

从零开始的51单片机开发

一、代码

1.LED流水灯操作：（通过控制寄存器实现）

```
#include <REGX52.H>
#include <INTRINS.H>
void Delay500ms()      //进行延迟500毫秒
{
    unsigned char i, j, k;

    _nop_();
    _nop_();
    i = 23;
    j = 205;
    k = 120;
    do
    {
        do
        {
            while (--k);
        } while (--j);
    } while (--i);
}

void main()
{
    while(1)      //从第一个灯开始依次亮灭，把LED灯的编号从二进制转化为十六进制
    {
        P2=0xFE; //11111110
        Delay500ms();
        P2=0xFD; //11111101
        Delay500ms();
        P2=0xFB; //11111011
        Delay500ms();
        P2=0xF7; //11110111
        Delay500ms();
        P2=0xEF; //11101111
        Delay500ms();
        P2=0xDF; //11011111
        Delay500ms();
        P2=0xBF; //10111111
        Delay500ms();
        P2=0x7F; //01111111
        Delay500ms();
    }
}
```

改进版：

```
#include <REGX52.H>
#include <INTRINS.H>
```

```

void Delay500ms(unsigned int x)    //进行延迟500毫秒
{
    unsigned char i, j, k;
    while (x)
    {
        i = 12;
        j = 169;
        do
        {
            while (--j);
        } while (--i);
        x--;
    }
}

void main()
{
    while(1)    //从第一个灯开始依次亮灭，把LED灯的编号从二进制转化为十六进制
    {
        P2=0xFE; //11111110
        Delay500ms(500);
        P2=0xFD; //11111101
        Delay500ms(500);
        P2=0xFB; //11111011
        Delay500ms(500);
        P2=0xF7; //11110111
        Delay500ms(500);
        P2=0xEF; //11101111
        Delay500ms(500);
        P2=0xDF; //11011111
        Delay500ms(500);
        P2=0xBF; //10111111
        Delay500ms(500);
        P2=0x7F; //01111111
        Delay500ms(500);
    }
}

```

独立按键控制LED开关

```

#include <REGX52.H>

void main()
{
    while (1)
    {
        if(P3_1 == 0)    //P3_1是指第一个独立按键，其余依次是P3_0,P3_2,P3_3
        {
            P2_0 = 0;    //P2_0是指第一个LED灯
        }
        else
        {
            P2_0 = 1;
        }
    }
}

```

```
}  
}
```

按键控制LED开关

```
#include <REGX52.H>  
void Delay1ms(unsigned int x)    //自定义的延时代码  
{  
    unsigned char i, j;  
    while(x){  
        i = 2;  
        j = 239;  
        do  
        {  
            while (--j);  
        } while (--i);  
        x--;  
    }  
}  
  
void main()  
{  
    while (1)  
    {  
        if (P3_1 == 0)    //当按键被按下  
        {  
            Delay1ms(20);    //由硬件知识可知，按下后需要延迟  
            while (P3_1 == 0);  
            Delay1ms(20);  
            P2_0 = ~P2_0;    //按完后，取反，使其开灯或者关灯  
        }  
    }  
}
```

LED灯显示二进制

```
#include <REGX52.H>  
void Delay1ms(unsigned int x)    //自定义的延时代码  
{  
    unsigned char i, j;  
    while(x){  
        i = 2;  
        j = 239;  
        do  
        {  
            while (--j);  
        } while (--i);  
        x--;  
    }  
}  
  
void main()  
{
```

```

unsigned char LED = 0;
while (1)
{
    if (P3_1 == 0)
    {
        Delay1ms(20);
        while (P3_1 == 0);
        Delay1ms(20);
        LED++;
        P2 = ~LED; //通过取反进行二进制显示
    }
}
}

```

按键控制LED灯移位

```

#include <REG52.H>
void Delay1ms(unsigned int x)
{
    unsigned char i, j;
    while(x){
        i = 2;
        j = 239;
        do
        {
            while (--j);
        } while (--i);
        x--;
    }
}

unsigned char LED;

void main()
{
    while (1)
    {
        if(P3_1 == 0)
        {
            Delay1ms(20);
            while (P3_1 == 0);
            Delay1ms(20);
            if(LED >= 8)
            {
                LED = 0;
            }
            P2 = ~(0x01<<LED);
            LED++;
        }
        if (P3_0 == 0)
        {
            Delay1ms(20);
            while (P3_0 == 0);
            Delay1ms(20);
            if (LED <= 0)
            {
                LED = 7;
            }
        }
    }
}

```

```

    }
    P2 = ~(0x01<<LED);
    LED--;
}
}
}

```

(有小BUG, 懒得改了)

静态数码管显示

```

#include <REGX52.H>

unsigned char NUMLED[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F};

void Nixie(unsigned char Location, num)
{
    switch (Location)
    {
        case 1:P2_4 = 1;P2_3 = 1;P2_2 = 1;break;      //从左到右第一个, 由硬件图知对应Y7
        case 2:P2_4 = 1;P2_3 = 1;P2_2 = 0;break;
        case 3:P2_4 = 1;P2_3 = 0;P2_2 = 1;break;
        case 4:P2_4 = 1;P2_3 = 0;P2_2 = 0;break;
        case 5:P2_4 = 0;P2_3 = 1;P2_2 = 1;break;
        case 6:P2_4 = 0;P2_3 = 1;P2_2 = 0;break;
        case 7:P2_4 = 0;P2_3 = 0;P2_2 = 1;break;
        case 8:P2_4 = 0;P2_3 = 0;P2_2 = 0;break;
    }
    P0 = NUMLED[num];      //表示数码管通过十六进制显示数字
}

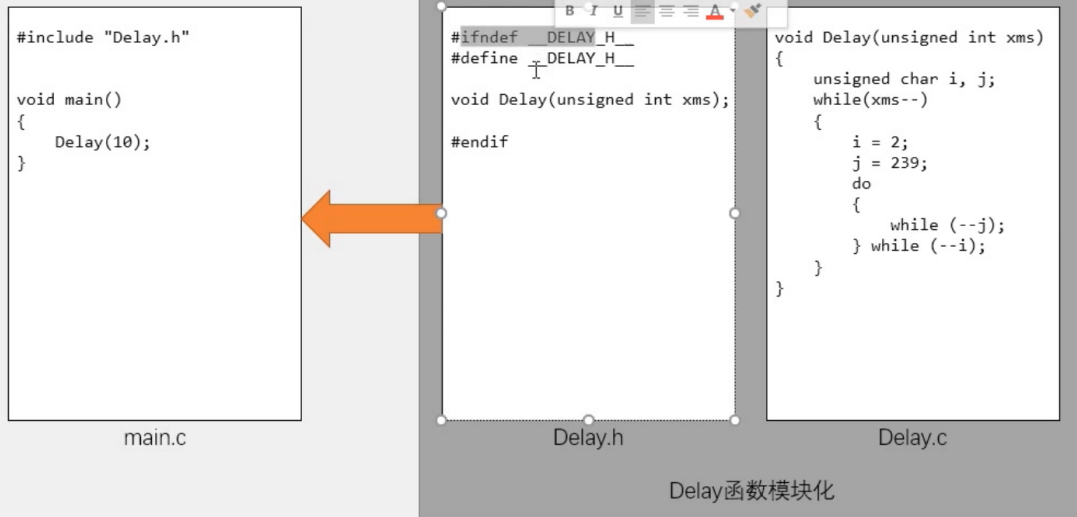
void main()
{
    Nixie(2, 3);
    while (1)
    {

    }
}

```

模块化编程

模块化编程框图



LCD显示屏显示

```
#include <REG52.H>
#include "LCD1602.h"

void main()
{
    LCD_Init();           //LCD初始化
    LCD_ShowChar(1, 1, 'A'); //在第一行第一列显示
    LCD_ShowString(2, 1, "Hello,world!!!");
    while (1)
    {
    }
}
```

更多用法如下图

LCD1602调试工具

- 使用LCD1602液晶屏作为调试窗口，提供类似printf函数的功能，可实时观察单片机内部数据的变换情况，便于调试和演示。
- 本视频提供的LCD1602代码属于模块化的代码，使用者只需要知道所提供函数的作用和使用方法就可以很容易的使用LCD1602

函数	作用
LCD_Init();	初始化
LCD_ShowChar(1,1,'A');	显示一个字符
LCD_ShowString(1,3,"Hello");	显示字符串
LCD_ShowNum(1,9,123,3);	显示十进制数字
LCD_ShowSignedNum(1,13,-66,2);	显示有符号十进制数字
LCD_ShowHexNum(2,1,0xA8,2);	显示十六进制数字
LCD_ShowBinNum(2,4,0xAA,8);	显示二进制数字

通过矩阵键盘更改LCD屏幕上的数字

```
#include <REGX52.H>
#include "Delay.h"

unsigned char MatrixKey()
{
    unsigned char KeyNUM = 0;
    P1 = 0xFF;          //按键通电
    P1_3 = 0;           //按键第一列
    if (P1_7 == 0)      //第一行
    {
        Delay1ms(20);
        while (P1_7 == 0);
        Delay1ms(20);
        KeyNUM = 1;
    }
    if (P1_6 == 0)
    {
        Delay1ms(20);
        while (P1_6 == 0);
        Delay1ms(20);
        KeyNUM = 5;
    }
    if (P1_5 == 0)
    {
        Delay1ms(20);
        while (P1_5 == 0);
        Delay1ms(20);
        KeyNUM = 9;
    }
    if (P1_4 == 0)
    {
        Delay1ms(20);
        while (P1_4 == 0);
        Delay1ms(20);
        KeyNUM = 13;
    }
    P1 = 0xFF;
    P1_2 = 0;
    if (P1_7 == 0)
    {
        Delay1ms(20);
        while (P1_7 == 0);
        Delay1ms(20);
        KeyNUM = 2;
    }
    if (P1_6 == 0)
    {
        Delay1ms(20);
        while (P1_6 == 0);
        Delay1ms(20);
        KeyNUM = 6;
    }
}
```

```

    if (P1_5 == 0)
    {
        Delay1ms(20);
        while (P1_5 == 0);
        Delay1ms(20);
        KeyNUM = 10;
    }

    if (P1_4 == 0)
    {
        Delay1ms(20);
        while (P1_4 == 0);
        Delay1ms(20);
        KeyNUM = 14;
    }

    P1 = 0xFF;
    P1_1 = 0;
    if (P1_7 == 0)
    {
        Delay1ms(20);
        while (P1_7 == 0);
        Delay1ms(20);
        KeyNUM = 3;
    }

    if (P1_6 == 0)
    {
        Delay1ms(20);
        while (P1_6 == 0);
        Delay1ms(20);
        KeyNUM = 7;
    }

    if (P1_5 == 0)
    {
        Delay1ms(20);
        while (P1_5 == 0);
        Delay1ms(20);
        KeyNUM = 11;
    }

    if (P1_4 == 0)
    {
        Delay1ms(20);
        while (P1_4 == 0);
        Delay1ms(20);
        KeyNUM = 15;
    }

    P1 = 0xFF;
    P1_0 = 0;
    if (P1_7 == 0)
    {
        Delay1ms(20);
        while (P1_7 == 0);
        Delay1ms(20);
        KeyNUM = 4;
    }

    if (P1_6 == 0)
    {
        Delay1ms(20);
        while (P1_6 == 0);
        Delay1ms(20);

```



```

        KeyNUM = 8;
    }
    if (P1_5 == 0)
    {
        Delay1ms(20);
        while (P1_5 == 0);
        Delay1ms(20);
        KeyNUM = 12;
    }
    if (P1_4 == 0)
    {
        Delay1ms(20);
        while (P1_4 == 0);
        Delay1ms(20);
        KeyNUM = 16;
    }
    return KeyNUM;
}

```

```

#include <REGX52.H>
#include "LCD1602.h"
#include "MatrixKey.h"
#include "Delay.h"

unsigned char KeyNUM;

void main()
{
    LCD_Init();
    LCD_ShowString(1, 1, "NUMBER:");    //获得需要显示的数字
    while (1)
    {
        KeyNUM = Matrixkey();
        if (KeyNUM)
        {
            LCD_ShowNum(2, 1, KeyNUM, 2);
        }
    }
}

```

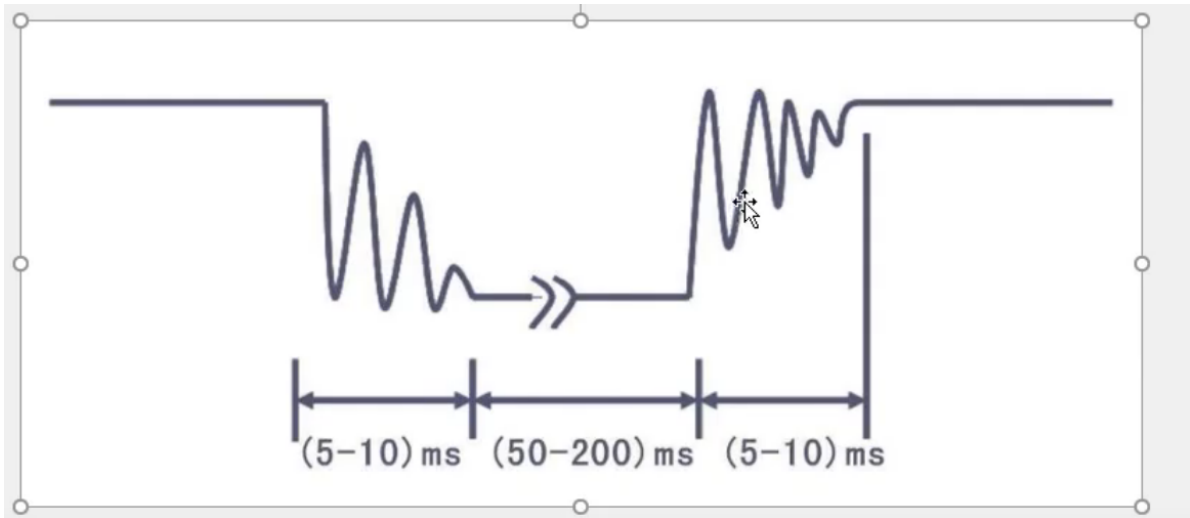
单片机的硬件部分认识

独立按键

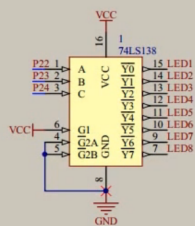
- 轻触按键：相当于是一种电子开关，按下时开关接通，松开时开关断开，实现原理是通过轻触按键内部的金属弹片受力弹动来实现接通和断开

按键的抖动

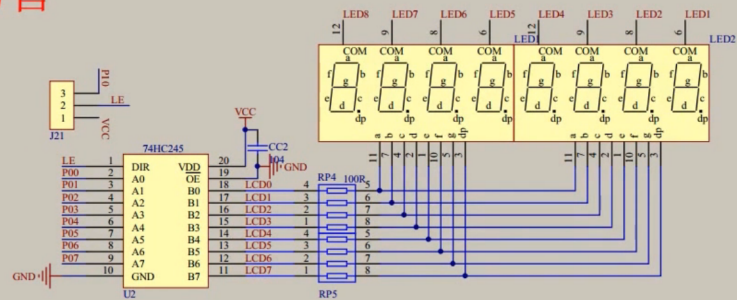
- 对于机械开关，当机械触点断开、闭合时，由于机械触点的弹性作用，一个开关在闭合时不会马上稳定地接通，在断开时也不会一下子断开，所以在开关闭合及断开的瞬间会伴随一连串的抖动



138译码器

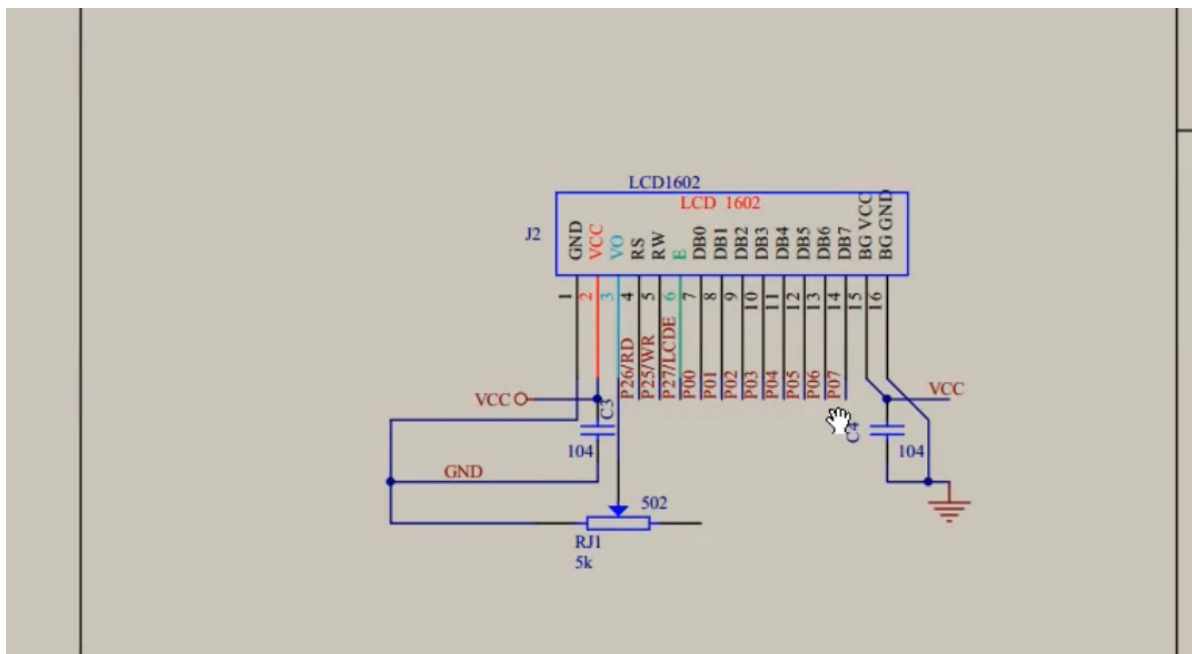


数码管



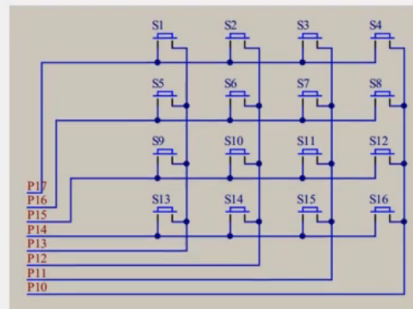
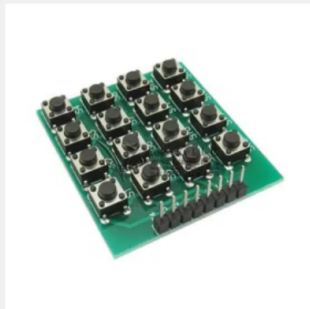
数码管段码表

0	1	2	3	4	5	6	7	8	9
0x3F	0x06	0x5B	0x4F	0x66	0x6D	0x7D	0x07	0x7F	0x6F
A	B	C	D	E	F	空			
0x77	0x7C	0x39	0x5E	0x79	0x71	0x00			

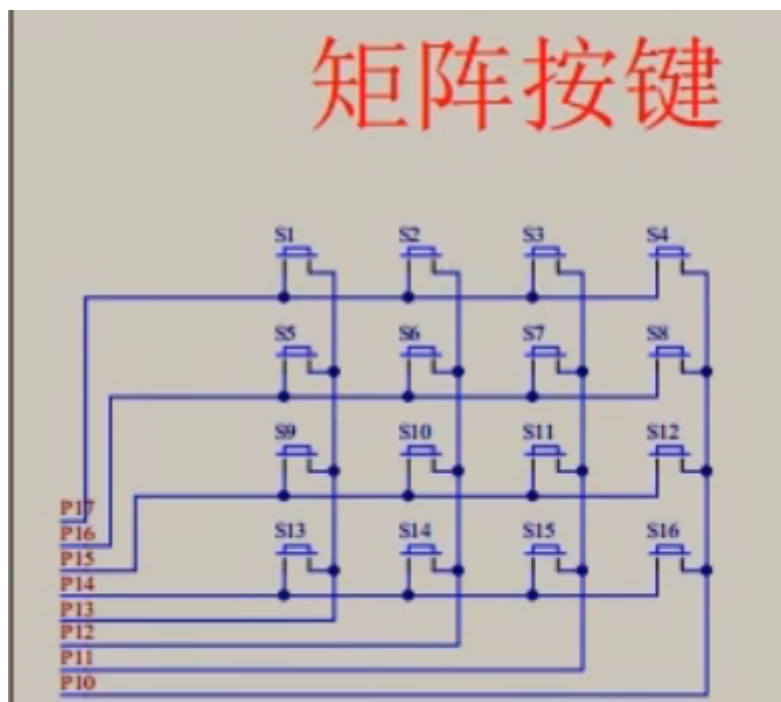


矩阵键盘介绍

- 在键盘中按键数量较多时，为了减少I/O口的占用，通常将按键排列成矩阵形式
- 采用逐行或逐列的“扫描”，就可以读出任何位置按键的状态



矩阵按键



- 数码管扫描（输出扫描）

原理：显示第1位→显示第2位→显示第3位→……，然后快速循环这个过程，最终实现所有数码管同时显示的效果

- 矩阵键盘扫描（输入扫描）

原理：读取第1行(列)→读取第2行(列)→读取第3行(列)→……，然后快速循环这个过程，最终实现所有按键同时检测的效果

- 以上两种扫描方式的共性：节省I/O口