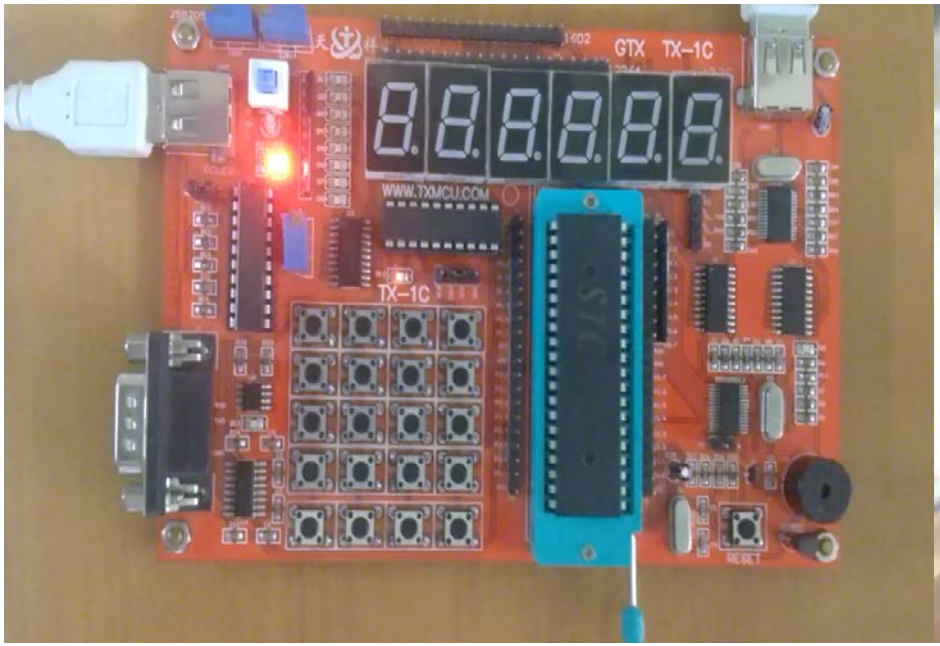
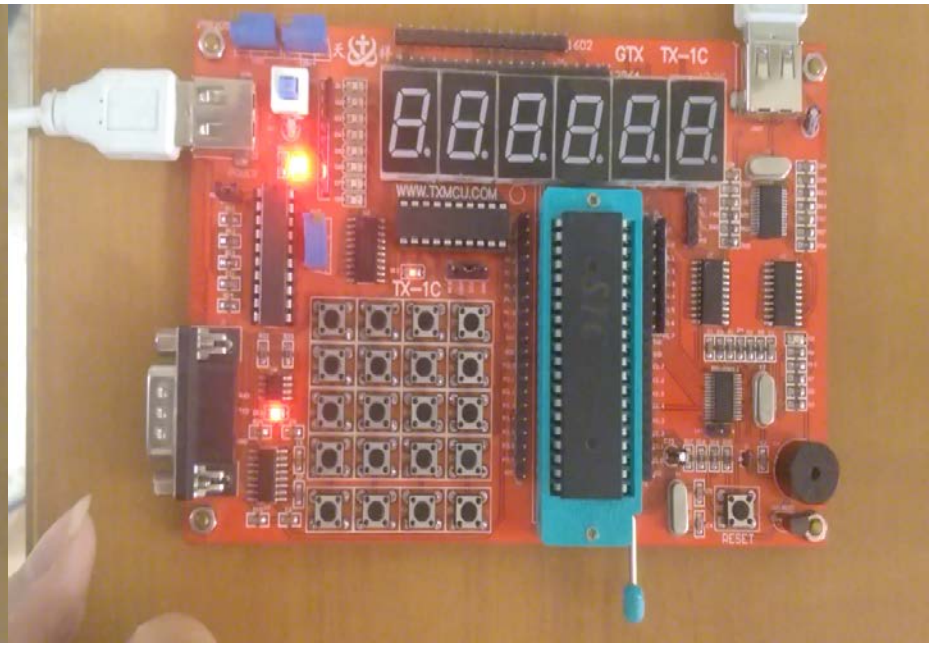
单片机原理与应用

北京航空航天大学
电子信息工程学院
张玉玺
zhangyuxi@buaa.edu.cn

第五章 按键控制

- 本章实现功能
  - 单片机独立按键控制一个LED亮灭
  - 单片机4×4矩阵键盘控制数码管从0～F





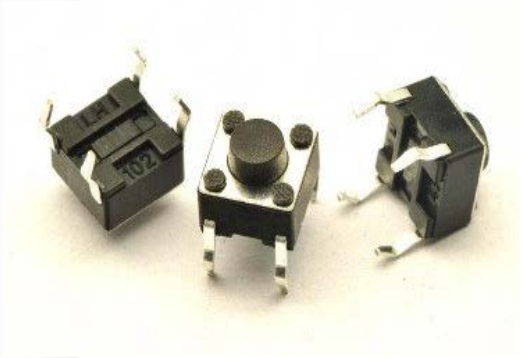
第五章 按键控制

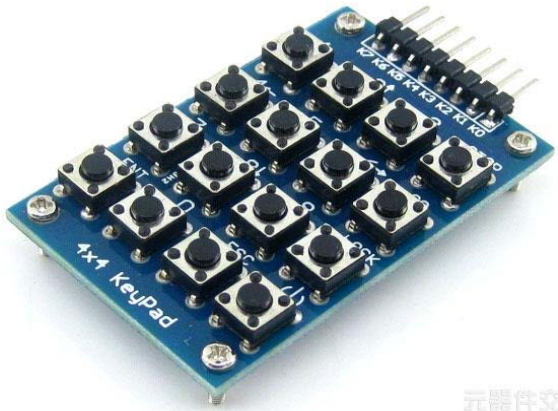
- 如何实现功能
  - 按键检测原理
  - I/O寄存器控制
  - 独立按键
    - 流程图
    - 程序代码、烧写
  - 矩阵键盘
    - 流程图
    - 程序代码、烧写

第五章 按键控制

■ 按键检测原理

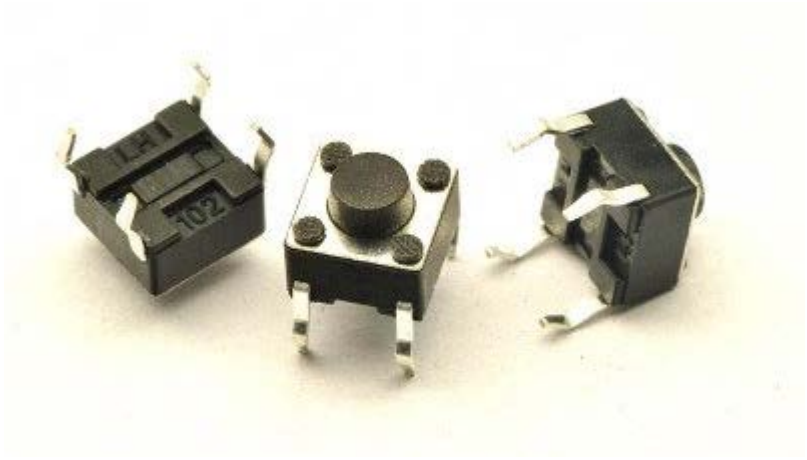
- ➢ 按键作为输入设备，机械弹性开关。
- ➢ 按键按下，线路导通，两端电平相等
- ➢ 两种连接方式
- ➢ 消抖的必要性

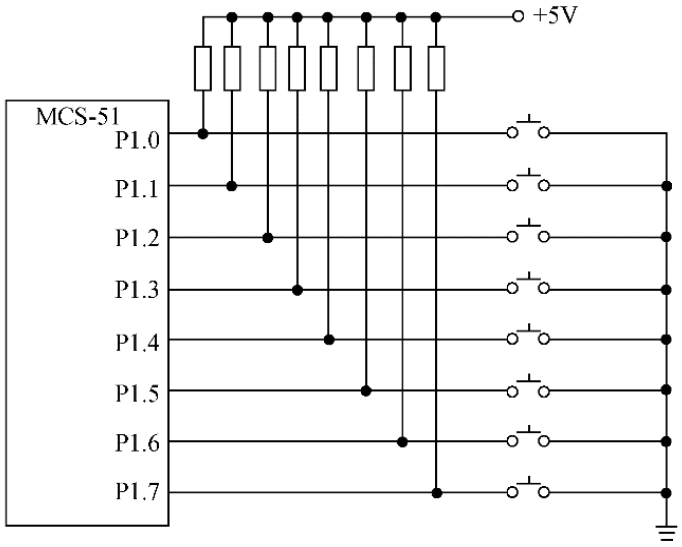




第五章 按键控制

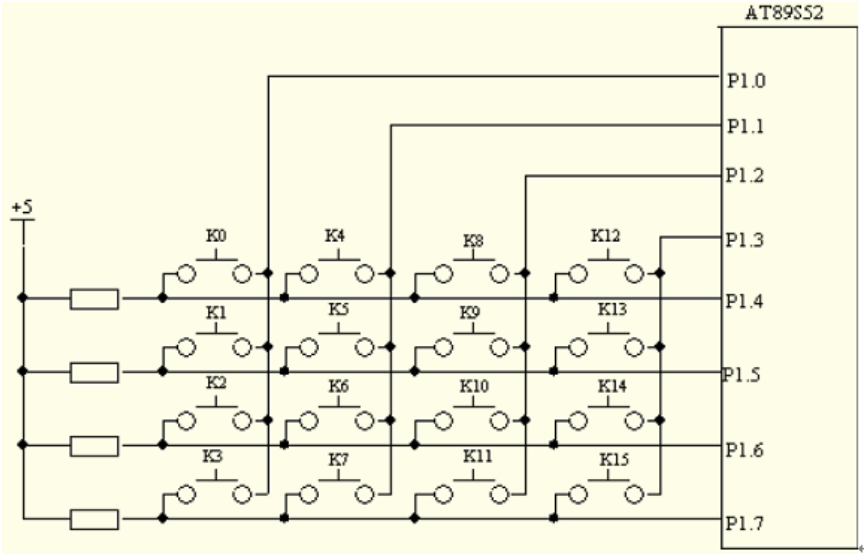
- 两种连接方式
  - 独立按键
    - 一端接地，一端接单片机I/O
    - 读取I/O端是否低电平
    - 连接简单
    - 适用于按键数量少，单片机空余引脚多的场合





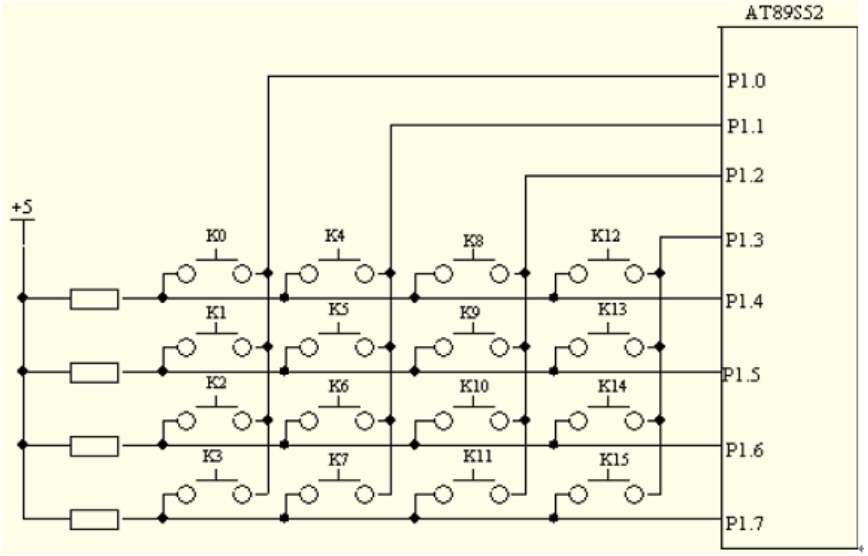
第五章 按键控制

- 两种连接方式
  - 矩阵键盘
    - 两端均接单片机I/O
    - 读取两端电平是否相等
    - 连接复杂
    - 适用于按键较多的场合，这种连接方法大大节省了单片机引脚



第五章 按键控制

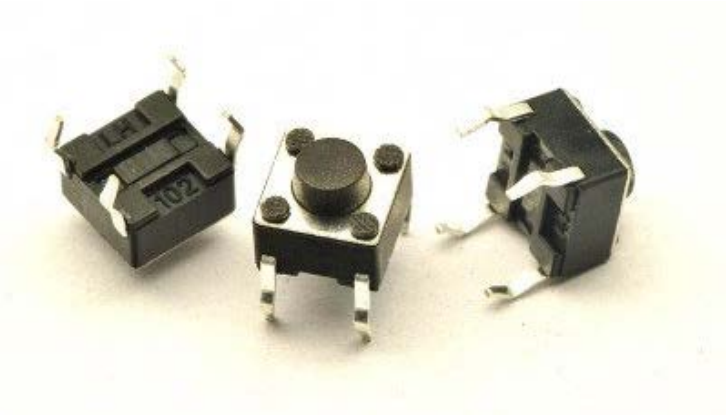
- 两种连接方式
  - 矩阵键盘
    - 先将一行置低电平，依次检测各列是否有低电平
      （列扫描）
    - 轮流将各行置低电平，依次检测各列是否有低电平
      （行扫描）
    - 也可以先行扫描再列扫描

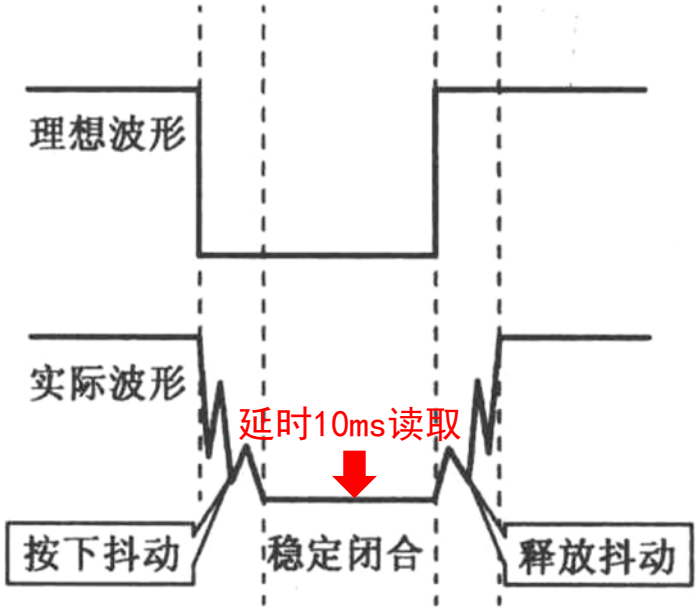


第五章 按键控制

- 为什么要消抖？
  - 波形是非理想的！

由于机械抖动，单片机的引脚会检测到多次按键闭合，从而出现一次按键多次响应的现象。



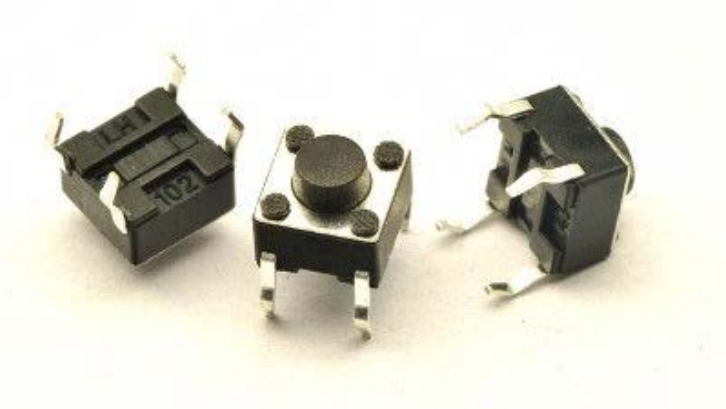


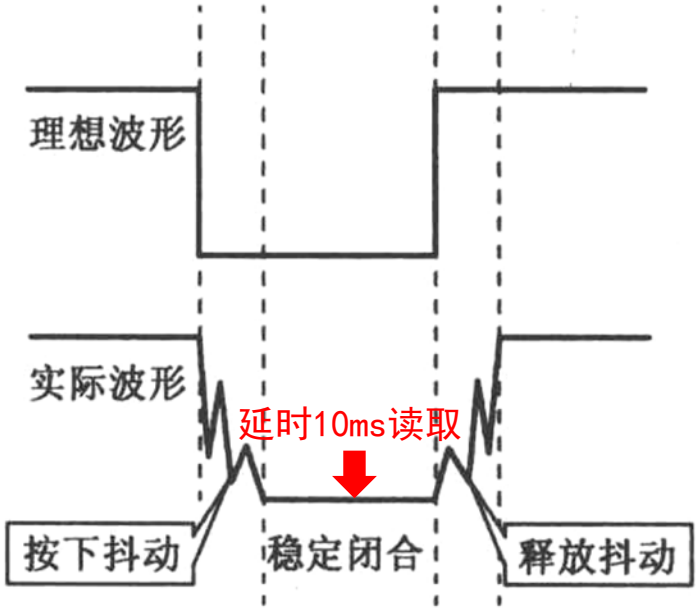
- 怎么消抖？
  - 硬件消抖：去抖电路，防抖动装置…
  - 软件消抖：延时检测

第五章 按键控制

- 延时程序
  - 通过for循环
    - 精度要求不高的场合
  - 通过定时器计数
    - 精度要求高的场合

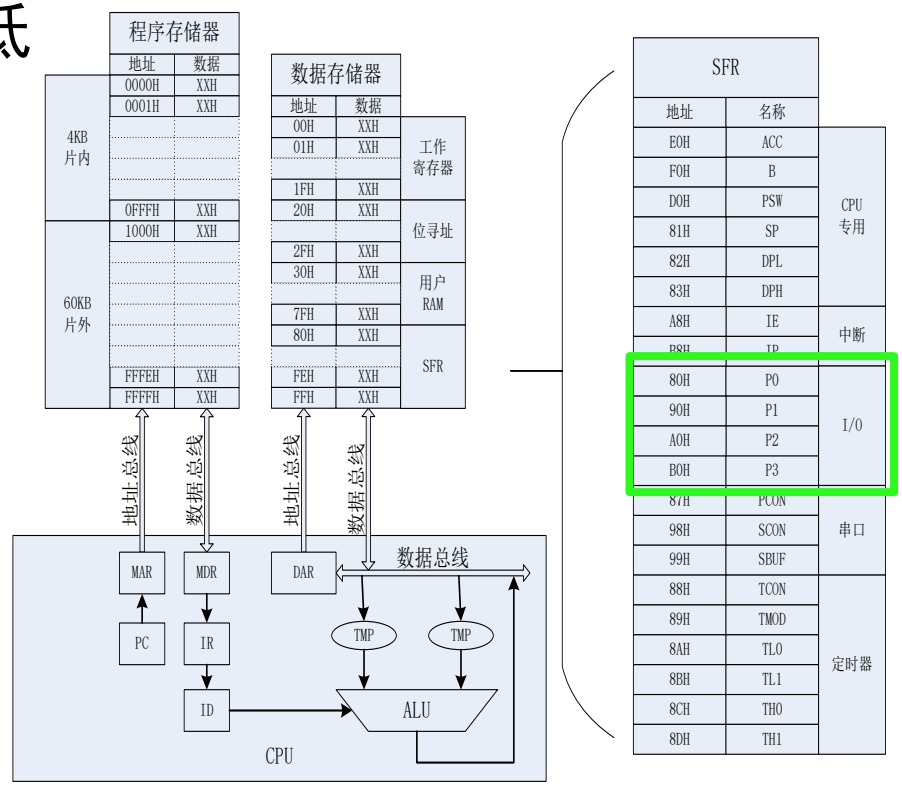
```
void delayms(unsigned int z)
{
  unsigned int i,j;
  for(i=z;i>0;i--)
    for(j=110;j>0;j--);
}
```





第五章 按键控制

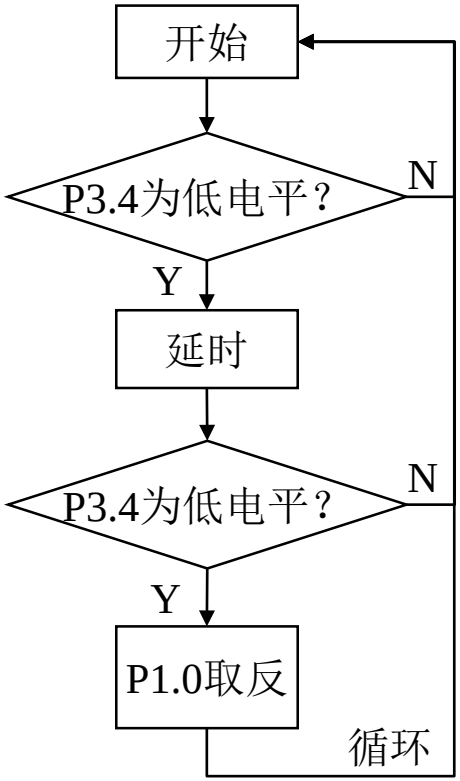
- I/O寄存器控制
  - 控制单片机I/O可读入高低电平，完成对按键的判断
  - 通过I/O寄存器可完成对I/O口控制
  - 单片机4个I/O寄存器地址
    - P0:80H
    - P1:90H
    - P2:A0H
    - P3:B0H
  - 支持段寻址和位寻址

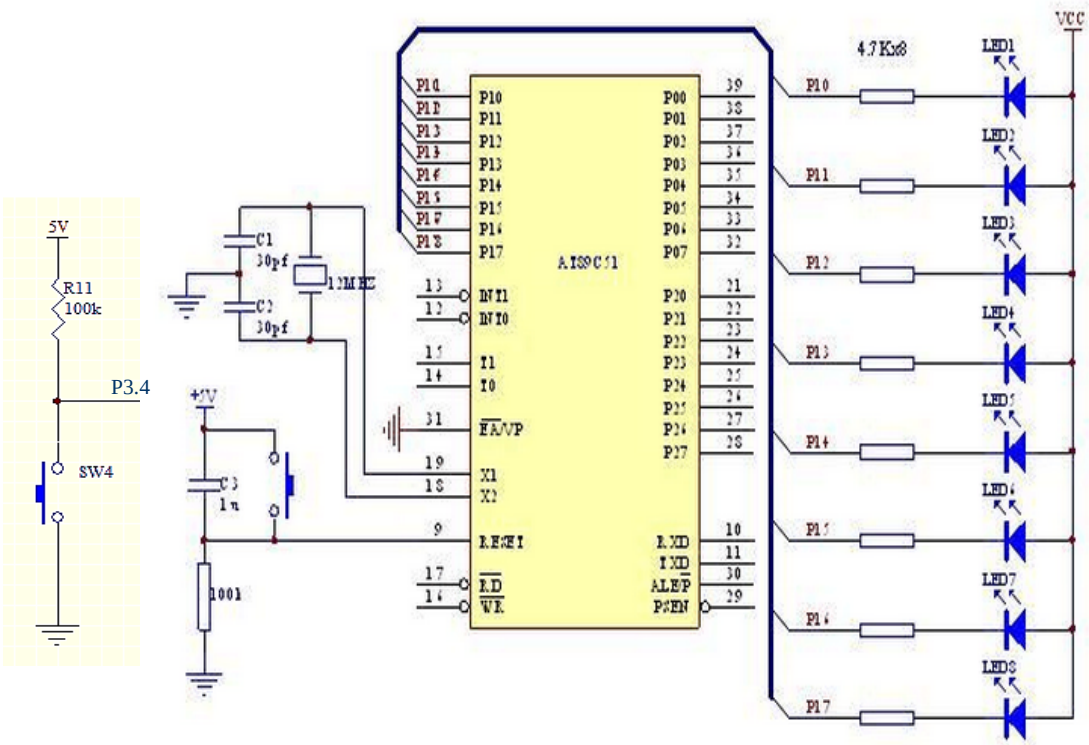


第五章 按键控制

流程图

独立按键



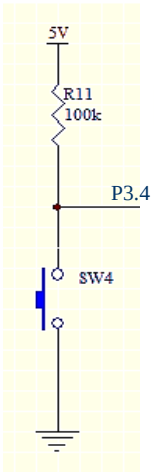


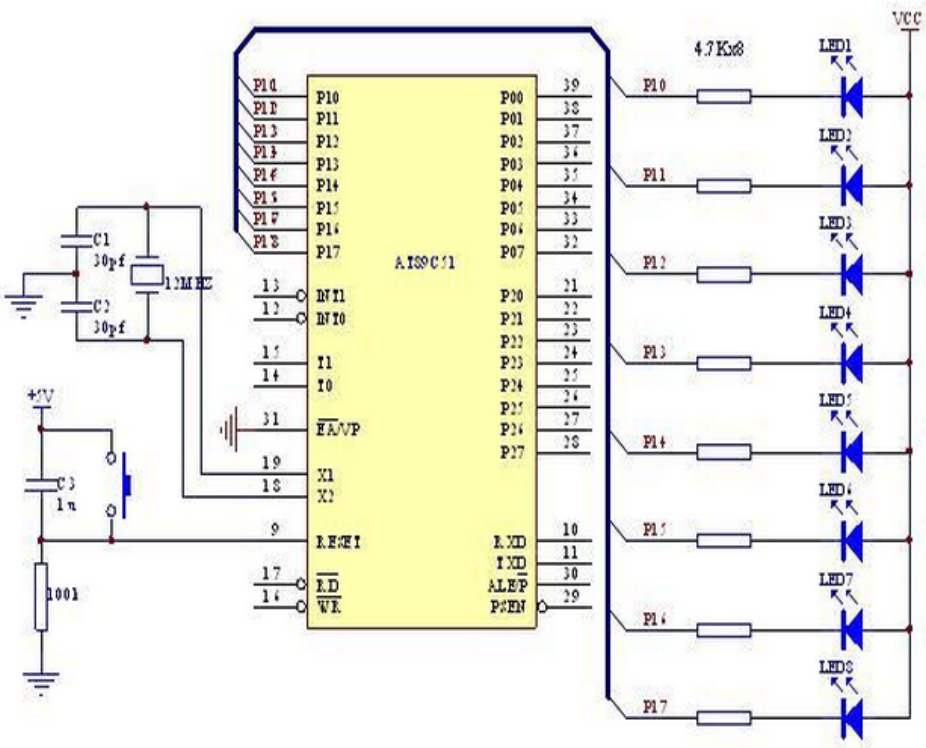
第五章 按键控制

程序代码

独立按键

```
#include <reg51.h>
sbit led0 = P1^0;
sbit key0 = P3^4;
void main()
{
  while(1)
   {
     key0 = 1; //置1为输入状态
     if(!key0)
     { //延时消抖
       delayms(10);
       if(!key0)
           led0 = ~led0;
       while(!key0); //等待释放
     }
   }
}
```

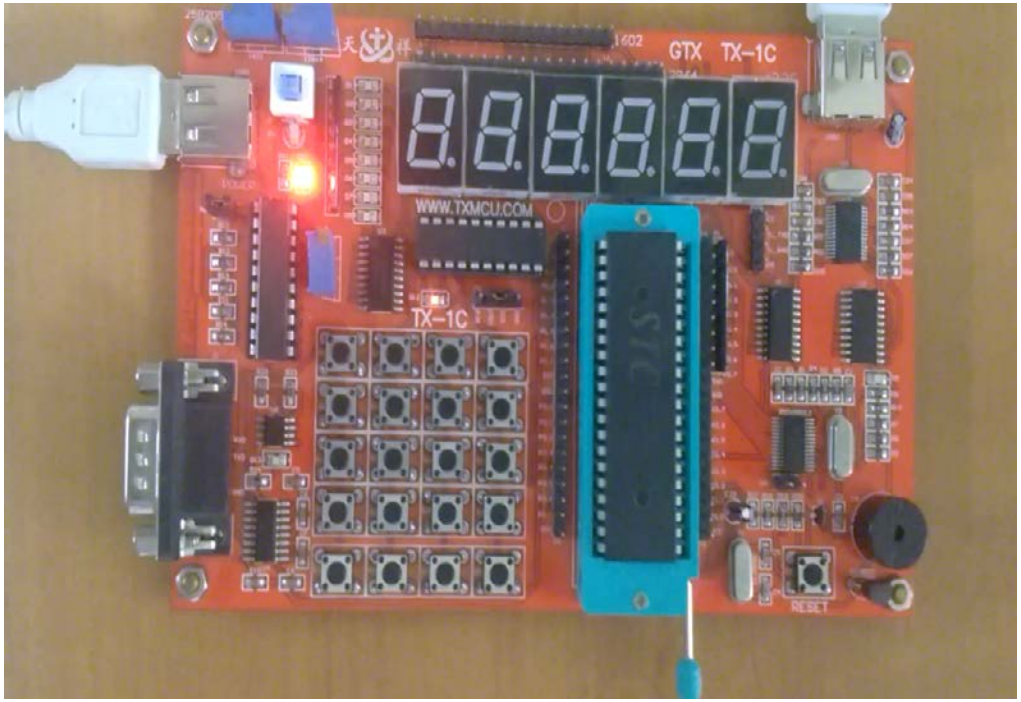


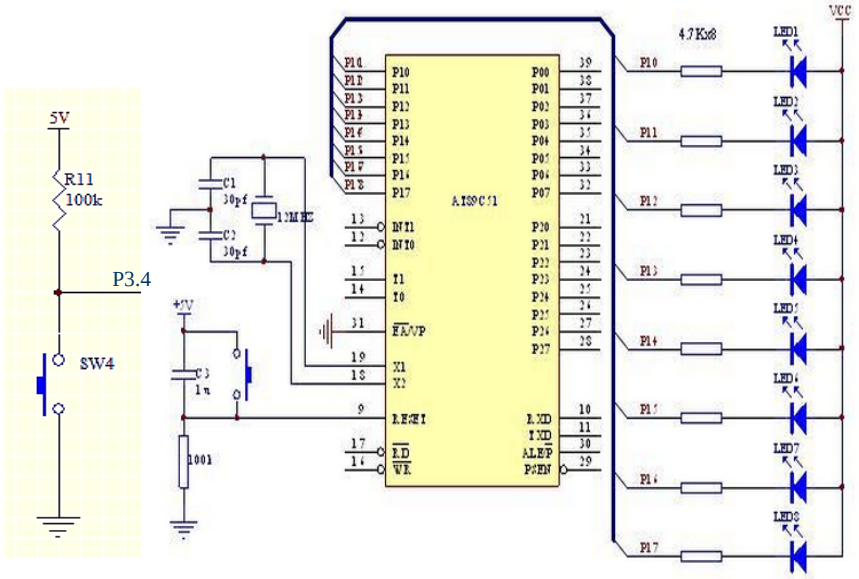


第五章 按键控制

```
#include <reg51.h>
sbit led0 = P1^0;
sbit key0 = P3^4;
void main()
{
  while(1)
  {
    key = 1; //置1为输入状态
    if(!key0)
    { //延时消抖
      delayms(10);
      if(!key0)
         led0 = ~led0;
      while(!key0); //等待释放
    }
  }
}
```

- 烧写
  - 观察现象：

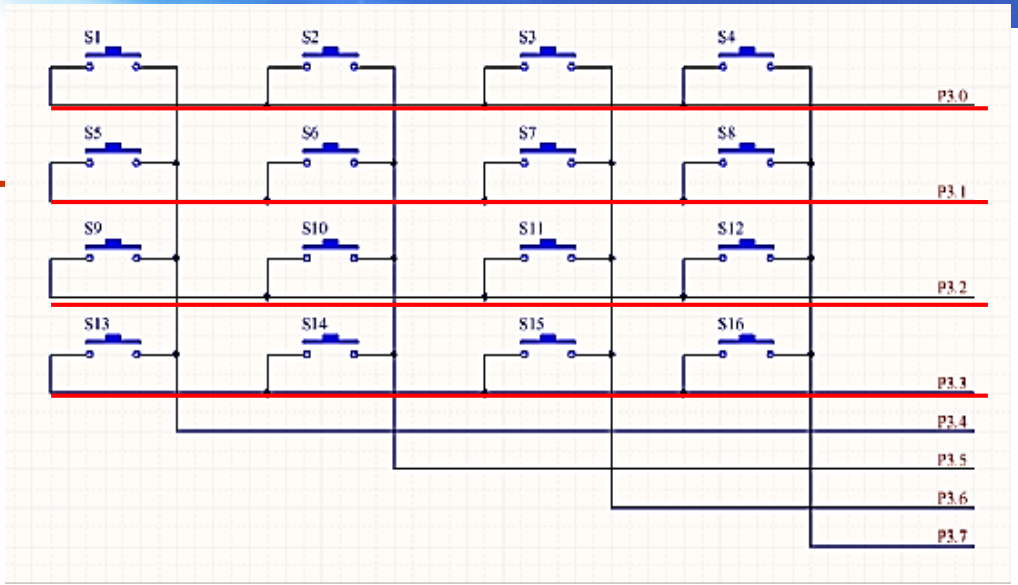


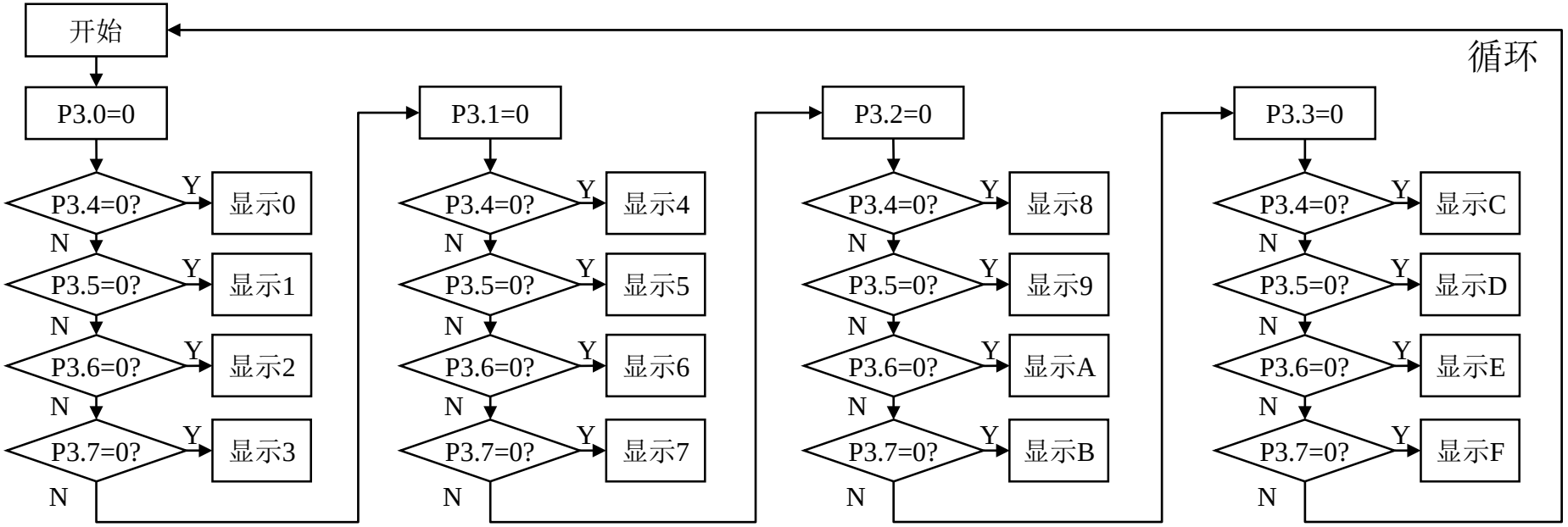


第五章 按键控制

流程图

矩阵键盘



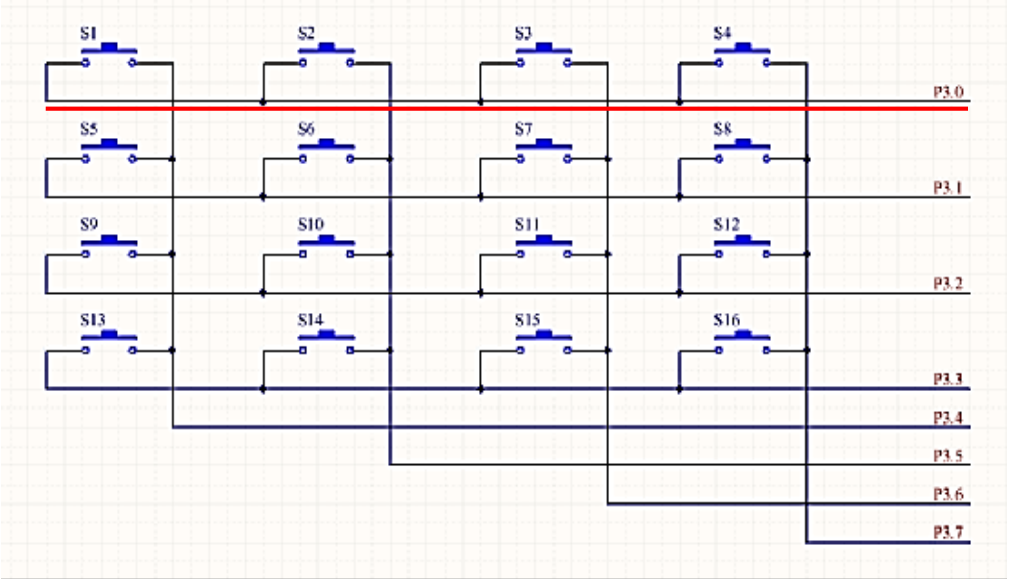


第五章 按键控制

程序代码

矩阵键盘

```
//矩阵键盘扫描函数(部分)
void matrixkeyscan( )
{
     P3=0xfe;   //将第一行线置低电平
     temp=P3;   //读取P3口状态
     temp=temp&0xf0;
     if(temp!=0xf0) //延时消抖
     {
       delayms(10);
       temp=P3;   //重新读P3口数据
       temp=temp&0xf0;
       if(temp!=0xf0)
       {
              temp=P3;
              …
```

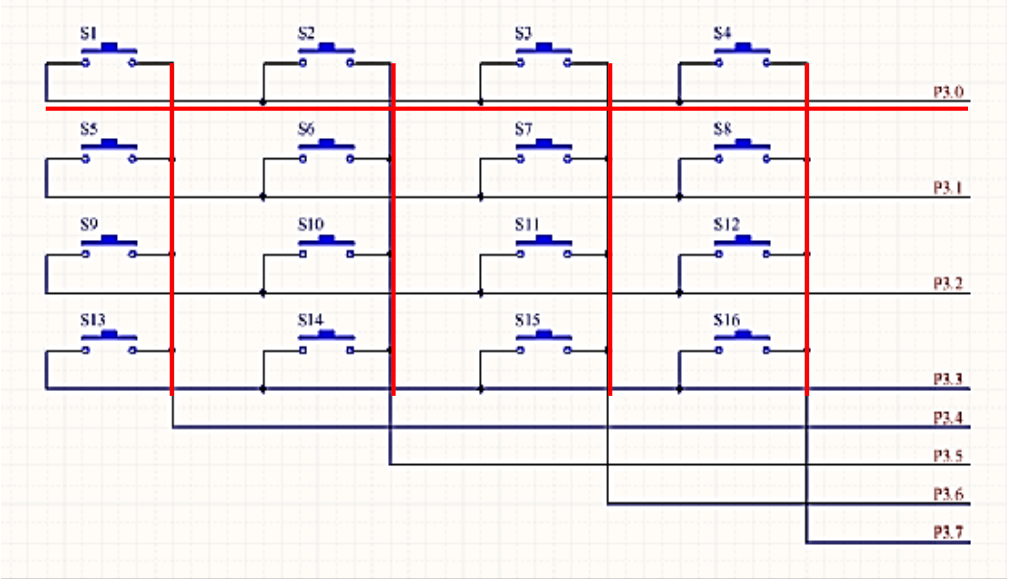


第五章 按键控制

程序代码

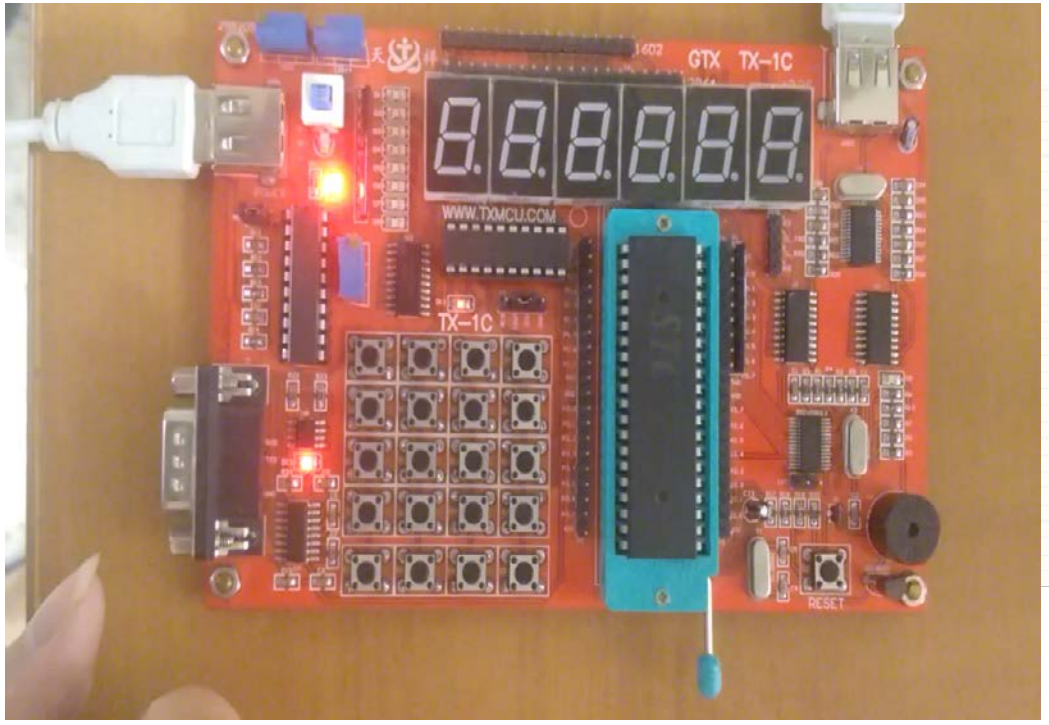
矩阵键盘

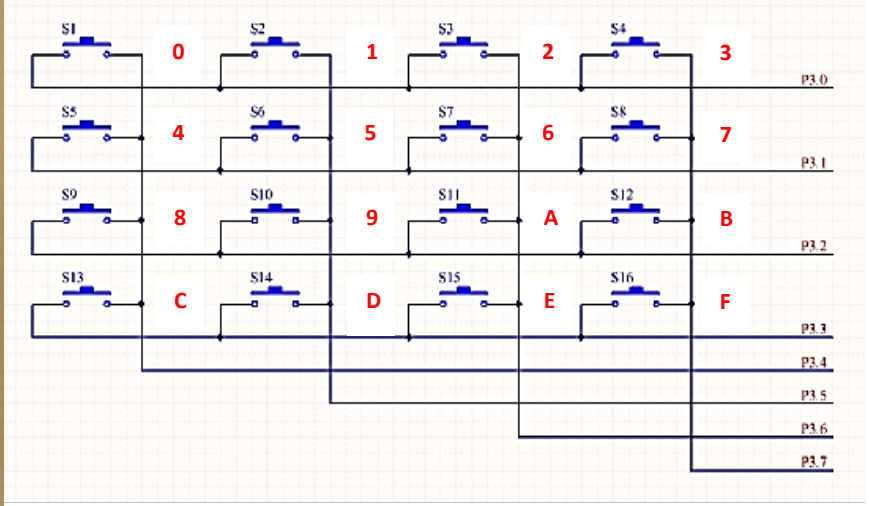
```
…
switch(temp) //判定按键位置
{    case 0xee:   //第一行第一个按键
     display(1);
     break;
     case 0xde:   //第一行第二个按键
     display(2);
     break;
     case 0xbe:   //第一行第三个按键
     display(3);
     break;
     case 0x7e:   //第一行第四个按键
     display(4);
     break;   }
}
…
```



第五章 按键控制

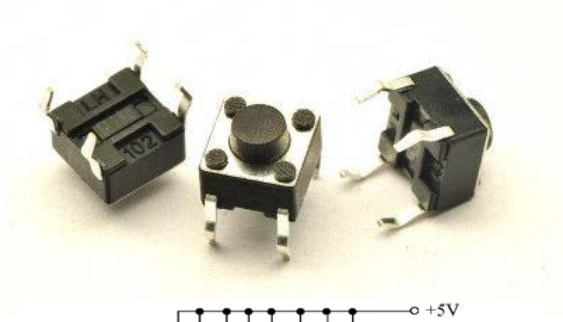
- 烧写
  - 观察现象：

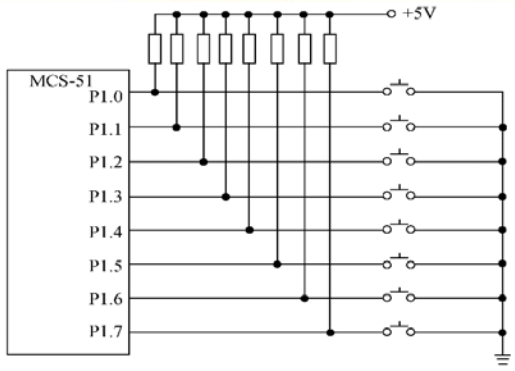


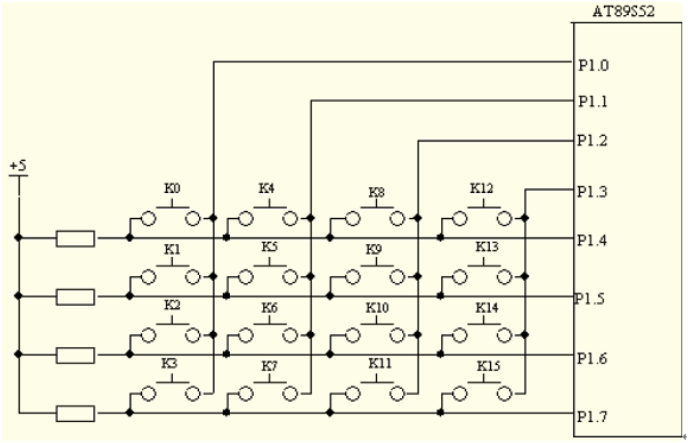


小结

- 按键是输入设备
  - 独立按键
  - 矩阵键盘（扫描原理）
  - 延时消抖原理
- I/O寄存器P0~P3
  - 控制I/O口读取高低电平







再见