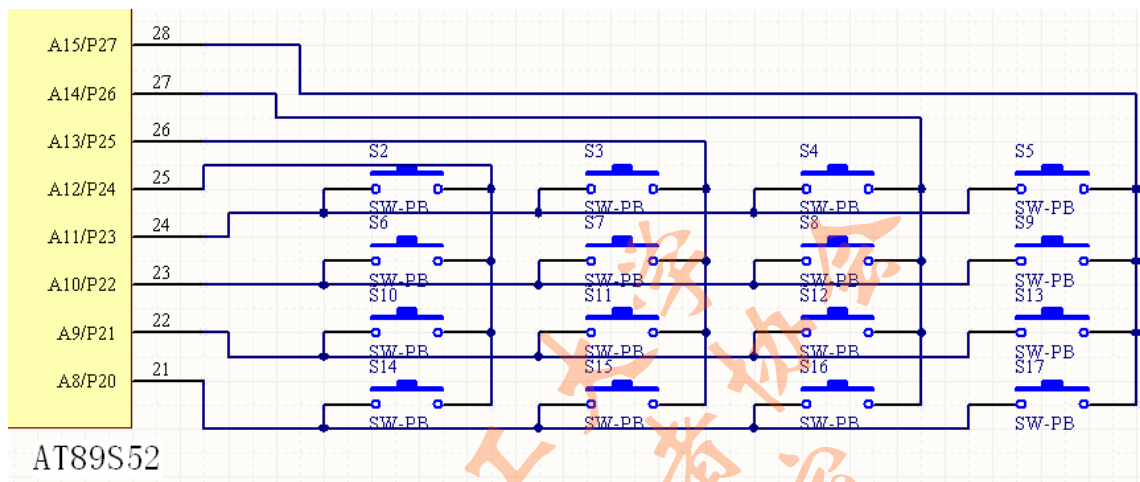


## 实验六:矩阵键盘实验

### 一. 实验目的:

- 1, 掌握用查表法给按键解码的方法。
- 2, 熟悉数码管显示的应用

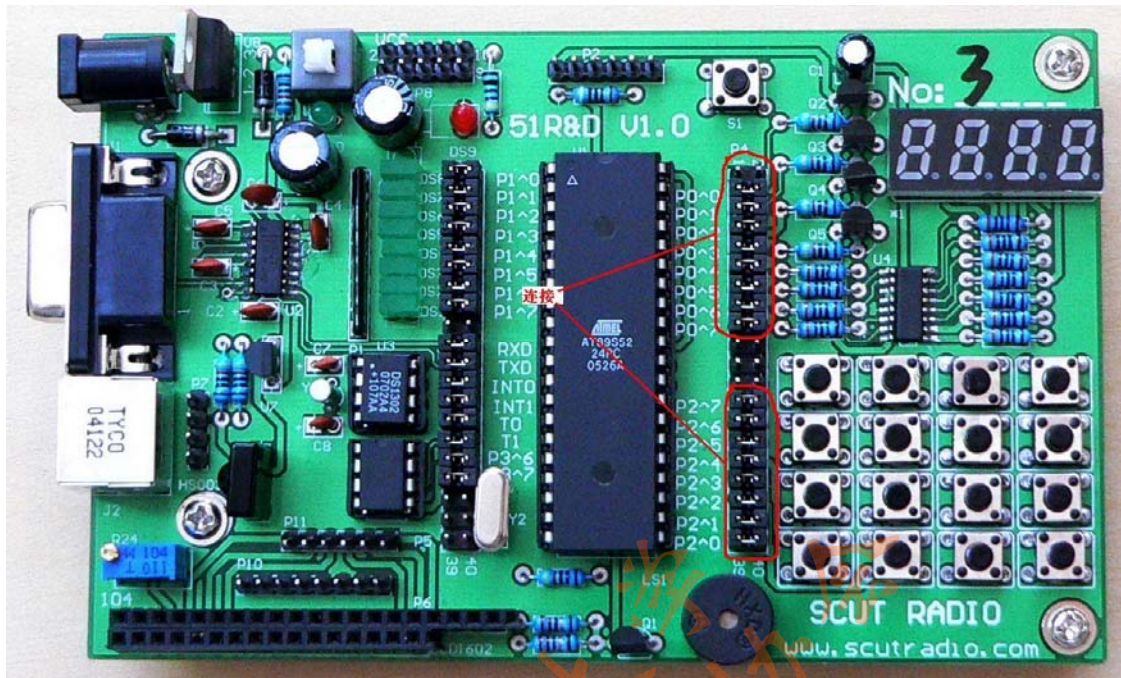
### 二. 实验原理:



4\*4键盘原理图

根据以上的电路图,若有一个按键被按下,则单片机的p2口就有两个io连在一起。若令其中一个为高电平,另一个为低电平,那么高电平的那个io也会被下拉到低电平。我们先令p2 io口低四位为高电平,高四位为低电平,若有按键按下,则低四位必有一个io口被拉到低电平,只要找出那个io就能确定按键在第几行,若令口低四位为低电平,高四位为高电平,若有按键按下,则高四位必有一个io口被拉到低电平,只要找出那个io就能确定按键在第几列,这样就能确定按键的行列值,即按键所在的位置。为了提高处理速度,我们直接先令p2 io口低四位为高电平,高四位为低电平,读一下p2口的值,记下那个值,再令口低四位为低电平,高四位为高电平,再读一下p2口,根据两次结果进行查表,若表中有数据与这两个结果匹配,便能直接从表中得到按键值。

### 三. 硬件连接图:



将下图中画圈的16个双排针用排帽短接即可。

#### 四. 实验代码:

```
#include<AT89X52.H>
#include"led.h" //该头文件包含数码管驱动动
/*****
    按键扫描函数
    若有按键按下：返回 键值（1-16）；
    若没按键按下：返回 255
*****/
unsigned char getch ( )
{
    unsigned char X=0, Y=0, Z=0, k=255;
    P2=0xff;
    P2=0x0f; //先对P2置数 行扫描
    if(P2!=0x0f) //判断是否有键按下
    {
        //延时,软件去干扰
        if(P2!=0x0f) //确认按键按下X = P2;
        {
            X=P2; //保存行扫描时有键按下时状态
            P2=0xf0; //列扫描
            Y=P2; //保存列扫描时有键按下时状态
            Z=X|Y; //取出键值
            switch ( Z ) //判断键值（那一个键按下）
            {
                case 0xee: k=0; break; //对键值赋值
                case 0xde: k=1; break;
            }
        }
    }
}
```

```
        case 0xbe: k=2; break;
        case 0x7e: k=3; break;
        case 0xed: k=4; break;
        case 0xdd: k=6; break;
        case 0xbd: k=7; break;
        case 0x7d: k=8; break;
        case 0xeb: k=9; break;
        case 0xdb: k=7; break;
        case 0xbb: k=10; break;
        case 0x7b: k=11; break;
        case 0xe7: k=12; break;
        case 0xd7: k=13; break;
        case 0xb7: k=14; break;
        case 0x77: k=15; break;
    }
    while(P2!=0xf0);
}
}
return k;
}
void main ()
{
    unsigned char temp=0, key=255;
    while (1)
    {
        key=getch(); //扫描
        if (key!=255) //有按键按下
        {
            temp=key;
        }
        led_display(temp); //显示按键值
    }
}
```

按硬件连接所示,把硬件连接好(红色框内部分),将程序编译后写进单片机之后,按下不同的按键,就可以在数码管上看出他们对应的键值.