
使能脉冲宽度	Pweh	450	—	ns
使能升、降时间	Ter,Tef	—	25	ns
地址建立时间	Tas	140	—	ns
地址保持时间	Tah	10	—	ns
数据延迟时间	Tddr	—	320	ns
数据保持时间	Tdhr	10	—	ns

4. 信号真值表

RS	R/W	E	功 能
0	0	下降沿	写指令代码
0	1	高电平	读忙标志和AC值
1	0	下降沿	写数据
1	1	高电平	读数据

1、清屏

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

运行时间（250Khz）：1.64 μs

功能：清DDRAM和AC值。

2、归位

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

运行时间（250Khz）：1.64 μs

功能：AC=0，光标、画面回HOME位。

3、输入方式设置

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

运行时间（250Khz）：40 μs

功能：设置光标、画面移动方式。

其中：I/D=1：数据读、写操作后，AC自动增一；

I/D=0：数据读、写操作后，AC自动减一；

S=1：数据读、写操作，画面平移；

S=0：数据读、写操作，画面不动。

4、显示开关控制

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

运行时间（250Khz）：40 μs

功能：设置显示、光标及闪烁开、关。

其中：D表示显示开关：D=1为开，D=0为关；

C表示光标开关：C=1为开，C=0为关；

B表示闪烁开关：B=1为开，B=0为关。

5、光标、画面位移

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

运行时间（250Khz）：40 μs
功能：光标、画面移动，不影响DDRAM。
其中：S/C=1：画面平移一个字符位；
S/C=0：光标平移一个字符位；
R/L=1：右移；R/L=0：左移。

6、功能设置

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

运行时间（250Khz）：40 μs
功能：工作方式设置（初始化指令）。
其中：DL=1，8位数据接口；DL=0，4位数据接口；
N=1，两行显示；N=0，一行显示；
F=1，5×10点阵字符；F=0，5×7点阵字符。

7、CGRAM地址设置

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	A5	A4	A3	A2	A1	A0

运行时间（250Khz）：40 μs
功能：设置CGRAM地址。A5～A0=0～3FH。

【任务思考】

1.设计题

使字符在按键没有松动的情况下连续移动； 实现字符的循环移动，即当字符串移动到边界时仍可以移动，显示不完的部分从另一边显示出来。

2.思考题

（1）怎么通过键盘控制Hello或者中文字符在LCD上左右、上下移动？

（2）怎么使用字符生成软件来实现任意字符代码的生成和显示？

3.技能提高

训练任务1：独立设计一段代码，自定义一些字符、图形并显示出来，方案如何设计？

评价标准：流程图绘制、硬件电路原理图修改、软件程序修改、软硬件联调、实物连接。

训练任务2：查阅详细的技术资料，练习一些扩展指令的使用，能够显示文本、数据，并能够实时更新，电路如何连接？程序如何设计？

评价标准：硬件电路原理图修改、软件程序修改、软硬件联调、实物连接。