

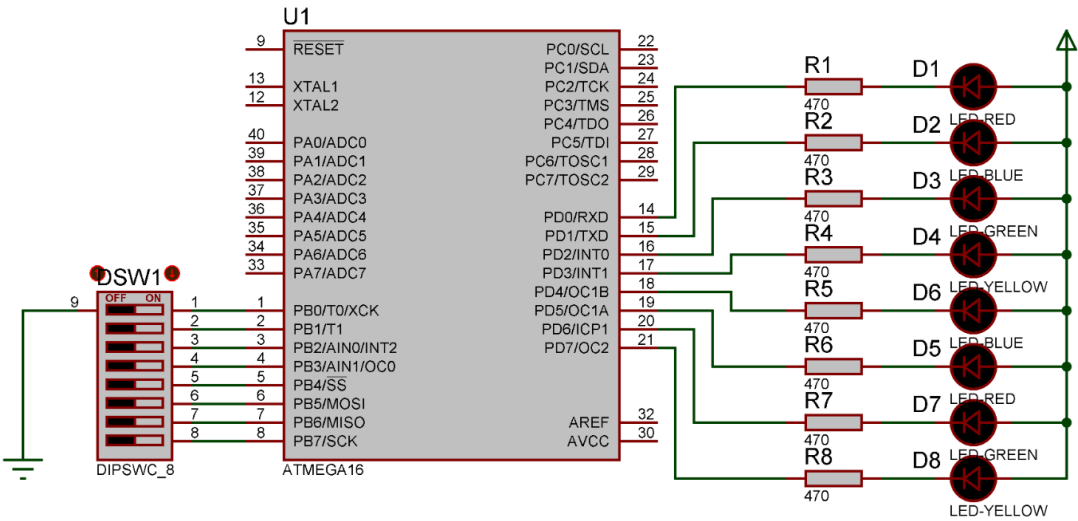
试验 3 8 路指示灯读出 8 路拨动开关的状态

1. 试验描述

利用 PB 端口的 8 路拨动开关控制 PD 端口相应的 8 路 LED 指示灯亮灭。从而理解单片机中数和位的概念以及数据传递的概念。

2. 系统框图

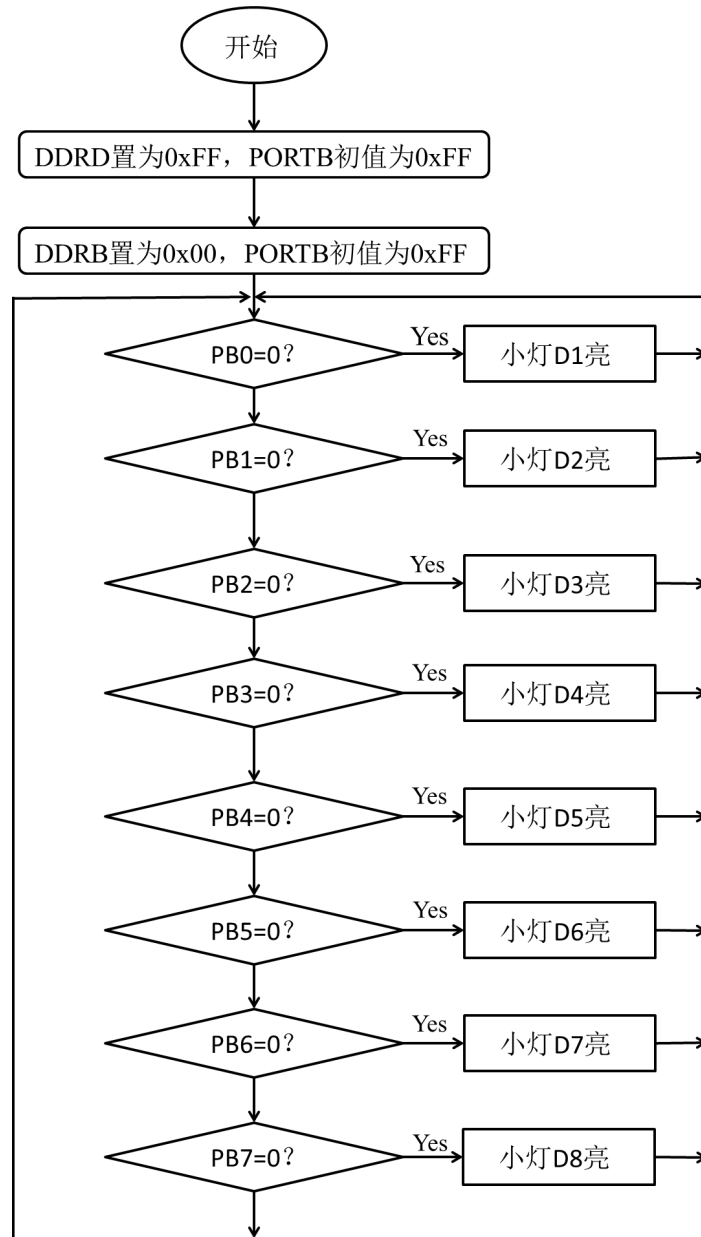
➤ 硬件电路



➤ 元件清单

单片机 ATmega16	电阻 RES	8 路开关 DIPSWC_8	发光二极管 LED-RED
发光二极管 LED-BLUE	发光二极管 LED-GREEN	发光二极管 LED-YELLOW	

➤ 软件流程



3.程序设计

➤ ICC 程序

```
#include <iom16v.h>
```

```
/* 逐位查询 PB 口是否为 0，若为 0，将 PD 口对应 IO 连接的 LED 点亮，PD 其他 IO 连接的 LED 熄灭 */
```

```
void key(void)
{
```

```

    unsigned char i = 0;
    for (; i < 8; i++)
    {
        if ((PINB & (1 << i)) == 0)
            PORTD = ~(1 << i);
    }
}

void main(void)
{
    DDRD = 0xff; // 设置 PD 口输出
    PORTD = 0xff; // LED 全灭
    DDRB = 0x00; // 设置 PC 口输入
    PORTB = 0xff; // 上拉电阻有效
    while (1)
    {
        key();
    }
}

```

➤ CAVR 程序

```

#include <mega16.h>

/* 逐位查询 PB 口是否为 0，若为 0，将 PD 口对
   应 IO 连接的 LED 点亮，PD 其他 IO 连接的 LED 熄灭 */

void key(void)
{
    unsigned char i = 0;
    for (; i < 8; i++)
    {
        if ((PINB & (1 << i)) == 0)
            PORTD = ~(1 << i);
    }
}

void main(void)
{
    DDRD = 0xff; // 设置 PD 口输出
    PORTD = 0xff; // LED 全灭
    DDRB = 0x00; // 设置 PC 口输入
    PORTB = 0xff; // 上拉电阻有效
    while (1)
    {

```

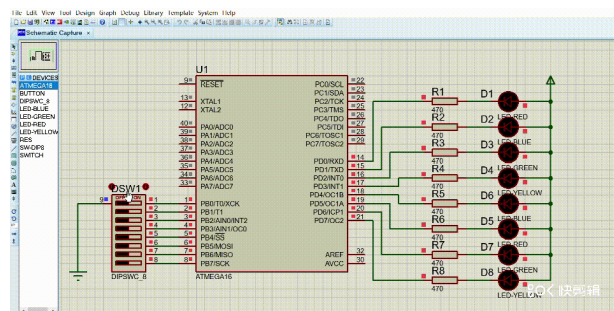
```

        key();
    }
}

```

4. 仿真结果

按下 8 路开关中的某个开关时，相应的小灯会被点亮，如：按下 8 路开关的第四个开关，小灯 D4 被点亮。



（双击图片，演示仿真结果动画）

5. 改进

例程的代码并没有对关闭拨码开关的情况做出判断和处理，导致打开 8 路拨码开关后至少有一个 LED 会被点亮，不能关闭。并且在点亮一个 LED 时关闭其他 LED，在多个拨码开关打开时尽管看起来对应的 LED 都被点亮，但是同一时刻最多只有一个 LED 点亮，随着点亮的 LED 增多，LED 的亮度逐渐变暗。因此将代码优化。

➤ 程序

```

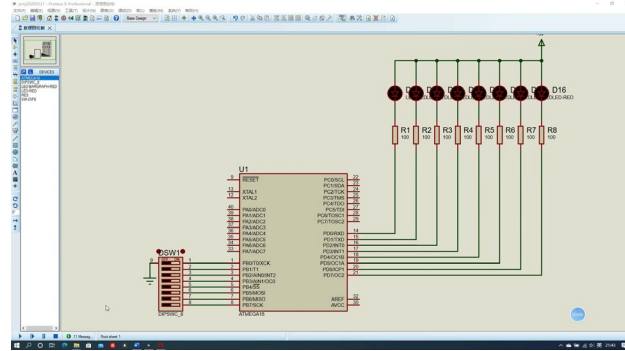
#include <mega16.h>

void main(void)
{
    DDRD = 0xff; // PD 口输出方式
    PORTD = 0xff; // LED 全灭
    DDRB = 0x00; // PB 口输入方式
    PORTB = 0xff; // 上拉电阻有效
    while (1)
    {
        PORTD = PINB; // 将 PB 口的值赋给 PD 口
    }
}

```

➤ 仿真结果

打开 8 路拨码开关对应的 LED 被点亮，亮度不会降低。关闭所有拨码开关，所有的 LED 也熄灭。

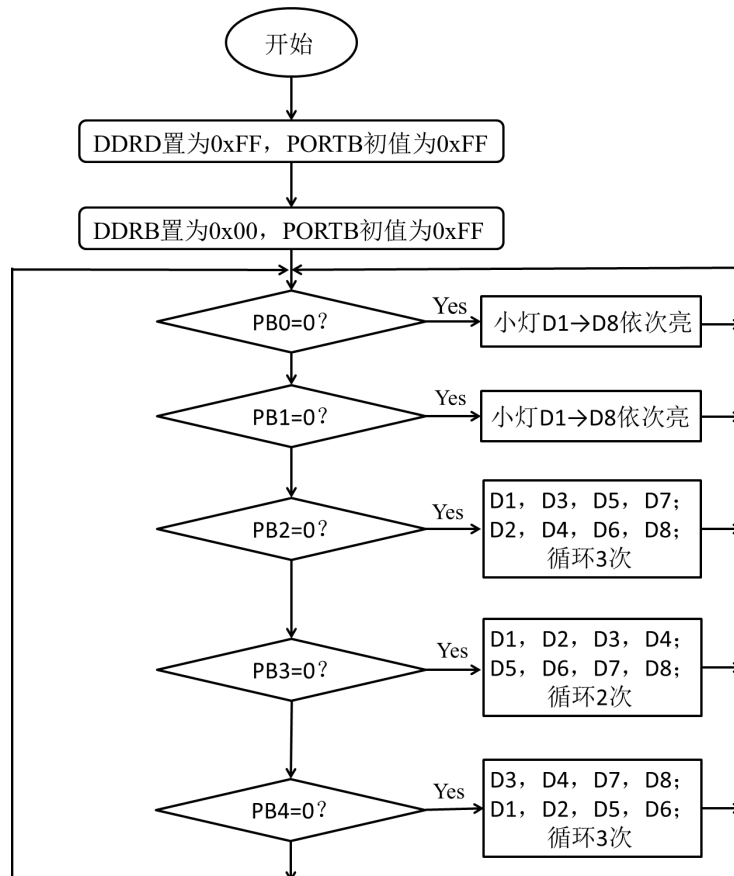


(双击图片，演示仿真结果动画)

练习

使用 PB 端口和 PD 端口控制 LED0-LED7(即 D1-D8)进行花样显示;

➤ 软件流程



➤ 程序

```
#include "mega16.h"
#define uchar unsigned char
#define uint unsigned int
//定义开关及 LED 与端口的连接
#define k1 PINB .0
#define k2 PINB .1
#define k3 PINB .2
#define k4 PINB .3
#define k5 PINB .4
flash uchar tab1[8] = {0xFE, 0xFD, 0xFB, 0xF7, //正向流水灯
                        0xEF, 0xDF, 0xBF, 0x7F};
flash uchar tab2[8] = {0x7F, 0xBF, 0xDF, 0xEF, //反向流水灯
                        0xF7, 0xFB, 0xFD, 0xFE};
flash uchar tab3[6] = {0xaa, 0x55, 0xaa, 0x55, //隔灯闪烁
                        0xaa, 0x55};
flash uchar tab4[4] = {0xF0, 0x0F, 0xF0, 0x0F}; //高四盏低四盏闪烁
flash uchar tab5[6] = {0x33, 0xcc, 0x33, 0xcc, //隔灯闪烁
                        0x33, 0xcc};

void delay(uint k)
{
    uint m, n;
    for (m = 0; m < k; m++)
    {
        for (n = 0; n < 1140; n++)
        {
            ;
        }
    }
}

void main(void)
{
    uint i;
    DDRB = 0x00;
    PORTB = 0xFF;
    DDRD = 0xFF;
    PORTD = 0xFF;
    while (1)
    {
        if (k1 == 0)
        {
            for (i = 0; i < 8; i++)
```

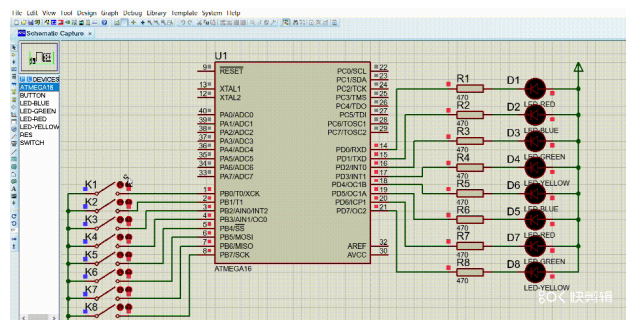
```

    {
        PORTD = tab1[i];
        delay(100);
    }
}
else if (k2 == 0)
{
    for (i = 0; i < 8; i++)
    {
        PORTD = tab2[i];
        delay(100);
    }
}
else if (k3 == 0)
{
    for (i = 0; i < 6; i++)
    {
        PORTD = tab3[i];
        delay(100);
    }
}
else if (k4 == 0)
{
    for (i = 0; i < 4; i++)
    {
        PORTD = tab4[i];
        delay(100);
    }
}
else if (k5 == 0)
{
    for (i = 0; i < 6; i++)
    {
        PORTD = tab5[i];
        delay(100);
    }
}
else
{
    PORTD = 0xFF;
}
}
}

```

➤ 仿真结果

该设计显示规律为：开关 K1 闭合时，D1→D8，即 8 个 LED 每隔 1s 依次左移点亮；开关 K2 闭合时，D8→D1，即 8 个 LED 每隔 1s 依次右移点亮；开关 K3 闭合时，D1, D3, D5, D7 亮 1s 后熄灭，接着 D2, D4, D6, D8 亮 1s 后熄灭……循环 3 次；开关 K4 闭合时，D1, D2, D3, D4 亮 1s 后熄灭，接着 D5, D6, D7, D8 亮 1s 后熄灭……循环 2 次；开关 K5 闭合时，D3, D4, D7, D8 亮 1s 后熄灭，接着 D1, D2, D5, D7 亮 1s 后熄灭……循环三次；没有任何开关闭合时，8 个 LED 灯全部熄灭。



(双击图片，演示仿真结果动画)