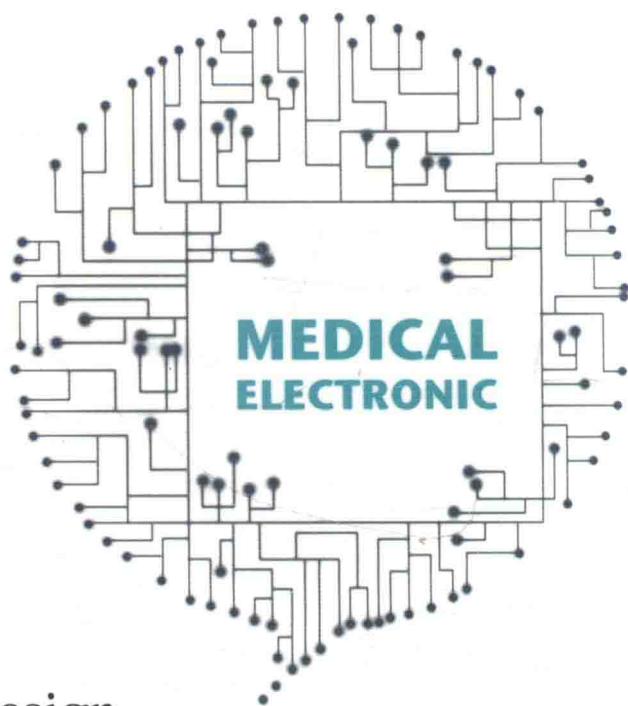




Electronic Circuit Design
Simulation
and Production

电子电路设计、仿真与制作

医用电子电路 设计及应用

◆ 周润景 邢 婧 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

Electronic Circuit Design Simulation and Production

电子电路设计、仿真与制作

常用控制电路设计及应用

常用驱动电路设计及应用

常用电源电路设计及应用

► 医用电子电路设计及应用

单片机技术及应用

常用传感器技术及应用



电子信息出版分社微博
<http://weibo.com/etpublish>



策划编辑: 张 剑
责任编辑: 夏平飞
封面设计: 徐海燕



官方微信平台

ISBN 978-7-121-31407-0



9 787121 314070 >

定价: 39.80 元

电子电路设计、仿真与制作

医用电子电路设计及应用

周润景 邢 婧 编著

電子工業出版社.

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书介绍了14个典型的医用电子电路设计案例,内容包含视觉疲劳检测与消除电路设计,精神压力自测电路设计,助听、催眠、增强记忆三用电路设计,声光电子催眠电路设计,视力保护测光电路设计,脉冲电疗电路设计,心率检测电路设计,血管弹性测量电路设计,呼吸测量电路设计,口吃矫正器电路设计,心音测量电路设计,电子治疗仪电路设计,心电信号显示检测仪电路设计,基于脉搏波的提取电路设计。这些案例均来源于作者多年的实际科研项目,因此具有很强的实用性。通过对本书的学习和实践,读者可以很快掌握常用医用电子电路设计的基础知识及应用方法。

本书适合电子电路设计爱好者自学使用,也可作为高等学校相关专业课程设计、毕业设计及电子设计竞赛的指导书籍。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

医用电子电路设计及应用/周润景,邢婧编著. —北京:电子工业出版社,2017.5

(电子电路设计、仿真与制作)

ISBN 978-7-121-31407-0

I. ①医… II. ①周… ②邢… III. ①医用电气机械—电子电路—电路设计—研究 IV. ①TH772
②TN702

中国版本图书馆CIP数据核字(2017)第085015号

策划编辑:张 剑(zhang@phei.com.cn)

责任编辑:夏平飞

印 刷:北京京师印务有限公司

装 订:北京京师印务有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编 100036

开 本:787×1092 1/16 印张:14 字数:335千字

版 次:2017年5月第1版

印 次:2017年5月第1次印刷

定 价:39.80元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888,88258888。

质量投诉请发邮件至 zlt@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:zhang@phei.com.cn。

前 言

本书通过一系列特定案例分析，向读者详细地介绍了医用电子电路的设计思路和方法，是一本独特的实用技术书籍。

全书介绍了 14 个电路设计，包括视觉疲劳检测与消除电路设计，精神压力自测电路设计，助听、催眠、增强记忆三用电路设计，声光电子催眠电路设计，视力保护测光电路设计，脉冲电疗电路设计，心率检测电路设计，血管弹性测量电路设计，呼吸测量电路设计，口吃矫正器电路设计，心音测量电路设计，电子治疗仪电路设计，心电信号显示检测仪电路设计，基于脉搏波的提取电路设计。书中详细介绍了电路的设计任务、基本要求、原理及具体的设计、PCB 设计等内容，并且用 Proteus 和 Multism 软件对电路进行了功能仿真，验证了电路工作的正确性，提高了电路开发效率，降低了开发成本，使读者能够更深刻地理解电路的工作原理，学习医用电子电路的设计方法。

书中多次出现了滤波电路、隔离电路、放大电路的设计，详细论述了生物电信号的放大、处理等多方面技术。电路的设计思路一般为：传感器采集生理信号，转换成电信号后信号比较微弱，易受干扰，所以设计隔离电路（电源隔离、信号隔离）、滤波电路（带阻滤波电路、带通滤波电路）来减少干扰，设计放大信号对电信号进行放大，以便后续的处理。14 个电路设计，从简单的视觉疲劳检测与消除电路设计，一直到较复杂的脉搏波提取电路设计，知识点涵盖模拟电路设计、单片机程序设计、芯片应用等，有助于读者全面认识医用电子电路的设计、制作与调试，启发读者的创新能力，提高读者的动手能力。书中内容简明扼要，适合想了解医用电子电路的读者阅读，也适合设计医用电子电路的读者作为参考。

本书的内容大多来自作者的科研与实践，有关内容的讲解并没有过多的理论推导，而代之以实用的电路设计，因此实用是本书的一大特点。

本书力求做到精选内容，推陈出新；讲清基本概念、基本电路的工作原理和基本分析方法。本书语言生动精炼，内容详尽，并且包含了大量可供参考的实例。

本书由周润景、邢婧编著。其中，邢婧编写了项目 1，其余项目由周润景编写，全书由周润景统稿。另外，参加本书编写的还有谢亚楠、丁岩、陈萌、井探亮、张赫、韩亦俚、刘艳珍、刘百灵、王洪艳、张红敏、张丽敏、周敬和宋志清。

由于医用电子电路涵盖内容非常广泛，不可能通过一本书就能介绍详尽，加上时间与水平有限，书中可能存在一些错误、遗漏和不妥之处，敬请读者批评指正。

作 者

目 录

项目 1 视觉疲劳检测与消除电路设计	1
设计任务	1
基本要求	1
设计思路	1
系统组成	1
电路原理图	2
模块详解	2
软件设计	4
调试与仿真	8
电路 PCB 设计图	10
实物测试	10
思考与练习	11
项目 2 精神压力自测电路设计	12
设计任务	12
基本要求	12
设计思路	12
系统组成	12
电路原理图	13
模块详解	13
Proteus 仿真	16
电路 PCB 设计图	17
实物测试	18
思考与练习	18
项目 3 助听、催眠、增强记忆三用电路设计	20
设计任务	20
基本要求	20
设计思路	20
系统组成	20
电路原理图	21
模块详解	21
电路 PCB 设计图	26
实物图	26

思考与练习	26
特别提醒	27
项目 4 声光电子催眠电路设计	28
设计任务	28
基本要求	28
设计思路	28
系统组成	28
电路原理图	28
模块详解	30
软件设计	36
电路 PCB 设计图	41
实物图	41
实际运行	42
思考与练习	42
项目 5 视力保护测光电路设计	44
设计任务	44
基本要求	44
设计思路	44
系统组成	44
电路原理图	45
模块详解	46
软件设计	48
Proteus 仿真	52
电路 PCB 设计图	55
实物测试	56
思考与练习	56
特别提醒	56
项目 6 脉冲电疗电路设计	57
设计任务	57
基本要求	57
设计思路	57
系统组成	57
电路原理图	58
模块详解	58
电路 PCB 设计图	66
实物测试	66
思考与练习	67

项目7 心率检测电路设计 68

设计任务 68

基本要求 68

设计思路 68

系统组成 68

电路原理图 69

模块详解 69

软件设计 78

电路PCB设计图 83

实物测试 84

思考与练习 84

特别提醒 84

项目8 血管弹性测量电路设计 85

设计任务 85

基本要求 85

设计思路 85

系统组成 85

电路原理图 86

模块详解 87

电路PCB设计图 92

实物测试 92

思考与练习 93

特别提醒 93

项目9 呼吸测量电路设计 94

设计任务 94

基本要求 94

设计思路 94

系统组成 94

电路原理图 95

模块详解 97

电路PCB设计图 110

实物测试 111

思考与练习 111

特别提醒 111

项目10 口吃矫正器电路设计 112

设计任务 112

基本要求 112

设计思路·····	112
系统组成·····	112
电路原理图·····	113
模块详解·····	114
电路 PCB 设计图 ·····	118
实物图·····	118
实际运行·····	118
思考与练习·····	119
特别提醒·····	119
项目 11 心音测量电路设计 ·····	120
设计任务·····	120
基本要求·····	120
设计思路·····	120
系统组成·····	120
电路原理图·····	121
模块详解·····	121
电路 PCB 设计图 ·····	127
实物测试·····	127
思考与练习·····	127
特别提醒·····	128
项目 12 电子治疗仪电路设计 ·····	129
设计任务·····	129
基本要求·····	129
设计思路·····	129
系统组成·····	129
电路原理图·····	130
模块详解·····	130
软件设计·····	137
电路 PCB 设计图 ·····	138
实物测试·····	139
思考与练习·····	139
项目 13 心电信号显示检测仪电路设计 ·····	140
设计任务·····	140
基本要求·····	140
设计思路·····	140
系统组成·····	140
模块详解·····	141

软件设计·····	164
系统仿真·····	187
项目 14 基于脉搏波的提取电路设计 ·····	189
设计目的·····	189
设计任务·····	189
设计思路·····	189
系统组成·····	190
模块详解·····	190
软件设计·····	203
电路原理图·····	207
系统仿真·····	209
实物图·····	210
思考与练习·····	211
特别提醒·····	211

项目 1 视觉疲劳检测与消除电路设计



设计任务

设计一个简单的人体视觉疲劳检测与消除电路，通过电路可以初步检测人眼疲劳和初步消除视觉疲劳，同时还可以预防因视觉疲劳而造成的近视等问题。



基本要求

- ☺ 能够使人检测出自我处于视觉疲劳状态。
- ☺ 在检测出人眼处于视觉疲劳的状态时，能够初步消除视觉疲劳。



设计思路

人的眼睛有一个重要的被称为“视觉暂留特性”（也称“视觉惰性”）的现象，即人眼在观察景物时，光信号传入大脑神经，需经过一段短暂的时间，光的作用结束后，视觉形象并不立即消失，这种残留的视觉称“后像”，视觉的这一现象则被称为“视觉暂留”。利用人眼的“视觉暂留特性”，如果人眼处于视觉疲劳的状态，那么 LED 在一定频率下闪烁，人眼就看不到灯光的闪烁。而利用四个 LED 灯依次闪烁，人眼随着 LED 灯闪烁的方向转动，可消除视觉疲劳。本设计通过对 AT89C51 单片机编程来控制 LED 灯的闪烁以及灯的闪烁频率，从而初步实现视觉疲劳检测和消除。



系统组成

视觉疲劳检查与消除系统主要分四个部分。

- ☺ 电源模块：为整个电路提供 +5V 的电压。
- ☺ AT89C51 单片机最小系统。
- ☺ 频率调节按键：长按可以切换工作区域，点按可以调节大灯的频率。
- ☺ LED 模块。



电路原理图（见图 1-1）

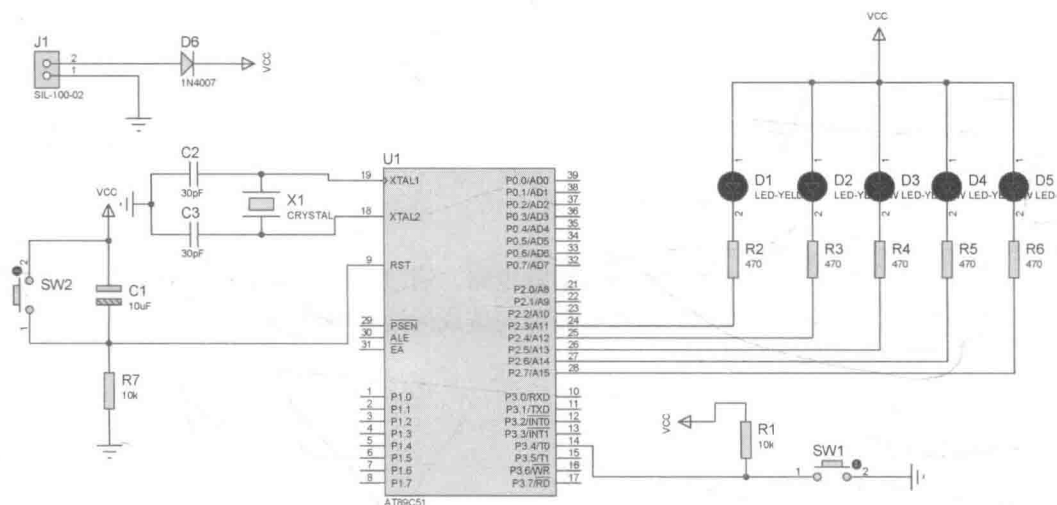


图 1-1 电路原理图



模块详解

1. 电源模块

电源电路如图 1-2 所示。其输出为 +5V 的电压，由 1N4007 整流二极管承受较强的浪涌，使得输出电压稳定。

1N4007 参数：

- ☺ 较强的正向浪涌承受能力：30A。
- ☺ 最大正向平均整流电流：1.0A。
- ☺ 极限参数为 $VRM \geq 50V$ 。
- ☺ 最高反向耐压：1000V。
- ☺ 最大反向漏电流： $5\mu A$ 。
- ☺ 正向压降：1.0V。
- ☺ 最大反向峰值电流： $30\mu A$ 。
- ☺ 典型热阻： $65^{\circ}C/W$ 。
- ☺ 典型结电容：15pF。
- ☺ 工作温度： $-50 \sim 150^{\circ}C$ 。

2. 调节按键模块

调节按键电路如图 1-3 所示，通过程序和按键来选择工作区域，调节 LED 灯闪烁的频率，长按频率调节按键可以通过程序来改变工作区域，点按可以控制 LED 灯在 10 ~ 60Hz 区间工作。

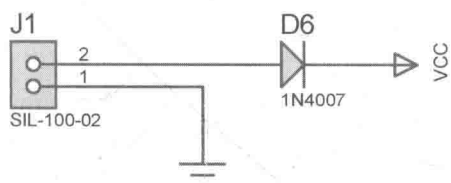


图 1-2 电源电路

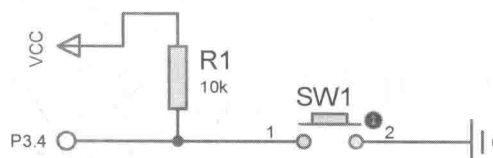


图 1-3 调节按键电路

3. LED 模块 (见图 1-4)

5 个 LED 灯分别接到单片机 P2.3、P2.4、P2.5、P2.6、P2.7 引脚，5V 供电，低电平驱动。

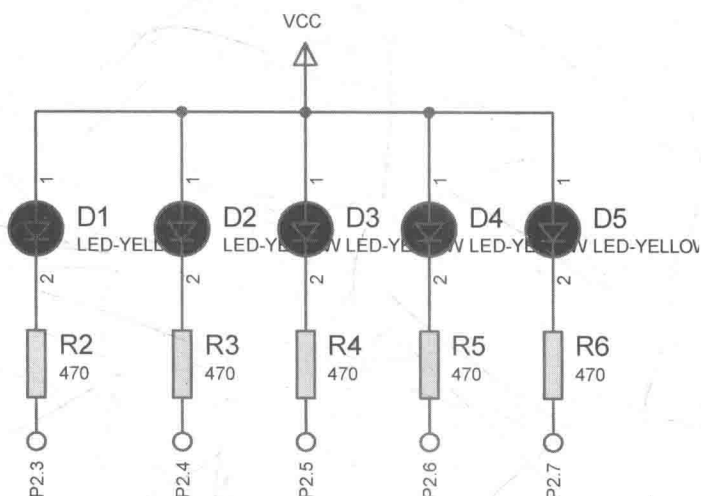


图 1-4 LED 电路

4. 单片机最小系统 (见图 1-5)

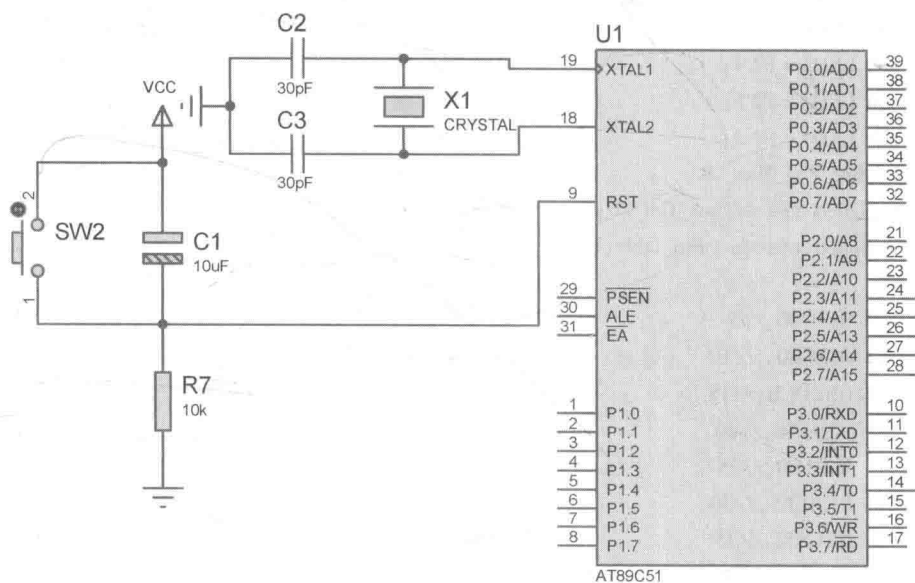


图 1-5 单片机最小系统电路



软件设计

本设计中，软件解决的主要问题是通过按键控制大 LED 灯和小 LED 灯之间的转换以及改变灯闪烁的频率。程序设计流程如图 1-6 所示。

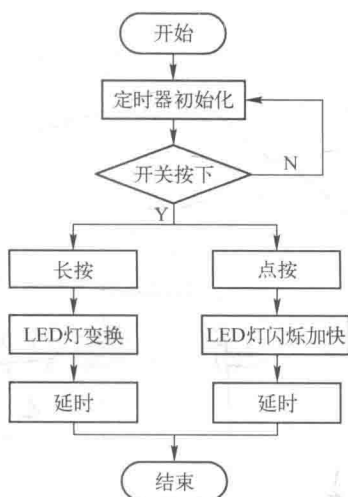


图 1-6 程序设计流程

程序：

```
#include <reg52. h> //包含头文件,一般情况不需要改动,头文件包含特殊功能寄存器的定义
sbit KEY_0 = P3^4; //定义 IN 输入端口
sbit LED_0 = P2^7; //定义 OUT 输出端口
sbit LED1 = P2^6; //定义 OUT 输出端口
sbit LED3 = P2^4; //定义 OUT 输出端口
sbit LED2 = P2^5; //定义 OUT 输出端口
sbit LED4 = P2^3; //定义 OUT 输出端口
#define Frq_Max 8
unsigned char ucTime_Get = 0; //现在的频率
unsigned int code Frq_table[8] =
{
    0x3CB0, //5
    0x3CB0, //10
    0X7DCB, //15
    0X9E58, //20
    0xB1E0, //25
    0XBEE5, //30
    0XC832, //35
    0XCF2C, //40
};
```

```

/* ----- 定时器初始化子程序 ----- */
void Init_Timer0( void)
{
    TMOD|=0x01;    //使用模式 1,16 位定时器,使用“|”符号可以在使用多个定时器时不受影响
    TH0 = ( 65536 - 50000)/256;    //赋值 50ms
    TL0 = ( 65536 - 50000)%256;
    EA = 1;    //总中断打开
    ET0 = 1;    //定时器中断打开
    TR0 = 1;    //定时器开关打开
    PT1 = 1;
}

void Init_Timer1( void)
{
    TMOD |= 0x10;    //使用模式 1,16 位定时器,使用“|”符号可以在使用多个定时器时不受影响
    TH1 = ( Frq_table[ ucTime_Get ])/256;
    TL1 = ( Frq_table[ ucTime_Get ])%256;
    ET1 = 1;    //定时器中断打开
    TR1 = 1;    //定时器开关打开
}

unsigned char ucKEY_Down_Flag=0;
unsigned char ucDis_Mode=0;
void Time_Change( void)
{
    if( ( ucKEY_Down_Flag&0x80) == 0x00)    //没有按键按下
    {
        if( KEY_0 == 0)
        {
            ucKEY_Down_Flag ++ ;
            if( ucKEY_Down_Flag&0x40 == 0x00)
            {
                if( ucKEY_Down_Flag >= 14)
                {
                    ucKEY_Down_Flag|=0x40;
                }
            }
            else if( ucKEY_Down_Flag&0x40 == 0x40)
            {
                if( ucKEY_Down_Flag >= 100) //长按
                {
                    ucKEY_Down_Flag = 0x80;
                    if( ucDis_Mode == 0)
                    {
                        ucDis_Mode = 1;
                        LED1 = 1;
                        LED2 = 1;
                    }
                }
            }
        }
    }
}

```

```

        LED3 = 1;
        LED4 = 1;
    }
    else
    {
        LED_0 = 1;
        ucDis_Mode = 0;
    }
}
}
else
{
    if(ucKEY_Down_Flag & 0x40 == 0x40) //短按
    {
        if(ucDis_Mode != 0)
        {
            ucTime_Get ++;
            if(ucTime_Get >= Frq_Max)
            {
                ucTime_Get = 0;
            }
        }
        ucKEY_Down_Flag = 0;
    }
    else if((ucKEY_Down_Flag & 0x80) == 0x80) //没有按键按下
    {
        if(KEY_0 == 1)
        {
            ucKEY_Down_Flag = 0;
        }
    }
}
}
/* -----
主程序
----- */
main()
{
    Init_Timer0();
    Init_Timer1();
    EA = 1;
    while(1);
}

```

```

/* -----
定时器中断子程序
----- */

char  cRand_Val = 0;
char  cStep = 0;
char  cTime_Total = 0;
void Timer0_isr( void) interrupt 1 using 1
{
    TH0 = ( 65536 - 5000 ) / 256;
    TL0 = ( 65536 - 5000 ) % 256;
    Time_Change ( ) ;
    if( ucDis_Mode == 0 )
    {
        if( ++ cTime_Total >= 60 )
        {
            cTime_Total = 0;
            cRand_Val = TH1 % 10;
            if( cRand_Val > 3 )
            {
                cRand_Val /= 3;

                switch( cRand_Val)           // 1 1 1 1 1 1 1 1
                {
                    case 0:                //LED1  P2. 6
                        LED1 = 0;
                        LED2 = 1;
                        LED3 = 1;
                        LED4 = 1;
                        break;
                    case 1:                //LED1  P2. 5
                        LED1 = 1;
                        LED2 = 0;
                        LED3 = 1;
                        LED4 = 1;
                        break;
                    case 2:                //LED1  P2. 4
                        LED1 = 1;
                        LED2 = 1;
                        LED3 = 0;
                        LED4 = 1;
                        break;
                    case 3:                //LED1  P2. 3
                        LED1 = 1;
                        LED2 = 1;
                        LED3 = 1;

```