

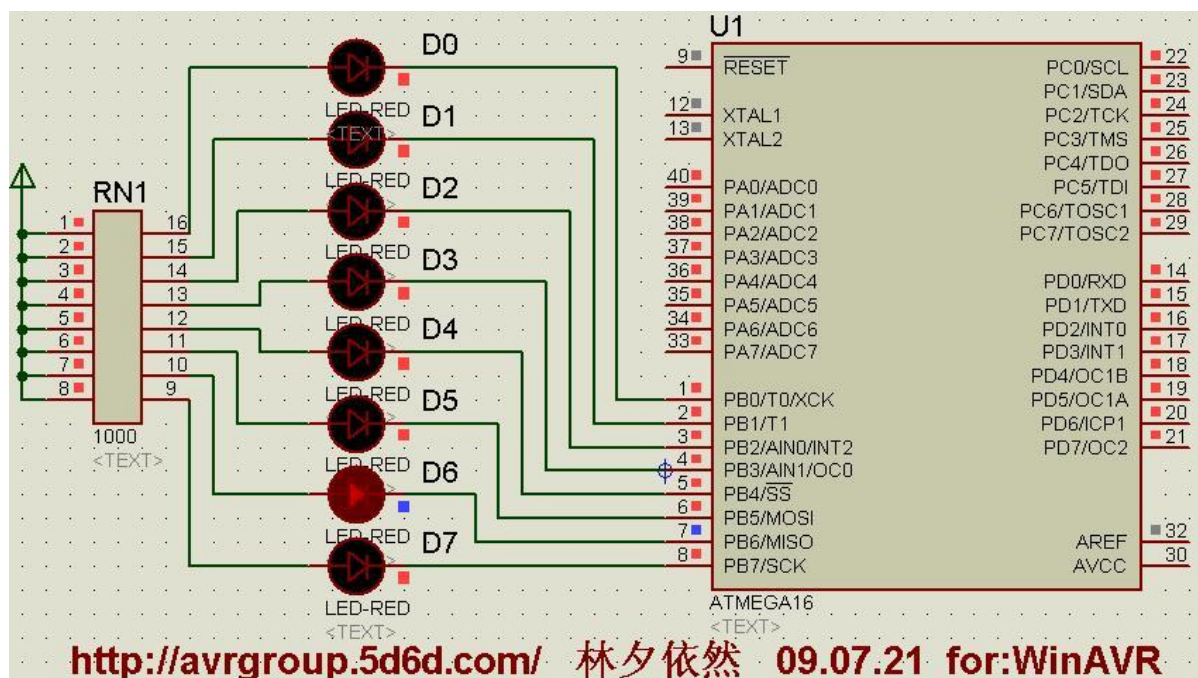
# 实验 1: 8 种 LED 点亮模式(GCC)

更多笔记: <http://bbs.armavr.com/>

## 一、程序结构



## 二、仿真效果



## 三、程序源码

### 1、main.c

```

/*****
Platform : AVR mega16 学习板 (www.iccavr.com)
Project  : 实验一: 8 种 LED 点亮模式
Clock F  : 3.6864M
Software : WinAVR-20071221+Proteus7.4
Author   : 林夕依然
Version  : 08.11.22
Updata   : 09.02.25 模块化
           09.04.28 增加了 proteus 仿真模型, 调试通过
           09.07.21 WinAVR-20071221 环境下编译通过
comments :
1、以学习板八个 LED 灯为硬件电路,JP7 短路块需装上
2、练习简单延时函数的编制
3、AVR 单片机端口寄存器的使用及理解

```

### comments :

- 1、以学习板八个 LED 灯为硬件电路,JP7 短路块需装上
- 2、练习简单延时函数的编制
- 3、AVR 单片机端口寄存器的使用及理解

- 4、练习程序模块化，结构化的书写
- 5、单片机休眠模式练习
- 6、8种LED点亮模式
- 7、使用GCC时的更改：延时函数使用GCC自带，main()返回类型改为int型，io头文件，位操作使用\_BV(i)

```
*****/
#include <avr/io.h>
#include "led01.h"
int main(void)
{
    int l,m,n,o,p,q,r,s,i,j;
DDRA =0X00;                //端口上拉输入
PORTA=0XFF;
    DDRB =0xFF;           //端口输出
    PORTB=0xFF;           //输出高电平，LED 熄灭
DDRC =0X00;
PORTC=0XFF;
DDRD =0X00;
PORTD=0XFF;
    for (r=0;r<5;r++)
    {
        for(l=0;l<5;l++)           //模式 1：顺序点亮
        {
            for (i = 0; i < 8; i++) //顺序单个点亮 LED
                LED_01(i);
            for (i = 7; i > 0; i--) //逆序单个点亮 LED
                LED_01(i);
        }
        LED_off();

        for(m=0;m<5;m++)           //模式 2：顺序单个间隔点亮
        {
            for (i = 0; i < 8; i += 2) //顺序间隔点亮 LED
                LED_01(i);
            for (i = 7; i > 0; i -= 2) //逆序间隔点亮 LED
                LED_01(i);
        }
        LED_off();

        for(n=0;n<5;n++)           //模式 3：间隔点亮
        {
            for (i = 2; i < 8; i++) //间隔顺序同时点亮
                LED_02(i);
            for (i = 6; i > 2; i--) //间隔逆序同时点亮
                LED_02(i);
        }
        LED_off();
    }
}
```

```
for(o=0;o<5;o++) //模式 4: 相临点亮
{
    for (i = 1; i < 8; i++) //相临顺序同时点亮
        LED_03(i);
    for (i = 6; i > 1; i--) //相临逆序同时点亮
        LED_03(i);
}
LED_off();
for(p=0;p<5;p++) //模式 5: 发散聚集点亮
{
    for(i=0;i<4;i++) //发散点亮
        LED_04(i);
    for(i=2;i>0;i--) //聚集点亮
        LED_04(i);
}
LED_off();
for(q=0;q<5;q++) //模式 6: 四四点亮
{
    for(i=0;i<4;i++) //四四顺序点亮
        LED_05(i);
    for(i=2;i>0;i--) //四四逆序点亮
        LED_05(i);
}
LED_off();

for(s=0;s<5;s++) //模式 7: 四四点亮
{
    for(i=0;i<2;i++) //四四顺序点亮
        LED_06(i);
}
LED_off();

for(j=0;j<10;j++) //模式 8: 全部点亮熄灭
{
    LED_on();
LED_off();
}
}
//MCUCR=0x40; //空闲模式, CPU 占用 100%
//MCUCR=0x50; //ADC 噪声抑制模式, CPU 占用 100%
//MCUCR=0x60; //掉电模式, CPU 占用 80%
//MCUCR=0x70; //省电模式, CPU 占用 4%
//MCUCR=0xE0; //Standby 模式, CPU 占用 80%
MCUCR=0xF0; //扩展 Standby 模式, CPU 占用 4%
asm("sleep"); //CPU 休眠指令
}
```

## 2、led01.h

/\*\*\*\*\*

Platform : AVR mega16 学习板 ([www.iccavr.com](http://www.iccavr.com))

function : 功能函数集

Clock F : 3.6864M

Software : WinAVR-20071221+Proteus7.4

Author : 林夕依然

Version : 09.02.25

Updata : 09.07.21 WinAVR-20071221 环境下编译通过

comments :

\*\*\*\*\*/

```
#include <avr/io.h>
#include <util/delay.h>
#define uchar unsigned char
#define uint unsigned int

void LED_on(void) //打开所有 LED
{
    PORTB =0X00;
    _delay_ms(100);
}

void LED_off(void) //关闭所有 LED
{
    PORTB = 0xFF;
    _delay_ms(100);
}

void LED_01(int i) //LED 亮灭控制
{
    PORTB = ~_BV(i); //输出低电平
    _delay_ms(100); //调用延时程序
}

void LED_02(int i) //间隔点亮
{
    PORTB=~(_BV(i)|_BV(i-2));
    _delay_ms(100);
}

void LED_03(int i) //相临点亮
{
    PORTB=~(_BV(i)|_BV(i-1)); //~后内容需用括号括起来
    _delay_ms(100);
}

void LED_04(int i) //发散聚集点亮
{
    switch(i)
    {
```

```
case 0:PORTB=0xE7;_delay_ms(100);break;    //延时 100ms
case 1:PORTB=0xDB;_delay_ms(100);break;
case 2:PORTB=0xBD;_delay_ms(100);break;
case 3:PORTB=0x7E;_delay_ms(100);break;
default:break;
}
}

void LED_05(int i)                        //00,0F,F0,FF 方式显示
{
switch(i)
{
case 0:PORTB=0x00;_delay_ms(100);break;    //延时 100ms
case 1:PORTB=0x0F;_delay_ms(100);break;
case 2:PORTB=0xF0;_delay_ms(100);break;
case 3:PORTB=0xFF;_delay_ms(100);break;
default:break;
}
}

void LED_06(int i)
{
switch(i)
{
case 0:PORTB=0XAA;_delay_ms(100);break;
case 1:PORTB=0X55;_delay_ms(100);break;
}
}
}
```

四、完整项目文件下载

<http://bbs.armavr.com/thread-788-1-2.html>