

实例 03 电子调光灯



设计思路

电子调光灯应用广泛。目前常见电子调光灯的调光方式主要有 3 种：第 1 种是改变晶闸管的导通角；第 2 种是利用变压器调节供电电压；第 3 种是利用电位器直接分压。较理想的方式是通过改变晶闸管的导通角实现调光。

晶闸管调光的原理是利用可调电阻改变电容的充放电速度，从而改变晶闸管的导通角，控制电子调光灯在交流电源一个正弦周期内的导通时间，进而达到调光的目的。

本实例主要采用双向晶闸管实现调光。使用者通过按键控制电子调光灯开、关，通过按键控制灯光的亮度。



器件介绍

双向晶闸管直接接在 220V 交流电路中，但是单片机采用低电压供电，因此需要采用一定的隔离措施，将 220V 强电与 5V 弱电隔离。本实例使用 MOC3021 作为强电与弱电的隔离器。

MOC3021 是美国摩托罗拉公司推出的器件，输入与输出采用光电隔离，绝缘电压可达 7500V，最大触发电流为 15mA。

MOC3021 可以用于驱动工作电压为 220V 的交流双向晶闸管，也可以直接驱动小功率负载，还适用于电磁阀及电磁铁控制、电机驱动、温度控制、固态继电器控制、交流电源开关控制等场合。MOC3021 能用 TTL 电平驱动，它很容易与微处理器接口连接，对各种自动控制设备进行实时控制。

MOC3021 采用 DIP-6 封装形式，其引脚如图 3-1 所示。1、2 引脚为输入端，输入级是一只砷化镓红外 LED；4、6 引脚为输出端，输出级为光控双向晶闸管；3、5 引脚为空脚。当砷化镓红外 LED 发射红外光时，触发光控双向晶闸管导通。MOC3021 的典型应用电路如图 3-2 所示。

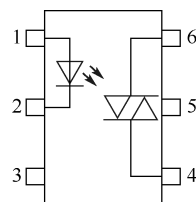


图 3-1 MOC3021 的引脚

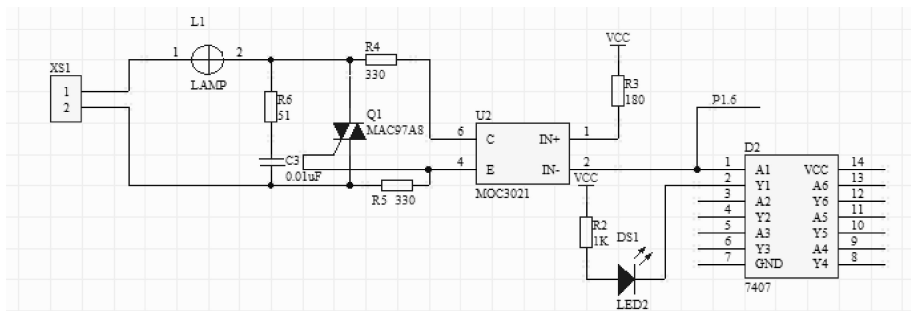


图 3-2 MOC3021 的典型应用电路



硬件设计

本实例通过单片机控制双向晶闸管的导通角来实现调光，如图 3-3 所示。整个电路主要包括双向晶闸管控制电路及过零检测电路。

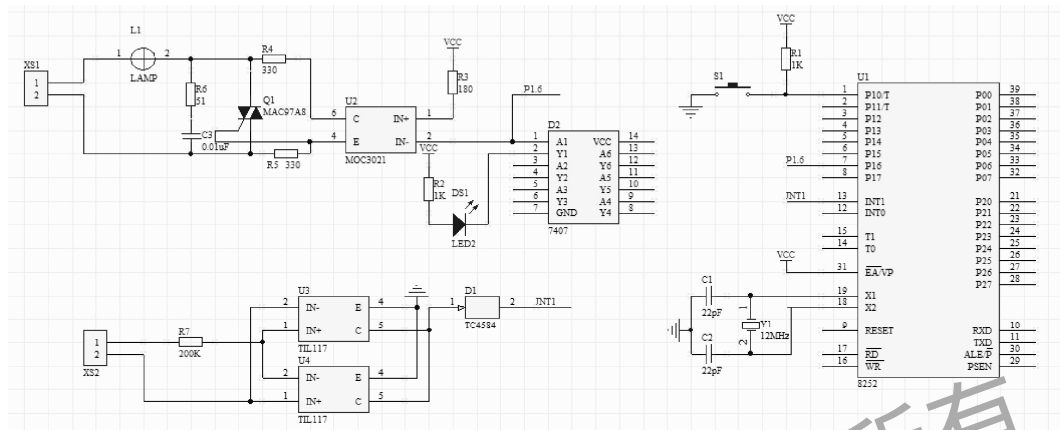


图 3-3 电子调光灯电路

MOC3021 是光电耦合器，用于可靠驱动双向晶闸管并实现强电与弱电的隔离。单片机的 P16 引脚负责驱动 MOC3021，控制双向晶闸管导通和关断。在电子调光灯的主回路中，灯与双向晶闸管串联，双向晶闸管导通角的变化会改变灯光亮度。XS1 是外部 220V 交流电源的接入口。

为了精确控制双向晶闸管的导通角，电子调光灯电路中还加入了过零检测电路。交流电源从 XS2 引入并送入两片光电耦合器 TIL117，两片光电耦合器 TIL117 的输入端是反相的。这样使得交流电压过零时，无论是由正电压变为负电压还是由负电压变为正电压，都能够在光电耦合器的输出端 C 上得到一个正向阶跃信号。其经过施密特触发器 TC4584 整形并反相输出到单片机的 INT1 引脚上，作为中断触发信号。单片机根据此信号获得交流电压每个正弦周期内的两个过零点。



程序设计

如前文所述，电子调光灯的核心是通过单片机控制双向晶闸管的导通角来实现调光。在交流电压的每个过零点，通过过零检测电路向单片机的 INT1 引脚发出中断触发信号，单片机获得控制周期的起点信号，控制双向晶闸管关断，并启动定时器。在定时器定时结束后才改变双向晶闸管控制端口的驱动信号，导通双向晶闸管。假设定时器的定时时间为 T ，则在交流电压的一个正弦周期 20ms 内，双向晶闸管导通的时间为 $20\text{ms}-2T$ 。

电子调光灯的主要功能由子程序 Check()和子程序 ServiceINT1()实现。子程序 Check()的功能是进行按键响应。该程序对按键的处理包括去抖及区分长时间按下、短时间按下，从而设置相应的标志位，为灯光控制决策提供依据。Check()的代码如下。

```
/*-----*/
```

```

void Check(void)
{
    LIGHT_KEY=HIGH;
    /*确保各输入引脚处于输入状态*/
    if (LIGHT_KEY==LOW)
    {
        if (LIGHT_KEYold==HIGH)
        {
            delay6ms();
            if (LIGHT_KEY==LOW)          /*调光键被按下, "LOW"为有效电平*/
            {
                if (LIGHT_KEYcounter>DEVIBRATE_FACTOR)
                {
                    LIGHTKeyPress=1;
                    LIGHT_KEYold=LOW;
                    LIGHT_KEYcounter=0;
                }
                else LIGHT_KEYcounter++;
            }
            else LIGHT_KEYcounter=0;
        }
        else
        {
            LIGHTKeyPress=1;
            LIGHT_KEYcounter=0;
        }
    }
    else
    {
        if (LIGHT_KEYold==LOW)
        {
            delay6ms();
            if (LIGHT_KEY==HIGH)          /*调光键被松开*/
            {
                if (LIGHT_KEYcounter>DEVIBRATE_FACTOR)
                {
                    LIGHTKeyPress=0;
                    LIGHT_KEYold=HIGH;
                    LIGHT_KEYcounter=0;
                }
                else LIGHT_KEYcounter++;
            }
            else LIGHT_KEYcounter=0;
        }
        else
        {
            LIGHTKeyPress=0;

```

```

        LIGHT_KEYcounter=0;
    }
}
}

```

外部中断响应子程序 `ServiceINT1()` 的功能是根据按键状态进行电子调光灯的亮灭或灯光的调节处理。按键的响应机制：当短时间按下按键（按下时间大于 6ms，否则认为是抖动，不予处理）时，电子调光灯在开和关两种状态下切换；当长时间按下按键时，电子调光灯进入调光状态，灯光先由暗到亮，再由亮到暗。`ServiceINT1()` 的代码如下。

```

/*-----*/
ServiceINT1() interrupt 2 using 1
{
    /*过零检测时间到*/
    TR2=0;
    LIGHT=HIGH;
    /*电子调光灯触发端关闭*/
    if (LIGHTKeyPress==1) /*调光键被按下*/
    {
        LIGHTKeyPressOld=1;
        if (LIGHTState==0) /*电子调光灯处于开关状态*/
        {
            if (LIGHTNum>=0x3f) /*调光键长时间被按下*/
            {
                LIGHTState=1;
                /*电子调光灯切换至调光状态*/
                LIGHTLevel=0x10;
                LIGHTNum=0;
            }
            else LIGHTNum++;
        }
        else /*电子调光灯处于调光状态*/
        {
            if (LIGHTSwitch==0)
            {
                if (LIGHTNum>=0x42)
                {
                    LIGHTSwitch=1;
                    LIGHTNum=0;
                }
                else LIGHTNum++;
            }
            else
            {
                if (LIGHTDimmer==0) /*进入调亮灯光环节*/
                {
                    if (LIGHTValue>0xffe0) /*0xffe0 是灯光最亮时的定时器计数限度*/

```

```

        LIGHTDimmer=1;
        /*设置调暗灯光标志*/
        else LIGHTValue+=0x13;
        /*增加定时器初值*/
    }
    else /*进入调暗灯光环节*/
    {
        if (LIGHTValue<0xe000) /*0xe000 是灯光最暗时的定时器计数限度*/
        {
            LIGHTDimmer=0;
            /*设置调亮灯光标志*/
            LIGHTSwitch=0;
            LIGHTNum=1;
        }
        else LIGHTValue-=0x13;
        /*减小定时器初值*/
    }
}
}
else /*调光键已抬起*/
{
    if (LIGHTKeyPressOld==1) /*调光键刚抬起*/
    {
        LIGHTKeyPressOld=0;
        LIGHTNum=0;
        if (LIGHTState==0) /*调光键抬起前为开关状态*/
        {
            if (LIGHTSwitch==0)
            {
                LIGHTSwitch=1;
                /*将电子调光灯设置为开*/
                LIGHTLevel|=0x10;
            }
            else
            {
                LIGHTSwitch=0;
                /*将电子调光灯设置为关*/
                LIGHTLevel&=0xef;
            }
        }
    }
    else /*调光键抬起前为调光状态*/
    {
        LIGHTState=0;
        /*电子调光灯恢复为开关状态*/
        LIGHTLevel&=0x10;
        if (LIGHTValue>0xe000)

```

51 单片机应用开发 66 例（视频点拨版）

```
        LIGHTLevel+=(LIGHTValue-0xe000)/0x01fe;
    else LIGHTLevel+=0;
    LIGHTValueH=LIGHTValue>>8;
    LIGHTValueL=LIGHTValue&0xff;
    EX0=1;
    }
    }
    }
    LIGHTValueH=LIGHTValue>>8;
    /*打开本次过零周期的定时器*/
    LIGHTValueL=LIGHTValue&0xff;
    TH2=LIGHTValueH;
    TL2=LIGHTValueL;
    TR2=1;
}
```

主程序主要用于初始化系统，调用相关子程序实现系统功能，代码如下。

```
#include "reg52.h"
#define LOW 0 /*低电平*/
#define HIGH 1 /*高电平*/
#define DEVIBRATE_FACTOR 0x03 /*去抖因子*/
sbit LIGHT=P1^6;
/*调光灯触发端*/
sbit LIGHT_KEY=P1^0;
/*调光键*/
typedef unsigned char uint8;
typedef unsigned int uint16;
bit LIGHT_KEYold;
/*按键去抖变量*/
bit LIGHTState;
/*调光标志位，0：开关状态；1：调光状态*/
bit LIGHTSwitch;
/*开灯标志位*/
bit LIGHTKeyPress;
/*当前时刻按键状态*/
bit LIGHTKeyPressOld;
/*前一刻按键状态*/
bit LIGHTDimmer;
/*调光过程，0：渐亮；1：渐暗*/
uint8 LIGHTValueH,LIGHTValueL;
/*调光定时器计数初值*/
uint8 LIGHT_KEYcounter;
/*去抖计数*/
uint8 LIGHTNum;
/*按键时长计数值*/
uint8 LIGHTLevel;
/*调光灯亮度等级*/
```

```

uint16 idata LIGHTValue;
/*定时器初值*/
/*函数定义*/
void Initialize(void);
/*初始化单片机*/
void Check();
/*检查按键和电子调光灯所处的状态*/
void delay6ms(void);
/*延时 6ms*/
void main(void)
{
    Initialize();
    do
    {
        Check();
    }
    while(1);
}
void Initialize(void)
{
    T2CON=0x00;          //T2 用于驱动电子调光灯
    PT0=1;
    IT1=1;
    TR0=0;
    TR2=0;
    ET0=1;
    ET2=1;
    EX1=1;
    EA=1;
    EX0=1;
    LIGHTState=0;
    LIGHTSwitch=0;
    LIGHTNum=0;
}
/*-----*/

void delay6ms(void)
{
    int delaycnt;
    for(delaycnt=0;delaycnt<=460;delaycnt++);
}
/*-----*/
void ServiceTimer2() interrupt 5 using 1
{
    if(LIGHTSwitch==1)
        LIGHT=LOW;      //触发电子调光灯
    TR2=0;
}

```

```
TF2=0;
```

```
}
```



经验总结

在电子调光灯电路中，最核心的部分是在灯泡供电电路中串入双向晶闸管。通过改变双向晶闸管的导通角调节灯光的亮度。在本实例调光控制的实现过程中，启用了单片机的一个定时器和一个外部中断引脚。在单片机外部中断触发时，即交流电压的过零点时刻，启动定时器计时，并关断双向晶闸管。当定时器计数值溢出，进入定时器中断后，双向晶闸管导通，灯泡获得电压而发亮。当再次到达过零点时刻时，关断双向晶闸管，如此反复。因此，随着按键的控制，改变定时器计数初值，即可改变双向晶闸管的导通角，从而实现调光。

本实例介绍的电子调光灯电路的调光基本原理和不采用单片机的普通调光灯电路是相同的，但由于加入了单片机控制，使得灯光调节可以控制得更加精确，实现多级、分级控制；另外，可以在本实例的基础上加入红外传输模块或其他无线通信模块，实现灯光的无线遥控；更进一步地，可以将其作为智能家居控制系统中的一个可控部分，利用网络或电话线等实现远程控制。



知识加油站

51 单片机的外部中断可以用程序设置为电平触发或跳变触发。若 TCON 中的 ITx 位为 0，则外部中断为电平触发，由 INTx 引脚上检测到的低电平触发；若 ITx 位为 1，则外部中断为跳变触发，即在两个周期内，若前一个周期在外部中断引脚上检测到高电平，而后一个周期在外部中断引脚上检测到低电平，则置位 TCON 寄存器中的中断请求标志位 IEx。