



小型电子产品软件开发

刘 锰 张平华◎主编



中央广播电视大学出版社
Central Radio TV University Press

小型电子产品软件开发

刘 锰 张平华 主编

中央广播电视大学出版社·北京

图书在版编目 (CIP) 数据

小型电子产品软件开发 / 刘锰, 张平华主编. -- 北
京: 中央广播电视大学出版社, 2016. 12
ISBN 978 - 7 - 304 - 08041 - 9

I. ①小… II. ①刘…②张… III. ①电子产品 - 软
件开发 - 教材 IV. ①TN02②TP311.52

中国版本图书馆 CIP 数据核字 (2016) 第 202299 号

版权所有, 翻印必究。

小型电子产品软件开发

XIAOXING DIANZI CHANPIN RUANJIAN KAIFA
刘 锰 张平华 主编

出版·发行: 中央广播电视大学出版社
电话: 营销中心 010 - 66490011
网址: <http://www.crtvup.com.cn>
地址: 北京市海淀区西四环中路 45 号
经销: 新华书店北京发行所

总编室 010 - 68182524

邮编: 100039

策划编辑: 苏 醒
责任编辑: 程业刚

责任校对: 曾繁荣
责任印制: 赵连生

印刷: 北京云浩印刷有限责任公司
版本: 2016 年 12 月第 1 版
开本: 787mm × 1092mm 1/16

2016 年 12 月第 1 次印刷
印张: 14.5 字数: 325 千字

书号: ISBN 978 - 7 - 304 - 08041 - 9
定价: 52.00 元

(如有缺页或倒装, 本社负责退换)

前言

QIANYAN

为促进高等教育健康发展，加强高等教育质量监管，实施全面、客观、科学的高等教育综合评价，湖南省高等教育界近年来探索并建立了学生专业技能抽查制度，以此来引导和激励职业院校进一步明确技能人才培养的专业定位与岗位指向，深化教育教学改革，逐步构建以能力为核心的课程体系。

软件编程是电子类专业学生技能抽查的重点之一，更是学生学习和老师教学的难点。本书主要围绕湖南省高等院校学生专业技能抽查标准与题库——《应用电子技术》的四大核心技能模块之一的“小型电子产品开发”题库进行撰写，通过对试题任务的知识技能点分析、Proteus 平台仿真和开发板实际测试等，系统、全面、有针对性地介绍和分析了全部 20 道题库，旨在提高学校针对应用电子技术专业技能抽查的教学与培训效率、学生学习效率，希望以此来提升学生的软件编程能力，从而能有效提高专业技能抽查的通过率。

本书适用于高等院校目前开设的电子信息技术、应用电子技术、电子声像技术、电子工艺与管理、电子产品质量检测、数字媒体技术和嵌入式系统工程等专业的专业技能抽查、单片机课程教学和培训等。

全书在编写过程中参考了一些国内外公开出版的文献资料及《应用电子技术》专业技能抽查题库要求等，在此特向原著作者表示敬意与感谢。

全部任务相关程序均已进行仿真与实际测试，但限于编者水平，书中难免存在错误和不足，恳请广大读者批评指正，以期在后续能够更加完善。

编者

目 录

M U L U

任务 1	按键控制彩灯软件开发	1
任务 2	计数器软件开发	11
任务 3	按键控制组合灯软件开发	20
任务 4	频率计软件开发	27
任务 5	示波器软件开发	35
任务 6	秒表软件开发	43
任务 7	倒计时秒表软件开发	50
任务 8	交通灯控制软件开发	58
任务 9	单片机数据接收	64
任务 10	单片机数据发送	73
任务 11	8×8LED 显示屏软件开发	84
任务 12	按键发音程序设计	93
任务 13	温度计程序设计	99

任务 14	时钟程序设计·····	114
任务 15	简易计算器程序设计·····	127
任务 16	六位密码锁程序设计·····	140
任务 17	啤酒生产计数器程序设计·····	163
任务 18	排球比分显示器程序设计·····	170
任务 19	反应时间测试仪程序设计·····	178
任务 20	篮球比赛计时计分器程序设计·····	185
附录 A	Keil 安装及使用·····	193
附录 B	Proteus 安装及使用·····	208
附录 C	开发板电路图·····	224
附录 D	开发板介绍·····	225

任务1

按键控制彩灯软件开发

一、任务

在下列开发板硬件基础上,编写完整程序实现如下模拟彩灯:开发板从左至右4个按键分别为S1~S4,从上至下8个灯为L1~L8,按下S1键8只LED小灯以1Hz频率闪烁,按下S2键8只灯奇偶交替点亮,间隔0.3s,按下S3键,L1~L4与L5~L8灯交替点亮,间隔0.2s,按下S4键熄灭所有灯。

二、要求

- (1) 根据提供的开发板硬件,分析系统功能,并画出系统主程序流程图。
- (2) 编写程序,程序代码要符合编程规范(函数/子程序名称、功能、入口参数、出口参数、注释等),易读性要好。
- (3) 编译调试、下载程序。
- (4) 调试系统达到功能指标。
- (5) 编写设计报告。

三、说明

- (1) 每个工位配有开发板硬件1套、51单片机开发工具一套件及相关工具。
- (2) 根据任务与要求的内容,编写程序,并下载调试,实现功能。
- (3) 系统上电前要仔细检查系统连线,拔插芯片等相关的操作必须停电后再进行。
- (4) 调试技术平台。

操作系统:WINDOWS XP;开发环境:Keil C51 uv2 750a;下载软件:STC-ISP;硬件:STC12C5A60S2单片机开发板。

- (5) 提供STC12C5A60S2等芯片的数据手册。
- (6) 遵循6S标准进行操作。

四、设计报告

功能分析

实验任务用到了STC单片机-USB下载学习开发板,所以必须掌握开发板电路结构,能够看懂开发板电路图并会对开发板下载调试。本实验用到了Keil和Proteus软件,所以需掌握Keil的工程建立,掌握Keil软件中芯片的选择和文件的添加,掌握C语言编程的

语法规则，会使用 Proteus 仿真软件，添加元器件并连线，加载 Keil 生成的 .hex 文件。本实验用到了定时器计数器，所以必须掌握一种定时器计时方式。本实验使用独立按键，因此还需掌握按键的扫描及确认，掌握如何用软件方法解决按键抖动的问题。

因为 $f_{osc} = 11.0592 \text{ MHz}$ ，机器周期 $= 12/11.0592 \text{ MHz} = 1.085 \mu\text{s}$ ，而定时/计数器在定时模式时就是对 f_{osc} 的 12 分频即机器周期进行计数，因此要定时 1ms 需要计数次数为 9216 次。为了使计算中断次数少而减少误差，初值应该尽量取值大一些，但模式 1 的 16 位寄存器的最大值为 65536，也就是说最大初值不能超过 65536，为了取整数计算方便我们这里取 46080，也就是 50 ms，要实现 1 Hz、0.3 s 和 0.2 s 进入中断次数分别为 10 次、6 次和 4 次。

注意：因大多数学习者还在使用 89C51 系列芯片，并且 Protues 软件中没有 STC12C5A60S2 芯片，以及本教学中没有过多涉及到 STC12C5A60S2 芯片的高级功能，所以在仿真中还是以 89C51 为例，只有在个别实物演示中的时序延时要求严格的实验中，重新计算时钟延时，对比 STC12C5A60S2 与 89C51 芯片的执行速度。

相关知识

1. STC12C5A60S2 介绍

STC12C5A60S2/AD/PWM 系列单片机是单时钟/机器周期 (1T) 的单片机，是高速/低功耗/超强抗干扰的新一代 8051 单片机，指令代码完全兼容传统 8051，但速度快 8~12 倍。内部集成 MAX810 专用复位电路，2 路 PWM，8 路高速 10 位 A/D 转换 (250 k/s)，针对电机控制强干扰场合。

特点：

- (1) 增强型 8051CPU，1T，单时钟/机器周期，指令代码完全兼容传统 8051。
- (2) 工作电压。STC12C5A60S2 系列工作电压：5.5 V ~ 3.3 V (5 V 单片机)；STC12LE5A60S2 系列工作电压：3.6 V ~ 2.2 V (3 V 单片机)。
- (3) 工作频率范围：0~35 MHz，相当于普通 8051 的 0~420 MHz。
- (4) 用户应用程序空间 8K/16K/20K/32K/40K/48K/52K/60K/62K 字节。
- (5) 片上集成 1280 字节 RAM。
- (6) 通用 I/O 口 (36/40/44 个)，复位后为：准双向口/弱上拉 (普通 8051 传统 I/O 口)，可设置成四种模式：准双向口/弱上拉，推挽/强上拉，仅为输入/高阻，开漏，每个 I/O 口驱动能力均可达到 20 mA，但整个芯片最大不要超过 120 mA。
- (7) ISP (在系统可编程) /IAP (在应用可编程)，无须专用编程器，无须专用仿真器，可通过串口 (P3.0/P3.1) 直接下载用户程序，数秒即可完成。
- (8) 有 EEPROM 功能 (STC12C5A62S2/AD/PWM 无内部 EEPROM)。
- (9) 内部集成 MAX810 专用复位电路 (外部晶体 12 M 以下时，复位脚可直接 1 K 电阻到地)。
- (10) 外部掉电检测电路：在 P4.6 口有一个低压门槛比较器，5 V 单片机为 1.32 V，误差为 $\pm 5\%$ ，3.3 V 单片机为 1.30 V，误差为 $\pm 3\%$ 。

(11) 时钟源：外部高精度晶体/时钟，内部 R/C 振荡器（温漂为 $\pm 5\%$ 到 $\pm 10\%$ 以内），用户在下载用户程序时，可选择使用内部 R/C 振荡器还是外部晶体/时钟，常温下内部 R/C 振荡器频率为：5.0 V 单片机为 11 ~ 15.5 MHz，3.3 V 单片机为 8 ~ 12 MHz，精度要求不高时，可选择使用内部时钟，但因为有制造误差和温漂，以实际测试为准。

(12) 共 4 个 16 位定时器：两个与传统 8051 兼容的定时器/计数器，16 位定时器 T0 和 T1，没有定时器 T2，但有独立波特率发生器，做串行通讯的波特率发生器，再加上 2 路 PCA 模块可再实现 2 个 16 位定时器。

(13) 2 个时钟输出口，可由 T0 的溢出在 P3.4/T0 输出时钟，可由 T1 的溢出在 P3.5/T1 输出时钟。

(14) 外部中断 I/O 口 7 路，传统的下降沿中断或低电平触发中断，并新增支持上升沿中断的 PCA 模块，Power Down 模式可由外部中断唤醒，INT0/P3.2，INT1/P3.3，T0/P3.4，T1/P3.5，RxD/P3.0，CCP0/P1.3（也可通过寄存器设置到 P4.2），CCP1/P1.4（也可通过寄存器设置到 P4.3）。

(15) PWM（2 路）/PCA（可编程计数器阵列，2 路）：

——也可用来当 2 路 D/A 使用；

——也可用来再实现 2 个定时器；

——也可用来再实现 2 个外部中断（上升沿中断/下降沿中断均可分别或同时支持）。

(16) A/D 转换，10 位精度 ADC，共 8 路，转换速度可达 250 k/s（每秒钟 25 万次）。

(17) 通用全双工异步串行口（UART），由于 STC12 系列是高速的 8051，可再用定时器或 PCA 软件实现多串口。

(18) STC12C5A60S2 系列有双串口，后缀有 S2 标志的才有双串口，RxD2/P1.2（可通过寄存器设置到 P4.2），TxD2/P1.3（可通过寄存器设置到 P4.3）。

(19) 工作温度范围：-40℃ ~ +85℃（工业级）/0℃ ~ 75℃（商业级）。

(20) 封装：当 PDIP-40，LQFP-44，LQFP-48，I/O 口不够时，可用 2 到 3 根普通 I/O 口线外接 74HC164/165/595（均可级联）来扩展 I/O 口，还可用 A/D 做按键扫描来节省 I/O 口，或用双 CPU，三线通信，串口相应增加。

2. 引脚功能

VCC：供电电压。

GND：接地。

P0 口：P0 口为一个 8 位漏级开路双向 I/O 口，每个管脚可吸收 8TTL 门电流。当 P1 口的管脚写“1”时，被定义为高阻输入。P0 能够用于外部程序数据存储器，它可以被定义为数据/地址的第八位。在 FLASH 编程时，P0 口作为原码输入口，当 FLASH 进行校验时，P0 输出原码，此时 P0 外部电位必须被拉高。

P1 口：P1 口是一个内部提供上拉电阻的 8 位双向 I/O 口，P1 口缓冲器能接收、输出 4 个 TTL 门电流。P1 口管脚写入“1”后，电位被内部上拉为高，可用作输入，P1 口被外部下拉为低电平时，将输出电流，这是由于内部上拉的缘故。在 FLASH 编程和校验时，P1 口作为第八位地址接收。

P2 口：P2 口为一个内部上拉电阻的 8 位双向 I/O 口，P2 口缓冲器可接收、输出 4 个 TTL 门电流。当 P2 口被写“1”时，其管脚电位被内部上拉电阻拉高，且作为输入。作为输入时，P2 口的管脚电位被外部拉低，将输出电流，这是由于内部上拉的缘故。P2 口当用于外部程序存储器或 16 位地址外部数据存储器进行存取时，P2 口输出地址的高 8 位。在给出地址“1”时，它利用内部上拉的优势，当对外部 8 位地址数据存储器进行读写时，P2 口输出其特殊功能寄存器的内容。P2 口在 FLASH 编程和校验时接收高 8 位地址信号和控制信号。

P3 口：P3 口管脚是 8 个带内部上拉电阻的双向 I/O 口，可接收、输出 4 个 TTL 门电流。当 P3 口写入“1”后，它们被内部上拉为高电平，并用作输入。作为输入时，由于外部下拉为低电平，P3 口将输出电流（ILL），也是由于上拉的缘故。P3 口也可作为 AT89C51 的一些特殊功能口，P3 口的第二功能，见表 1-1。

表 1-1 P3 口的第二功能

端口引脚	第二功能
P3.0	RXD（串口输入口）
P3.1	TXD（串口输出口）
P3.2	$\overline{\text{INT0}}$ （外部中断 0 输入线）
P3.3	$\overline{\text{INT1}}$ （外部中断 1 输入线）
P3.4	T0（定时器 0 外部输入）
P3.5	T1（定时器 1 外部输入）
P3.6	$\overline{\text{WR}}$ （外部数据存储器写选通信号输出）
P3.7	$\overline{\text{RD}}$ （外部数据存储器读选通信号输出）

同时 P3 口可为闪烁编程和编程校验接收一些控制信号。

RST：复位输入。当振荡器复位器件时，要保持 RST 脚两个机器周期的高平时间。

ALE/PROG：当访问外部存储器时，地址锁存允许的输出电平用于锁存地址的低位字节。在 FLASH 编程期间，此引脚用于输入编程脉冲。在平时，ALE 端以不变的频率周期输出正脉冲信号，此频率为振荡器频率的 1/6。因此它可用作对外部输出的脉冲或用于定时目的。然而要注意的是：每当用作外部数据存储器时，将跳过一个 ALE 脉冲。如想禁止 ALE 的输出可在 SFR 特殊功能寄存器的 8EH 地址上置 0。此时，ALE 只有在执行 MOVX，MOVC 指令时 ALE 才起作用。另外，该引脚被略微拉高。如果微处理器在外部执行状态 ALE 禁止，则置位无效。

PSEN：外部程序存储器的选通信号。在由外部程序存储器取址期间，每个机器周期 PSEN 两次有效。但在访问内部数据存储器时，这两次有效的 PSEN 信号将不出现。

EA/VPP：当 EA 端保持低电平时，访问外部 ROM；注意加密方式 1 时，EA 将内部锁定为 RESET；当 EA 端保持高电平时，访问内部 ROM。在 FLASH 编程期间，此引脚也用于施加 12V 编程电源（VPP）；

XTAL1：反向振荡放大器的输入及内部时钟工作电路的输入。

XTAL2：来自反向振荡器的输出。

软件流程图

按键控制彩灯软件流程，如图 1-1 所示。

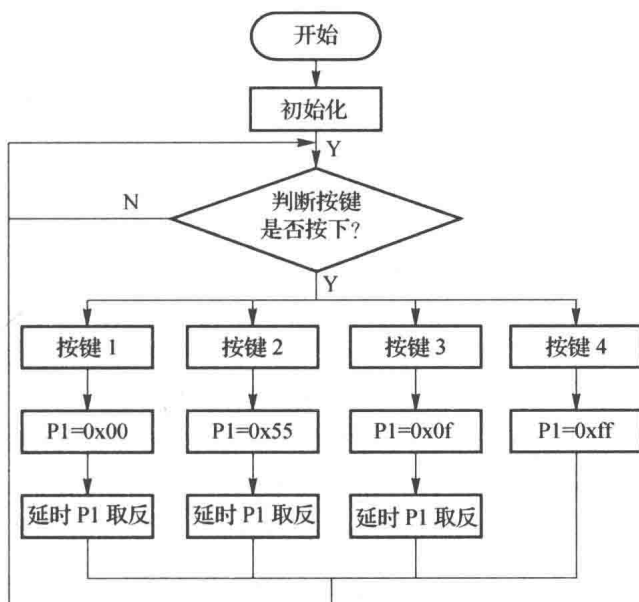


图 1-1 按键控制彩灯软件流程

程序代码



按键控制彩灯程序讲解视频

/* **** */

按键控制彩灯：编写完整程序实现如下模拟彩灯，开发板从左至右 4 个按键分别为 S1 ~ S4，从上至下 8 个灯为 L1 ~ L8，按下 S1 键 8 只 LED 小灯以 1 Hz 频率闪烁，按下 S2 键 8 只灯奇偶交替点亮，间隔 0.3 s，按下 S3 键，L1 ~ L4 灯与 L5 ~ L8 灯交替点亮，间隔 0.2 s，按下 S4 键熄灭所有灯。

晶振频率：11.059 2 MHz

/* **** */

```

#include <reg51.h>    //包含单片机寄存器的头文件
#define LED P1        //发光二极管接 P1
sbit S1 = P3^0;       //将 S1 位定义为 P3.0
sbit S2 = P3^1;       //将 S2 位定义为 P3.1
sbit S3 = P3^2;       //将 S3 位定义为 P3.2
sbit S4 = P3^3;       //将 S4 位定义为 P3.3
char flag = 0, q = 0; //flag 为延时变量标志，q 记录中断次数
  
```

/* **** */

函数功能：n 毫秒延时函数

入口参数: n 出口参数: 无

***** /

void delayms(unsigned int n)

```
{
    unsigned int j, k;
    for(j=0; j<n; j++)
        for(k=0; k<114; k++);
}
```

void main(void)

```
{
    TMOD = 0x01;           //使用定时器 T0 的模式 1
    TH0 = (65536 - 46080)/256; //定时器 T0 的高 8 位设置初值, 延时 50 ms
    TL0 = (65536 - 46080)%256; //定时器 T0 的低 8 位设置初值, 延时 50 ms
    EA = 1;                //开启总中断
    ET0 = 1;               //定时器 T0 中断允许
    while(1)
    {
        if(S1 == 0)        //如果按键 S1 按下
        {
            delayms(20);    //延时 20 ms 消抖
            if(S1 == 0)      //如果按键 S1 确实按下
            {
                while(!S1);  //等待按键抬起
                q = 0;        //中断计数次数清零
                flag = 1;     //标志为 1, 代表定时 1 s 的情况
                LED = 0xff;   //LED 全部关闭
                TR0 = 1;      //启动定时器 T0
            }

            //1 Hz 频率闪烁
        }
        if(S2 == 0)        //如果按键 S2 按下
        {
            delayms(20);    //延时 20 ms 消抖
            if(S2 == 0)      //如果按键 S2 确实按下
            {
                while(!S2);  //等待按键抬起
            }
        }
    }
}
```

```

        q = 0;                //中断计数次数清零
        flag = 2;            //标志为 2，代表定时 0.3 s 的情况
        LED = 0x55;          //1357 奇数位 LED 关闭，2468 偶数位 LED 点亮
                                01010101
        TR0 = 1;              //启动定时器 T0
    }
}
//0.3 s 间隔闪烁
if( S3 == 0)                //如果按键 S3 按下
{
    delayms(20);             //延时 20 ms 消抖
    if( S3 == 0)              //如果按键 S3 确实按下
    {
        while( ! S3);        //等待按键抬起
        q = 0;                //中断计数次数清零
        flag = 3;            //标志为 3，代表定时 0.2 s 的情况
        LED = 0x0f;          //低 4 位 LED 关闭，高 4 位 LED 点亮
        TR0 = 1;              //启动定时器 T0
    }
}
//0.2 s 高低 4 位间隔闪烁
if( S4 == 0)                //如果按键 S4 按下
{
    delayms(20);             //延时 20 ms 消抖
    if( S4 == 0)              //如果按键 S4 确实按下
    {
        while( ! S4);        //等待按键抬起
        q = 0;                //中断计数次数清零
        flag = 4;            //标志为 4，代表定时关闭的情况
        TR0 = 0;              //关闭定时器 T0
        LED = 0xff;           //LED 全部关闭
    }
}
}
}
}

```

/ *****

函数功能：定时器 T0 的中断服务函数

入口参数：n

出口参数：无

***** /

```
void Time0(void) interrupt 1 using 1    //定时器 T0 的中断编号为 1，使用第 1 组工作寄存器
```

```
{
```

```
    if(flag == 1)                    //延时变量为 1，代表定时 1 s 的情况
```

```
    {
```

```
        q ++ ;                      //累计次数
```

```
        if(q == 10)                 //如果中断次数计满 10 次即 500 ms
```

```
        {
```

```
            LED = ~LED;              //8 只 LED 小灯以 1 Hz 频率闪烁
```

```
            q = 0;                  //中断次数清零
```

```
        }
```

```
    }
```

```
    if(flag == 2)                    //延时变量为 2，代表定时 0.3 s 的情况
```

```
    {
```

```
        q ++ ;                      //累计次数
```

```
        if(q == 6)                  //如果中断次数计满 6 次即 0.3 s
```

```
        {
```

```
            LED = ~LED;              //8 只 LED 小灯以 0.3 s 间隔闪烁
```

```
            q = 0;                  //中断次数清零
```

```
        }
```

```
    }
```

```
    if(flag == 3)                    //延时变量为 3，代表定时 0.2 s 的情况
```

```
    {
```

```
        q ++ ;                      //累计次数
```

```
        if(q == 4)                  //如果中断次数计满 4 次即 0.2 s
```

```
        {
```

```
            LED = ~LED;              //8 只 LED 小灯以 0.2 s 间隔高低 4 位闪烁
```

```
            q = 0;                  //中断次数清零
```

```
        }
```

```
    }
```

```
    TH0 = (65536 - 46080)/256;        //定时器 T0 的高 8 位设置初值，延时 50 ms
```

```
    TL0 = (65536 - 46080)% 256;      //定时器 T0 的低 8 位设置初值，延时 50 ms
```

```
}
```

软件仿真

模拟彩灯仿真原理，如图 1-2 所示。仿真元器件，见表 1-2。



按键控制彩灯仿真视频

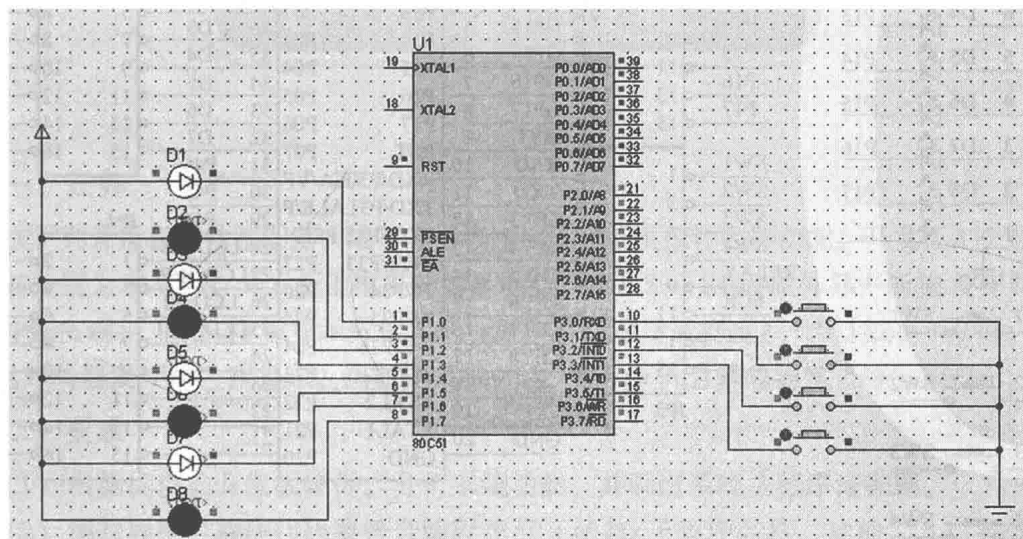


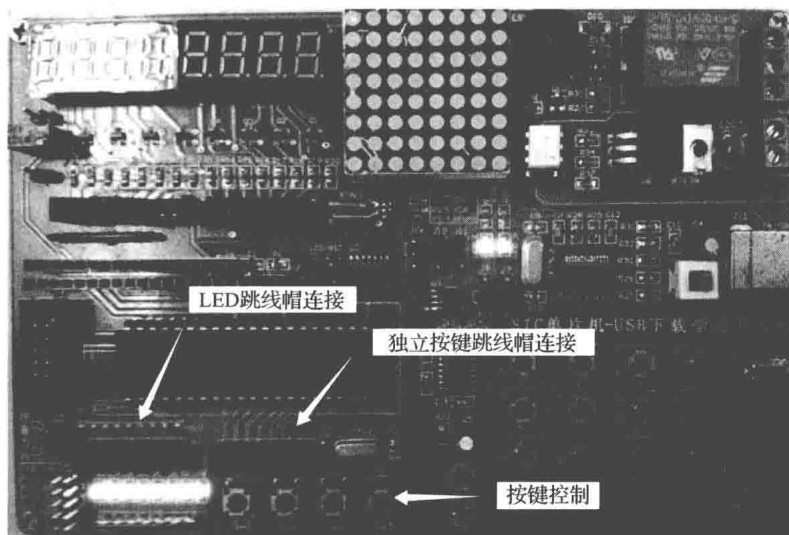
图 1-2 模拟彩灯仿真原理

表 1-2 仿真元器件

元器件名称	AT89C51	BUTTON	LED - BIBY
-------	---------	--------	------------

硬件调试

硬件调试的实物演示效果，如图 1-3 所示。实验电路原理，如图 1-4 所示。



实物演示视频1

图 1-3 实物演示效果

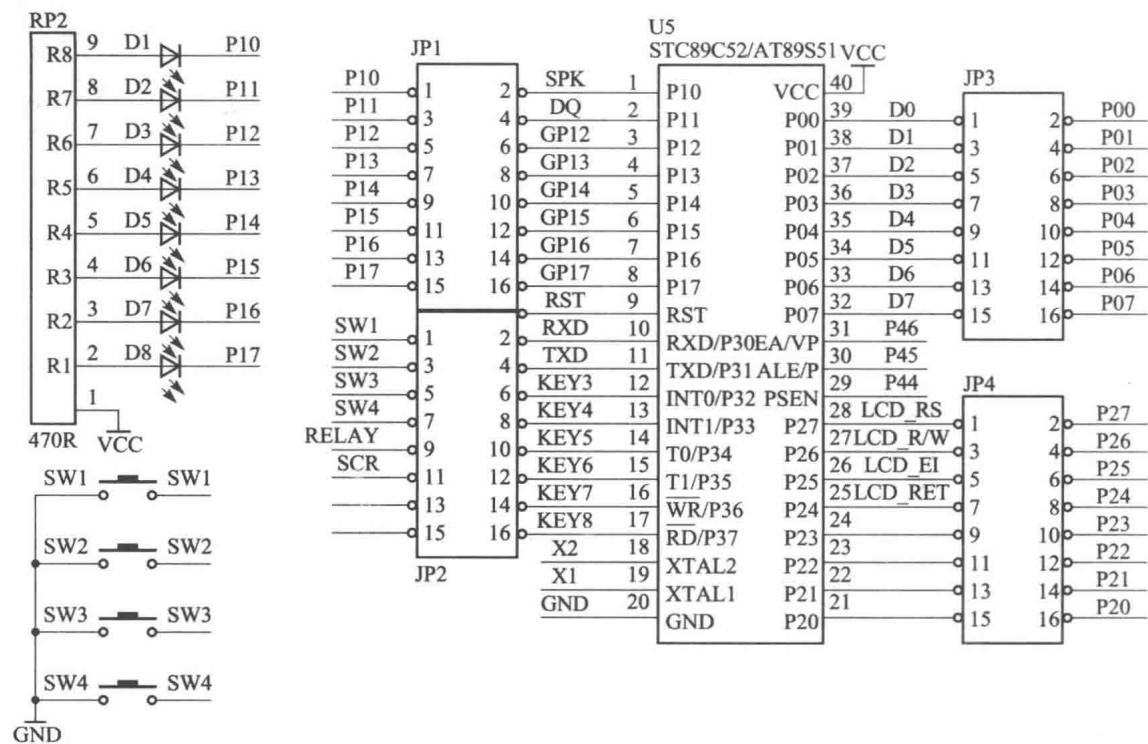


图 1-4 实验电路原理

任务实施

本次实验用到了 IO 口，P1 口作为输出，P3 口低四位作为输入。P1 口用跳线帽接 8 个 LED，P3 端口低四位接独立按键 S1、S2、S3、S4，用于调节彩灯的模式。Keil 程序中使用定时器 0 来设置彩灯闪烁的时间，当按键 S1 按下后，LED 灯以 500 ms 的时间闪烁，当按键 S2 按下后，LED 灯以 300 ms 的时间奇偶交替闪烁，当按键 S3 按下后，LED 灯以 200 ms 的时间高低位交替闪烁，当按键 S4 按下后，熄灭所有 LED 灯。按键彩灯闪烁显示正确，实现题目要求功能。

任务2

计数器软件开发

一、任务

在下列开发板硬件基础上，编写完整程序实现如下模拟计数器功能，开发板4个按键从左至右分别为S1~S4，开发板上电复位后显示0，利用S3、S4键做外部中断输入（要求用外部中断实现），按下S3键显示数据加1，按下S4键显示数据减1，当计数小于0时，显示0并且蜂鸣器发“嘟”声提示，当数据大于9时，显示9并且蜂鸣器发“嘟”声提示。

二、要求

- (1) 根据提供的开发板硬件，分析系统功能，并画出系统主程序流程图。
- (2) 编写程序，程序代码要符合编程规范（函数/子程序名称、功能、入口参数、出口参数、注释等），易读性要好。
- (3) 编译调试、下载程序。
- (4) 调试系统达到功能指标。
- (5) 编写设计报告。

三、说明

- (1) 每个工位配有开发板硬件1套、51单片机开发工具一套件及相关工具。
- (2) 根据任务与要求的内容，编写程序，并下载调试，实现功能。
- (3) 系统上电前要仔细检查系统连线，拔插芯片等相关的操作必须停电后再进行。
- (4) 调试技术平台。

操作系统：WINDOWS XP；开发环境：Keil C51 uv2 750a；下载软件：STC-ISP；硬件：STC12C5A60S2 单片机开发板。

- (5) 提供STC12C5A60S2等芯片的数据手册。
- (6) 遵循6S标准进行操作。

四、设计报告

功能分析

实验任务用到了STC单片机-USB下载学习开发板，所以必须掌握开发板电路结构，能够看懂开发板电路图并会对开发板下载调试。本实验用到了Keil和Proteus软件，所以需要掌握Keil的工程建立，掌握Keil软件中芯片的选择和文件的添加，掌握C语言编程的