信工专业单片机实验

信工 1602 2016014485 缪蓉

题目：

使用 STC 单片机上的红外接收器和配套的红外遥控器，实现对 STC 单片机实验箱上的资源进行控制和交互。

（1）STC 单片机能正确接收到红外遥控器的编码信息，并显示（不限制显示介质，串口或 1602）（50 分）

（2）能控制 LED（20 分）

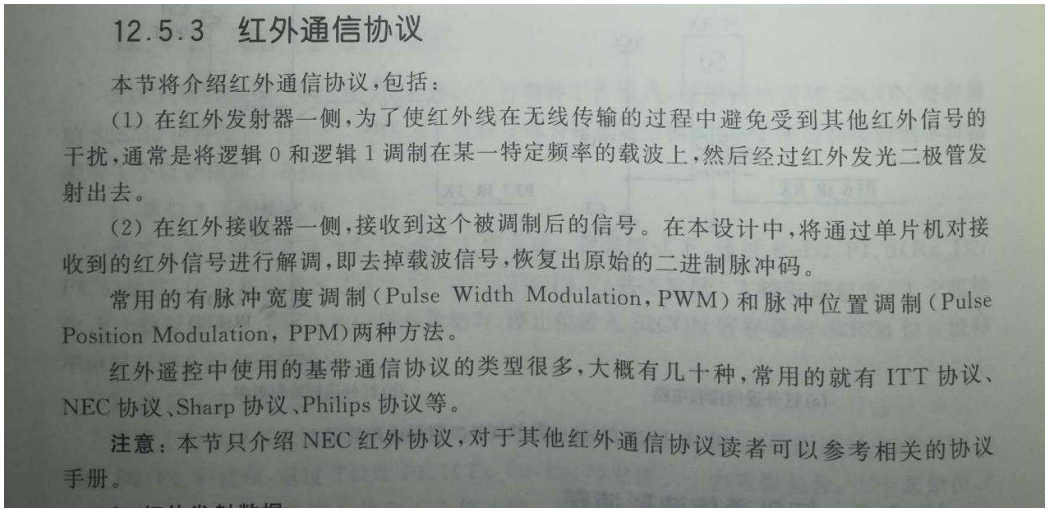
（3）能实现更复杂的显示交互和控制功能（30 分）

设计思路

本设计题的目的是使我们合理使用单片机内的红外收发器的电路原理，实红外接收的设计，加强对所学知识的理解并合理运用。可参考课本 P350 页例子。

我首先考虑了 STC 单片机能正确接收到红外遥控器的编码信息并且显示，用到了串口，那么就必须先将串口所需的寄存器，各种函数设置准备好。

根据红外接收协议，应该声明检测红外发送数据的高，低电平持续时间函数。设置外部中断 2 的服务程序，另外，使用了定时器 0 和定时器 2。



12.5.3 红外通信协议

本节将介绍红外通信协议，包括：

(1) 在红外发射器一侧，为了使红外线在无线传输的过程中避免受到其他红外信号的干扰，通常是将逻辑 0 和逻辑 1 调制在某一特定频率的载波上，然后经过红外发光二极管发射出去。

(2) 在红外接收器一侧，接收到这个被调制后的信号。在本设计中，将通过单片机对接收到的红外信号进行解调，即去掉载波信号，恢复出原始的二进制脉冲码。

常用的有脉冲宽度调制（Pulse Width Modulation，PWM）和脉冲位置调制（Pulse Position Modulation，PPM）两种方法。

红外遥控中使用的基带通信协议的类型很多，大概有几十种，常用的就有 ITT 协议、NEC 协议、Sharp 协议、Philips 协议等。

注意：本节只介绍 NEC 红外协议，对于其他红外通信协议读者可以参考相关的协议手册。

根据红外发射原理，包括：

载波波形，信号调制在 37.91HZ，占空比为 1/3。

数据格式，16 位用户吗，8 位数据码，8 位数据反码

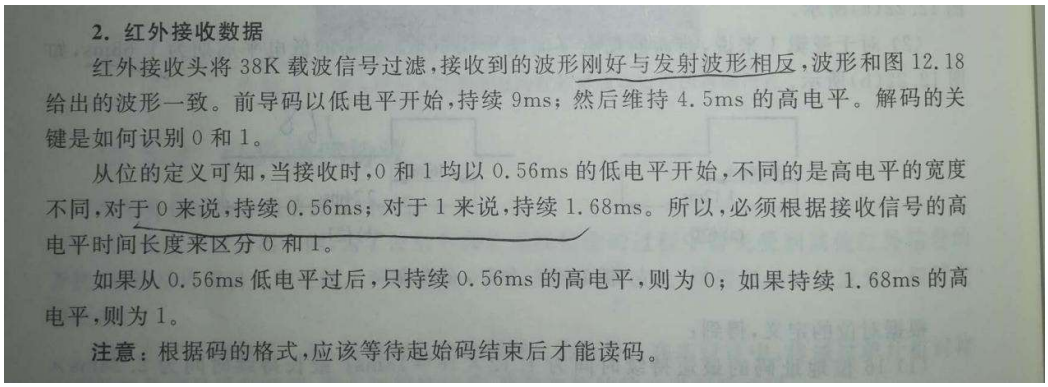
位定义，逻辑 0，高电平周期 0.56ms,低电平周期 0.56ms,

逻辑 1，高电平周期 0.56ms,低电平周期 1.68ms,

按键输出波形

重复码，单一按键波形，连续按键波形。

红外接收原理，



2. 红外接收数据

红外接收头将 38K 载波信号过滤，接收到的波形刚好与发射波形相反，波形和图 12.18 给出的波形一致。前导码以低电平开始，持续 9ms；然后维持 4.5ms 的高电平。解码的关键是如何识别 0 和 1。

从位的定义可知，当接收时，0 和 1 均以 0.56ms 的低电平开始，不同的是高电平的宽度不同，对于 0 来说，持续 0.56ms；对于 1 来说，持续 1.68ms。所以，必须根据接收信号的高电平时间长度来区分 0 和 1。

如果从 0.56ms 低电平过后，只持续 0.56ms 的高电平，则为 0；如果持续 1.68ms 的高电平，则为 1。

注意：根据码的格式，应该等待起始码结束后才能读码。

在设计 LED 灯时，首先 P46 灭，然后遥控按键 1 时，灯亮，遥控按键 2 时灯灭。

根据之前电子时钟和 1602 的知识，设计电子时钟，使用了定时器 1，将外部中断控制改为红外遥控器按键控制。

遇到的问题及解决方法

1. 设计串口显示时，由于所用寄存器较多，出现设置失误，经过排查后，找到错误，正确显示；

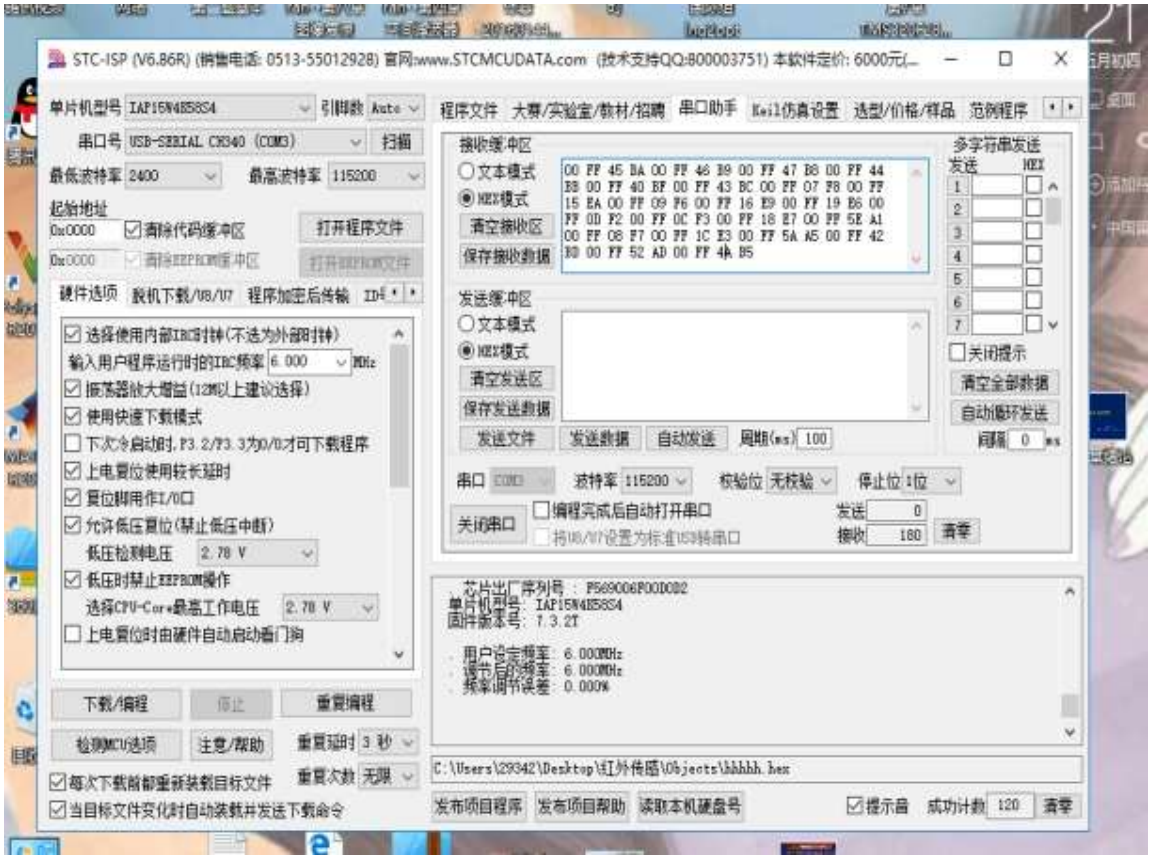
2. 设计电子时钟时，最开始采用了分频计数，由于频率问题，计时太快，后来改用手动计数，调整好了计时方式与频率，并正确计数。

实验结果

1. 串口显示：

例如：

遥控按键 1，显示 00 FF 0C F3, 按键 2，显示 00 FF 18 E7，按键 3，显示 00 FF 5E A1 等等。



2. LED 灯显示：

在设计 LED 灯时，首先 LED10 灭，然后遥控按键 1 时，灯亮，遥控按键 2 时灯灭。

led灯.mp4

(视频文件也有单独的文件)

该部分主要代码：

```
if(flag==1)

        {

            flag=0;

            for(k=0;k<4;k++)

            {

                SendData(irdata[k]);
```

```
        if(irdata[2]==0x0C)//LED 灯控制
        {
          P46=0;
        }
        if(irdata[2]==0x18)
          {
           P46=1;
          }
    }
```

3.能实现更复杂的显示交互和控制功能：

这里我设计了电子时钟，进行加一计数，

红外遥控器按键“3”，控制时钟的“时”，按下一次“时”加一；

红外遥控器按键“4”，控制时钟的“分”，按下一次“分”加一；

红外遥控器按键“5”，控制时钟的“秒”，按下一次“秒”加一；

(视频文件也有单独的文件)

电子时钟.mp4

代码：

main.c:

```
#include "reg51.h"

#include "stdio.h"

#include "intrins.h"

#include "led1602.h"
```

```
#define FOSC 6000000L//声明单片机主时钟频率 6MHZ

#define BAUD 115200//声明串口通信的波特率 115200

#define TIMS 3036

sfr AUXR       =0x8E;//声明 AUXR 寄存器的地址 0x8E

sfr AUXR1      =0xA2;//声明 AUXR 寄存器的地址 0xA2

sfr AUXR2          =0x8F;//声明 AUXR 寄存器的地址 0x8F

sfr TH2            =0xD6;//声明 TH2 寄存器的地址 0xD6

sfr TL2            =0xD7;//声明 TL2 寄存器的地址 0xD7

//sfr S2CON        =0x9A;//声明 S2CON 寄存器的地址 0x9A

//sfr S2BUF        =0x9B;//声明 S2BUF 寄存器的地址 0x9B

sfr P3M1           =0xB1;//声明 P3M1 寄存器的地址 0xB1sfr P3M0
    =0xB2;//声明 P3M0 寄存器的地址 0xB2

sfr P3M0=0xB2;

sbit P36  =P3^6;

sfr P4     =0xC0;

sbit P46  =P4^6;

int postd[2][3] = {{5,8,11},{5,10,13}};

int n=0,num=0;

int hour0=0,minu0=0,sec0=0;//时分秒初始化

bit busy=0;

unsigned char irdata[4]={0,0,0,0};
```

```
bit flag=0;
void SendData(unsigned char dat)//声明 SendData 子函数
{
    while(busy);
    SBUF=dat;
    busy=1;
}
void SendString(char *s)//声明 SendString 子函数
{
    while(*s!='\0')
        SendData(*s++);
}
unsigned int high_level_time()//检测红外发送数据的高电平持续时间函数
{
    TL0=0;
    TH0=0;
    TR0=1;
    while(P36==1)
    {
        if(TH0>=0xEE)
            break;
```

```
    }
    TR0=0;
    return(TH0*256+TL0);
}
unsigned int low_level_time()//检测红外发送数据的低电平持续时间函数
{
    TL0=0;
    TH0=0;
    TR0=1;
    while(P36==0)
    {
        if(TH0>=0xEE)
            break;
    }
    TR0=0;
    return(TH0*256+TL0);
}
void int2() interrupt 10//声明外部中断 2 的服务程序
{
    unsigned char i,j;
    unsigned int count=0;
```

```
unsigned char dat=0;


AUXR2&=0x00;
count=low_level_time();
    if(count<4000 || count>5000)
{
    AUXR2|=0x10;
   return;
}
count=high_level_time();
if(count<2000 || count>2500)
{AUXR2|=0x10;
  return;
}
for(i=0;i<4;i++)
{
P36=1;
    dat=0;
   for(j=0;j<8;j++)
      {
         count=low_level_time();
```

```
            if(count<150 || count>400)
            {AUXR2|=0x10;
          return;}
            count=high_level_time();
            if(count>150 && count<400)
                dat>>=1;
            else if(count>700 && count<1100)
            {
                dat>>=1;
                dat|=0x80;
            }
            else
                {
                AUXR2|=0x10;
                return;}
          }
          irdata[i]=dat;
    }
     flag=1;
    AUXR2|=0x10;
}
```

```
void uart1() interrupt 4//声明uart串口1中断服务程序
{
    if(RI)
        RI=0;
    if(TI)
        TI=0;
    busy=0;
}
    void add_time() interrupt 3
{
    TH1=(65535-50000)/256;
    TL1=(65535-50000)%256;
    num++;
    if(num==20)
        {
        num=0;
        sec0++;
      if(sec0 >= 60)
            {
            sec0 = 0;minu0++;
         }
```

```
        if(minu0 >= 60)
              {
              minu0 = 0;hour0++;
            }
            if(hour0 >= 24)
              {
              hour0 = 0;
            }
     }
}
void print_time()//时分秒显示
    {
    char h[3],m[3],s[3];
    sprintf(h,"%02d",hour0);
     sprintf(m,"%02d",minu0);
    sprintf(s,"%02d",sec0);
    lcdshowstr(0,0,"TIME:");
    if(sec0%2){
        lcdshowstr(10,0,":");
        lcdshowstr(7,0,":");
    }
```

```
    lcdshowstr(5,0,h);
    lcdshowstr(8,0,m);
    lcdshowstr(11,0,s);
}
void main()//主程序
{
    unsigned char k;//定义无符号的字符变量 k
  P46=1;//灭

    P36=1;//设置 P3.6 引脚为高
    P3M1=0x00;//将 P3 端口设置为准双向弱上拉
    P3M0=0x00;//通过 P3M1 和 P3M0 寄存器，设置 P3 端口模式
    TMOD=0x00;//设置定时器 0 的工作模式
    SCON=0x50;
    AUXR=0x14;//定时器 2/12，启动定时器 2，定时器模式
    AUXR|=0x01;
    TL2=(65536-((FOSC/4)/BAUD));//计数初值低 8 位给定时器 2 的 TL2 寄存器
    TH2=(65536-((FOSC/4)/BAUD))>>8;//计数初值高 8 位给定时器 2 的 TH2 寄存器
TH1=(65535-50000)/256;
    TL1=(65535-50000)%256;
```

```
AUXR2|=0x10;//允许外部中断 2
ES=1;
EA=1;//CPU 允许响应中断请求
TMOD = 0x80;
TR1 = 1;//启动定时器
ET1 = 1;//使能定时器 1 中断
P0M1 = 0;//通过 P0M0 和 P0M1 寄存器将 P0 口定义为准双向
P0M0 = 0;
P2M1 = 0;//通过 P0M0 和 P0M1 寄存器将 P2 口定义为准双向
P2M0 = 0;
lcdwait();
lcdinit();
 while(1)
    {
       print_time();
     if(flag==1)
     {
         flag=0;
         for(k=0;k<4;k++)
         {
             SendData(irdata[k]);
```

```
        if(irdata[2]==0x0C)//LED 灯控制
        {
          P46=0;
        }
        if(irdata[2]==0x18)
          {
           P46=1;
         }
    }
if(irdata[2]==0x5E)//按键控制电子时钟
     {
         TR1=0;
         hour0++;
        if(hour0==24)
        {
            hour0=0;
        }
          TR1=1;
            print_time();
     }
          if(irdata[2]==0x08)
```

```
            {
                    TR1=0;
             minu0++;
               if(minu0==60)
               {
                  minu0=0;
                }
                TR1=1;
               print_time();
            }
                if(irdata[2]==0x1C)
               { TR1=0;
                  sec0++;
                  if(sec0==60)
                     sec0=0;
                        TR1=1;
                          print_time();
                }
        //print_time();
         AUXR2|=0x10;
}
```

```
    }

}
```