



“冯如杯” 创意大赛

保障睡眠的多用仿自然眼罩与震动手环

摘 要

在一些集体生活中，尤其是先现在高校园宿舍的生活中，到深夜时常存在室外环境十分嘈杂，或者室友到深夜会开灯开电脑进行一些活动，这些声与光带来的干扰将会严重破坏正常时间睡眠的人的睡眠环境。为了能保障这部分人群的睡眠质量，本文设计了一款新式的睡眠保障设备。

本文是单片机技术的应用，设计了一款新式的仿自然眼罩和睡眠振动手环，眼罩与手环均用 89C52 单片机编程进行控制，并建立了一套极有可行的睡眠保障方案。仿自然眼罩平时可以当电子钟使用，睡觉时可以当做眼罩使用，阻挡光线，避免光干扰，在人苏醒前又可模拟太阳升起的自然现象；震动手环既可以当无声闹钟将人唤醒，又可以当做随身定时器使用；再配合使用市场上销售的隔音耳塞，即可全方位的保障睡眠。本文也对 LED 灯模拟太阳升起等技术进行了多方案讨论。本文论述项目是一款较有新意且较为实用的新式多用途睡眠保障产品。

关键字：眼罩，手环，单片机，仿自然

Abstract

Summaries in a number of communal life, especially now in college dorm life, late into the night often exist outside environment is very noisy, or roommate late at night will turn the light or turning on the computer for some activities, of these acoustic and optical interference connect people who can seriously disrupt normal sleep sleep environment. In order to ensure this part of the crowd sleep quality, this paper designs a new sleep support equipment.

This is the application of single-chip microcomputer, designed a new style of natural vibration of goggles and sleep-like bracelet, goggles and bracelet are made of 89C52 single-chip control program, and the establishment of a highly viable sleep protection programme. Imitates the natural eye usually can be used when the clock, sleep can be used as eye, blocking light, avoid interference, in person before it can simulate natural phenomena of the Sun; the earthquake started to ring either when the silent alarm clock will wake people, and can be used as timer with; then noise ear plugs used on the market, to all-round protection of sleep. This also LED light simulation technical discussions conducted several programmes, such as the Sun. Projects are discussed in this article is a relatively new and more practical new type multi-purpose sleep product.

Keywords : goggle, bracelet, SCM, imitating the nature, multi-purpose

目录

1 产品概述.....	1
1.1 产品需求分析.....	1
1.2 产品主体设计.....	2
1.2.1 眼罩的设计.....	2
1.2.2 震动手环的设计.....	2
2 产品关键技术.....	3
2.1 眼罩 LED 灯的设计.....	3
2.1.1 方案一 利于变阻器的调节。.....	3
2.1.2 方案二 利用点亮的灯泡数量调节.....	3
2.1.3 方案三 利用 PWM（Pulse Width Modulation）来控制 LED 灯泡的亮度。.....	3
2.2 电子钟计时和倒计时基本原理.....	4
3 优化设计.....	5
总结.....	6
附录 1.....	7
附录 2.....	8

1 产品概述

1.1 产品需求分析

人的一生约 1/3 是在睡眠中度过的，睡眠对人来说，就像饮食一基本生理需要。人们的睡眠与觉醒原生态是日出而作，日落而息。如果长期无法有规律地进行休养生息，就可引起很多身心问题，甚至罹患疾病。

但是在一些集体生活中，尤其是在高校园的宿舍生活中，很多人的睡眠无法得到保障。例如对于大学生，12 点以后也时常会遇到有人在宿舍门外打三国杀，大声吆喝；或者舍友正在看电影或打着游戏；或者室友在挑灯夜战，努力学习……这些声与光的干扰都会使人难以入眠。现在市面上销售的用于保护睡眠的有普通的眼罩和耳塞，但这些对于一个大学生来讲并不很合适。一方面，眼罩会从晚到早完全遮蔽光芒，使人无法感受到太阳升起的自然现象，与人日出而作，日落而息的习惯相违背，会渐渐打乱人体的生物钟；另一方面，隔音耳塞会使人与外界的声音隔绝，使其无法听到闹钟的声音，从而无法按时起床，扰乱第二天的各种学习生活的安排。

而本项产品就十分适合大学生。仿自然眼罩在预设时间 30 分钟前，眼罩内部的 LED 灯会缓慢亮起，模拟太阳升起的效果；由于佩戴了隔音耳塞，与外界声音基本隔绝，震动手环则可以按时将人震醒而又不会打扰到其他室友。而且眼罩和耳塞又有多种用途。眼罩平常时上面的液晶显示屏会显示年月日等时间，可以作为电子时钟摆放使用；睡觉时，眼罩上的显示屏还会显示使用者所设定的起床倒计时，能够让舍友随时知道使用者何时会起床。震动手镯则合适于在自习室、会议室等地方使用，且不会打扰到其他人。根据不同的情况可以把仿自然眼罩、震动手环和隔音耳塞选择性搭配使用，十分灵活。这些多种功能的设计和时尚的外形设计（见后面示意图图 1、图 2），又十分符合大学生时尚而又实用的心理。所以，本项产品势必会受到大学生和其他需求者的青睐。

1.2 产品主体设计

1.2.1 眼罩的设计

如图 1，该图是用 Photoshop 制作的产品示意图，眼罩可以基于普通的眼罩进行改进，



图 1 眼罩示意图

眼罩最外面是一块液晶显示屏，眼罩内部安装有 89C52 单片机及相应电路，点罩内部装有 LED 灯，液晶显示屏和 LED 灯均有 89C52 单片机控制。

液晶显示屏上有两种显示方式，可以互相切换。平时可以显示时间，可以作为电子钟摆放使用；睡眠使用时可以作为定时装置，并在预设时间前半个小时的时候，眼罩内侧的 LED 灯会逐渐的由暗到亮，

模拟太阳升起的效果。

眼罩左边的两个按钮和普通电子运动手表的下面两个按钮功能相似，可以调整时间。即一个按钮可以选择所需调整的目标——时、分、秒等，另一个按钮以每按一下数字大小向上加一的形式调节时间。所以左边的两个按钮负责调准电子钟的时间。眼罩右上方的按钮负责切换电子钟和睡眠倒计时装置的界面，右下方的按钮负责选定本次睡眠时常，5 小时起，每按一次增加半个小时，较长时间按住即可选定时间，开始进行起床倒计时，剩余半个小时的时候 LED 灯就会缓慢亮起，模拟太阳升起，再次按动右上方按钮，切回电子钟界面，且 LED 灯熄灭。

1.2.2 震动手环的设计

图 2 为震动手环的模拟示意图，震动手环主要由塑胶制成，材质较为柔软，佩戴舒适。



图 2 手环示意图

手环同眼罩一样，是由 89C52 单片机控制。手环中间是可以显示倒计时的液晶显示屏，椭圆处内部装有由单片机控制的振动器，手环左边按钮为手环开关，右边按钮与眼罩右下方按钮功能相同，

可以进行对起床倒计时的设置——时间从半小时起，每按一下增加半个小时，长时间按住选定时间，开始倒计时，时间一到则开始震动，按动左边的开关按钮即可停止震动。

2 产品关键技术

2.1 眼罩 LED 灯的设计

眼罩安装可以模拟太阳升起的 LED 灯，在预设起床时间的前半个小时的时候逐渐从昏暗到明亮。采用这种仿自然的方式，可以使使用者更容易在预设时间内苏醒，因为模仿日光符合人体的生物钟，这样又可以有效的减少醒后的困意，使使用者起床后更快的进入正常的生活状态。

关于对 LED 灯的灯泡亮度的调节有如下三种设计方案。

2.1.1 方案一 利于变阻器的调节。

经过单片机的学习和网上查阅资料，发现利用 89C52 单片机调节变阻器较为麻烦，所占体积较大，若利用其他方式也会使眼罩体积大幅度增大，不利于佩戴使用，所以否决了这一方案。

2.1.2 方案二 利用点亮的灯泡数量调节

眼罩每侧设置十个小 LED 灯，每隔 3 分钟点亮一个，直到到达预设时间三分钟的时候 10 个小灯泡就已全部点亮，达到最大亮度。利用同时点亮的小灯牌数量的增加，使总的灯泡亮度增加，从而达到模拟太阳升起的目的。但经过模拟测试，这种做法 LED 灯的亮度变化不均匀，例如在一个灯泡点亮的时候，就会出现中间较亮而旁边较暗的现象，突然的亮光会造成对眼部的不适，从而达不到理想的效果。

2.1.3 方案三 利用 PWM 来控制 LED 灯泡的亮度。

PWM（Pulse Width Modulation）是指脉冲宽度调制技术，通过对一系列脉冲的宽度进行调制，来等效地获得所需要波形（含形状和幅值）。

介绍一下大体思路，这种做法是通过控制单位时间内 LED 灯泡亮光的时间来控制 LED 灯的灯泡亮度。举个例子，根据眼罩的需求，我可以把亮灯分为十个等级，第一个等级最暗，第十个

等级最亮，每过 3 分钟增加一个等级。在第一个等级的时候，灯泡在一毫秒内点亮 0.1 毫秒，熄灭 0.9 毫秒，即灯泡有十分之一的时间处于点亮状态；同理，在第二个等级时，灯泡在一毫秒内点亮 0.2 毫秒，熄灭 0.8 毫秒，灯泡有十分之二的的时间处于点亮状态……直到第十个等级，灯泡处于全亮状态。这样可以达到使 LED 灯缓慢变亮的效果。

89C52 单片机的晶振频率是 11.0592MHZ，可以理解为在 1 秒内 89C52 单片机会完成 11.0592M 次运算，也即每完成 11.0592M 次运算花费的时间为 1s。依据这个原理，可以经过计算编如下的函数：

```
void delay(uint z)
{
    uint x,y;
    for(x=z;x>0;x--)
        for(y=110;y>0;y--);
}
```

那么对于这个函数，经过计算 z 每加 1，延长时间就能够多 1ms，即赋予 z 的值为持续时间的毫秒数。这样就给算法一个时间依据，利用 PMW 来控制 LED 灯泡的亮度的完整编程见附录 1。在此算法中，可以计算得出，每次的亮光周期为 1.85 毫秒。

而且在方案二中，没控制一组灯泡需要单片机的一个 IO 口，每眼对应十只灯泡则需十个 IO 口；但方案三中，对所有灯泡的控制只需要一个 IO 口，电路连接较为简便。

综合讨论以上三种方案，最终确定方案三为最终方案。

2.2 电子钟计时和倒计时基本原理

详细的电子钟计时编程见附录 2，该程序整理自郭天祥 51 单片机的教学视频。下面是简易的电子钟编程基本原理，即秒位值每加到 60，秒位清零，分位进一，显示时秒位会从 59 变为 0；同理，分位每加到 60，分位清零，时位进一，依次类推。秒位进一的时间间隔可以用上面提到的 void delay(uint z)函数进行确定，1s 时间即为 delay（1000）。

电子时钟计时原理基本思路如下：

```
if(miao==60)                //秒每到 60，秒清零，分加 1
{
    miao=0;
```



```
fen++;  
  
if(fen==60)           //分每到 60，分清零，时加 1  
{  
    fen=0;  
    shi++;  
    if(shi==24)        //时每到 24，时清零  
    {  
        shi=0;  
    }  
    write_sfm(4,shi); //  
}  
write_sfm(7, fen);  
}
```

相应的倒计时原理与正常计时的原理相反，且更为简单，编程思路如下：

```
If(miao==0)           //每次计时完秒后，秒回到 59，分减 1  
{  
    miao=59;  
    fen--;  
    if(fen==0)         //每次计时完分后，分回到 59，秒减 1  
    {  
        fen=59;  
        shi--;  
    }  
}
```

通过编程可以设置，当 `shi==0&&fen==0&&shi==0` 时，也即倒计时结束时单片机控制振动器震动。

3 优化设计

根据上述方案，在以下方面可以做出改进：

(1) . 眼罩和手环的显示屏可以在需要时选择性开启和关闭, 减少不看显示屏时的电量耗费;

(2) . 在眼罩中对 LED 灯亮度的控制方法加以改进, 寻找用变阻器控制灯泡亮度的新思路, 达到使灯泡亮度更柔和更均匀变化的效果;

(3) . 为眼罩添加应急灯功能, 眼罩 LED 灯可以手控开启, 如在在停电时或半夜起床时可以当临时应急的照明设施;

(4) . 优化定时算法, 使起床时间可以定为任意时刻;

(5) . 眼罩中一个按钮可以控制多种功能, 减少按钮个数, 可参照现代的电子运动表的模式进行编程设置;

(6) . 为眼罩添加更多的功能, 通过按钮切换功能, 如添加计时器功能。

对于这一系列的设想, 我们仍需更加深入的学习单片机相关的知识, 书写相关算法, 从而解决这些问题。

总结

创意提出以来, 我们经过两个月的时间查阅相关资料, 学习单片机方面的相关知识, 设计了上述比较具体的方案, 理论上基本可以解决睡眠保障的问题。虽然我们付出了大量努力, 但作品在一些地方肯定还会有一些不足, 我们会继续学习单片机方面的知识, 努力改进现有方案, 并争取在一年之内将创意变为现实, 做出成品, 并逐步走向市场。

附录 1

控制 LED 灯亮度调节的编程

```
#include<reg52.h>
#define uint unsigned int
#define uchar unsigned char
uchar ucT0;
uchar PWM=0;
bit Flag=1;
sbit LED=P0^0;
void Timer0_Init(void);
void delay(uint z)
{
    uint x,y;
    for(x=z;x>0;x--)
        for(y=110;y>0;y--);
}
void main()
{
    ET0=1;
    TMOD=0x02;
    TH0=0x47;
    TL0=0x47;
    TR0=1;
    EA=1;
    while(1)
    {
        delay(1000);           //delay 后面为 1000，则延迟为 1s，即每一秒亮度增加
        if(Flag)               //一次，若使其每 3 分钟增加一次亮度，则只需把
            PWM++;             //delay(1000)改为 delay(180000)
        Else
            PWM--;
        if(PWM>=10) Flag=0;
        if(PWM==0) Flag=1;
    }
}

void Timer0(void) interrupt 1
{
    ucT0++;
    if(ucT0==10)
    {
        ucT0=0;
        if(PWM!=0)
```

```
        LED=0;
    }
    if(ucT0==PWM)
    {
        LED=1;
    }
}
```

附录 2

单片机电子钟编程（整理自郭天祥 51 单片机教学视频）

```
#include<reg52.h>
#define uchar unsigned char
#define uint unsigned int
sbit dula=P2^6;
sbit wela=P2^7;
sbit rs=P3^5;
sbit lcden=P3^4;
sbit s1=P3^0;
sbit s2=P3^1;
sbit s3=P3^2;
sbit rd=P3^7;
uchar count,s1num;
char miao,shi,fen;           //定义时、分、秒
uchar code table[]=" 2007-7-30 MON";
uchar code table1[]=" 00:00:00";
void delay(uint z)
{
    uint x,y;
    for(x=z;x>0;x--)
        for(y=110;y>0;y--);
}

void write_com(uchar com)
{
    rs=0;
    lcden=0;
    P0=com;
    delay(5);
    lcden=1;
    delay(5);
    lcden=0;
}
```

```
void write_date(uchar date)
{
    rs=1;
    lcden=0;
    P0=date;
    delay(5);
    lcden=1;
    delay(5);
    lcden=0;
}

void init()
{
    uchar num;
    dula=0;
    wela=0;
    lcden=0;
//    fen=59;
//    miao=53;
//    shi=23;
    write_com(0x38);
    write_com(0x0c);
    write_com(0x06);
    write_com(0x01);
    write_com(0x80);
    for(num=0;num<15;num++)
        {
            write_date(table[num]);
            delay(5);
        }
    write_com(0x80+0x40);
    for(num=0;num<12;num++)
        {
            write_date(table1[num]);
            delay(5);
        }
    TMOD=0x01;
    TH0=(65536-50000)/256;
    TL0=(65536-50000)%256;
    EA=1;
    ET0=1;
    TR0=1;
}

void write_sfm(uchar add,uchar date)
{
    uchar shi,ge;
    shi=date/10;
    ge=date%10;
    write_com(0x80+0x40+add);
    write_date(0x30+shi);
    write_date(0x30+ge);
}
```

```
}  
void keyscan()  
{   rd=0;  
  if(s1==0)  
  {   delay(5);  
    if(s1==0)  
    {   s1num++;  
      while(!s1);  
      if(s1num==1)  
      {   TR0=0;  
        write_com(0x80+0x40+10);  
        write_com(0x0f);  
      }  
    }  
    if(s1num==2)  
      write_com(0x80+0x40+7);  
    if(s1num==3)  
      write_com(0x80+0x40+4);  
    if(s1num==4)  
    {   s1num=0;  
      write_com(0x0c);  
      TR0=1;  
    }  
  }  
  if(s1num!=0)  
  {   if(s2==0)  
    {   delay(5);  
      if(s2==0)  
      {   while(!s2);  
        if(s1num==1)  
        {   miao++;  
          if(miao==60)  
            miao=0;  
          write_sfm(10,miao);  
          write_com(0x80+0x40+10);  
        }  
        if(s1num==2)  
        {  
          fen++;  
          if(fen==60)  
            fen=0;  
          write_sfm(7,fen);  
          write_com(0x80+0x40+7);  
        }  
      }  
    }  
  }  
}
```

```
        if(s1num==3)
        {
            shi++;
            if(shi==24)
                shi=0;
            write_sfm(4,shi);
            write_com(0x80+0x40+4);
        }
    }
}
if(s3==0)
{
    delay(5);
    if(s3==0)
    {
        while(!s3);
        if(s1num==1)
        {
            /* if(miao==0)
            {
                miao=59;
                write_sfm(10,miao);
                write_com(0x80+0x40+10);
            }*/
            miao--;
            if(miao==-1)
                miao=59;
            write_sfm(10,miao);
            write_com(0x80+0x40+10);
        }
        if(s1num==2)
        {
            fen--;
            if(fen==-1)
                fen=59;
            write_sfm(7,fen);
            write_com(0x80+0x40+7);
        }
        if(s1num==3)
        {
            shi--;
            if(shi==-1)
                shi=23;
            write_sfm(4,shi);
```

```
        write_com(0x80+0x40+4);
    }
}

}

}

void main()
{
    init();
    while(1)
    {
        keyscan();
    }
// while(1);
}
void timer0() interrupt 1
{
    TH0=(65536-50000)/256;
    TL0=(65536-50000)%256;
    count++;
    if(count==18)
    {
        count=0;
        miao++;
        if(miao==60)                //秒每到 60，秒清零，分加 1
        {
            miao=0;
            fen++;
            if(fen==60)              //分每到 60，分清零，时加 1
            {
                fen=0;
                shi++;
                if(shi==24)          //时每到 24，时清零
                {
                    shi=0;
                }
                write_sfm(4,shi);    //
            }
            write_sfm(7,fen);
        }
        write_sfm(10,miao);
    }
}
```