

一、设计要求：

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）
- (2) 能控制 LED
- (3) 能实现更复杂的显示交互和控制功能

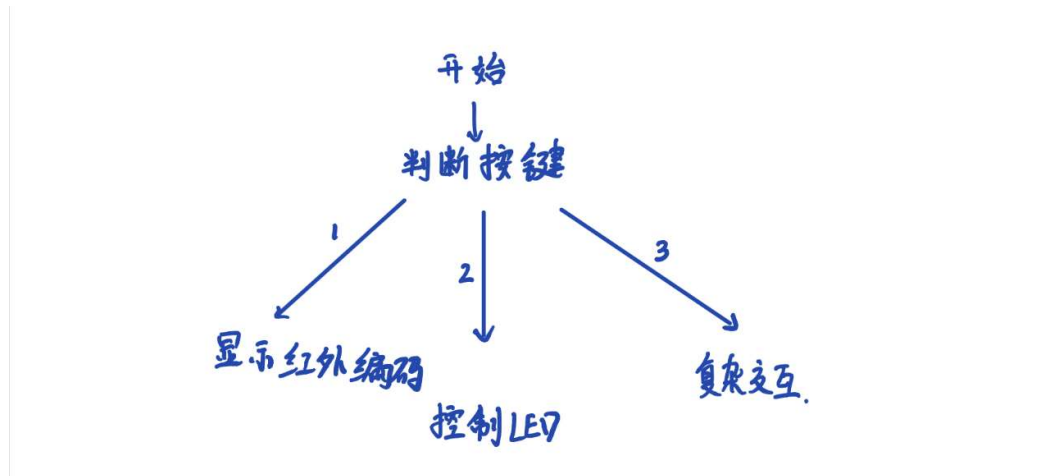
二、设计目标

这道题应该是考察我们对于红外通信的了解程度。能否可以自己写通信的程序。知道红外的编码规则是什么，知道如何实现这个通信。

三、设计方法

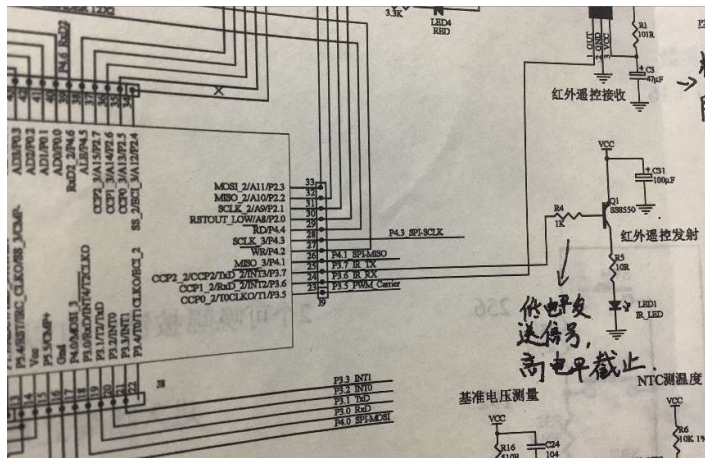
1、程序流程图

首先我先画了一个简单的功能框图图，如下图所示。



可以主要分成以下的几个方面。读入红外信号，转换成编码，lcd 显示，控制 led，附加的交互功能。

2、红外信号的读入



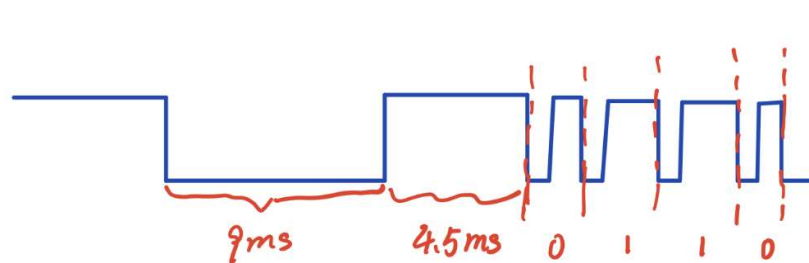
我们在图中我们就可以看到红外信号的输入的引脚是 P3. 6. 正好所对应的中断是外部中断 2. 我们可以把这个引脚设为双向的。就可以实现信号的读入了。

3、转化为编码

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。

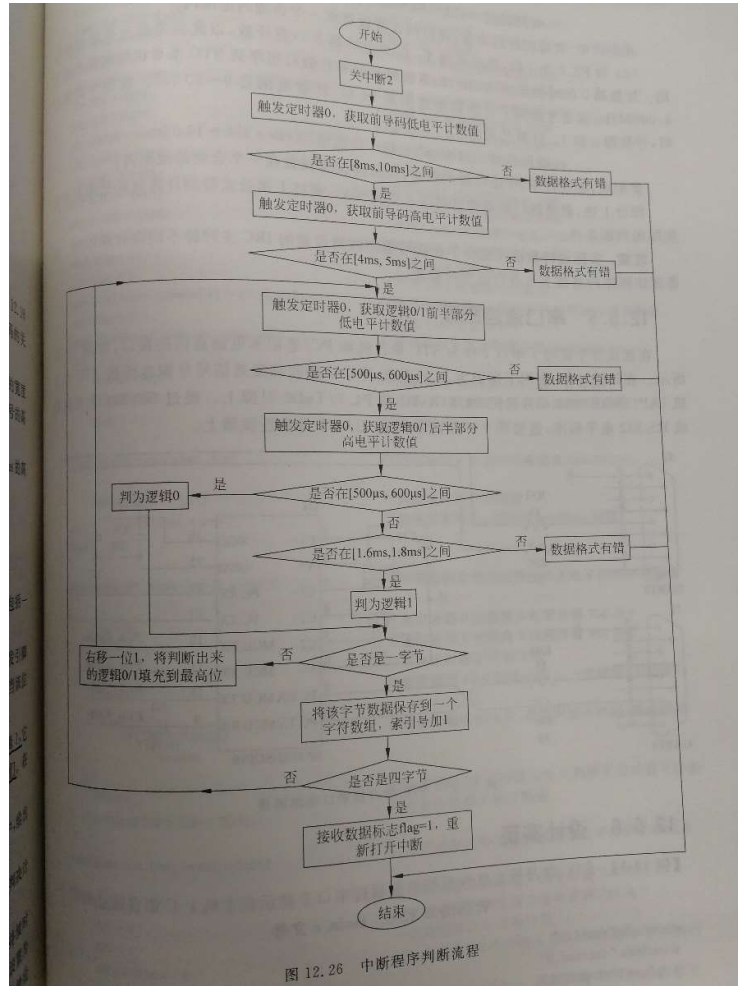
从位的定义可知,当接收时,0 和 1 均以 0.56ms 的低电平开始,不同的是高电平的宽度不同,对于 0 来说,持续 0.56ms; 对于 1 来说,持续 1.68ms。所以,必须根据接收信号的高电平时间长度来区分 0 和 1。

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。



红外传输一次, 传送了 4 个字节。16 位的地址码和 8 位数据码和 8 位数据反码。我们需要的就是 8 位的数据码。我们知道, 我们这个遥控器的前导码也就是持续 9ms 的低电平和持续 4.5ms 的高电平。当我们单片机收到这个信号的时候就说明信号传输开始了。我们接下来就只需要判断 0 和 1 即可。发送 0 的时候, 是 0.56ms 的低电平之后又持续 0.56ms 的高电平。发送 1 的时候, 是 0.56ms 的低电平之后有持续 1.68ms 的高电平。我们就可以通过定时器就可以判断高低电平的时间, 从而判断 0 或者 1。

程序流程图如图所示



对于定时器来说，我所选的系统时钟是 6MHZ，因此定时器时钟为 6MHZ/12. 因此我们经过 t 的时间定时器就加一。

$$t = \frac{1}{6\text{MHz}/12}$$

由此我们就可以通过判断定时器记了多少次数来判断经过了多长的时间。

$$\text{SYSCLK} = 6\text{MHz}$$

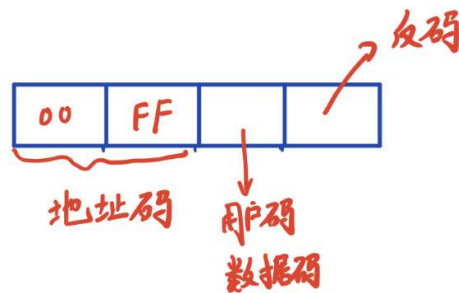
$$\text{定时器时钟} \quad \frac{\text{SYSCLK}}{12} = \frac{6\text{MHz}}{12}$$

每计数 Count 次.

$$t = \frac{1}{\frac{6\text{MHz}}{12}} \times \text{Count}$$

4、lcd 显示与控制 led

我们接收到的字节数据保存在 irdata 里面，然后我们所要的用户码也就是第三个。即 irdata[2]。我们如果需要判断遥控器的哪一个键被按下只需要把此数据码与遥控器的用户码进行对比即可。如果要是能在 lcd 显示出来，我们就只需要把它转化为 char 类型的就可以了。采用 sprintf 函数。



5、附加的交互功能

我打算做的是一个小游戏，也就是用 lcd 显示一个数字，然后在一定时间内单片机要接

收到这个数字即可，如果没有接收到这个数字，表示游戏失败。

他的主要的流程就是先随机产生一个 0 到 9 的数字，然后启动定时器并且开始计数。同时开启红外接受的中断。如果在规定时间内没有接收到信号，游戏失败，退出循环。

四、设计代码

```
#include "reg51.h"
#include "stdio.h"
#include "stdlib.h"
#include "intrins.h"
#include "led1602.h"

sbit P36  =P3^6;
sbit P46  =P4^6;
sbit P47  =P4^7;

sfr AUXR  = 0x8e;
sfr AUXR2 =0x8f;
sfr CLK_DIV =0x97;

sfr P3M1  =0xB1;
sfr P3M0  =0xB2;

sfr AUXR1 =0xA2;
sfr TH2    =0xD6;
sfr TL2    =0xD7;

sfr P_SW2 =0xBA;
sfr IE2    =0xAF;

unsigned int count,time;
unsigned int irdata[4]={0,0,0,0};
unsigned char dl[20];
bit flag=0;      //标志位，0：红外正在接收数据      1：数据接收完毕
bit flag1=0;

unsigned int id,id1,id2;
/*****随机编码*****/
unsigned int suiji()
{
    unsigned int a, i;
    //使用 for 循环生成 10 个随机数
    for (i = 0; i < 10; i++)
    {
        srand(time);
        a = rand()%10;
    }
}
```

```

    }

    switch(a)
    {
        case 0:{strshow(0,1,"0");id2=22;}break;
        case 1:{strshow(0,1,"1");id2=12;}break;
        case 2:{strshow(0,1,"2");id2=24;}break;
        case 3:{strshow(0,1,"3");id2=94;}break;
        case 4:{strshow(0,1,"4");id2=8;}break;
        case 5:{strshow(0,1,"5");id2=28;}break;
        case 6:{strshow(0,1,"6");id2=90;}break;
        case 7:{strshow(0,1,"7");id2=66;}break;
        case 8:{strshow(0,1,"8");id2=82;}break;
        case 9:{strshow(0,1,"9");id2=74;}break;
    }
    return id2;
}

/*****低电平持续时间*****/
unsigned int high_level_time()
{
    TL0=0;           //初值为0
    TH0=0;
    TR0=1;           //启动定时器开始计数
    while(P36==1)
    {
        if(TH0>=0xEE) //时间过长的话退出循环
            break;
    }
    TR0=0;           //停止计数
    return(TH0*256+TL0); //返回计数值
}

/*****低电平持续时间*****/
unsigned int low_level_time()
{
    TL0=0;           //初值为0
    TH0=0;
    TR0=1;           //启动定时器
    while(P36==0)
    {
        if(TH0>=0xEE)
            break;
    }
    TR0=0;

```

```

    return (TH0*256+TL0);
}

/*****外部中断 2 的中断服务函数，即红外的中断服务函数*****/
void int2() interrupt 10
{
    unsigned char i,j;
    unsigned int count=0;
    unsigned char dat=0;

    AUXR2&=0x00;          //关中断
    count=low_level_time(); //计算低电平时间

    if(count<4000 || count>5000)
    {
        AUXR2|=0x10;
        return;
    }
    count=high_level_time(); //计算高电平时间
    if(count<2000 || count>2500)
    {
        AUXR2|=0x10;
        return;
    }
    /*****接受红外编码信号*****/
    for(i=0;i<4;i++)      //四个字节
    {
        P36=1;
        dat=0;
        for(j=0;j<8;j++)    //8 位数据
        {
            count=low_level_time();
            if(count<200 || count>350)
            {
                AUXR2|=0x10;
                return;
            }
            count=high_level_time();
            if(count>200 && count<350)
                dat>>=1;      //右移，用 0 填充
            else if(count>700 && count<1100)
            {
                dat>>=1;      //右移，用 1 填充
                dat|=0x80;
            }
        }
    }
}

```

```

        }
        else
        {
            AUXR2|=0x10;
            return;
        }

    }

    irdata[i]=dat;
}

flag=1; //接收数据的标志位置 1
AUXR2|=0x10; //开中断
}

void timer1() interrupt 3
{
    count++;
}

void main()
{
    P36=1;
    P3M1=0x00; //P3 端口引脚模式
    P3M0=0x00;

    TMOD=0x00; //定时器 0 和 1 的配置
    AUXR=0x01; //打开定时器 0, 1.均为为系统时钟的 12 分频
    AUXR2|=0x10;
    EA=1;
    initlcd();
    while(1)
    {
        strshow(0,0,"1:CODE    2:LED");
        strshow(0,1,"3:GAME");
        if(flag==1)
        {
            flag=0;
            id=irdata[2];
            AUXR2|=0x10;
        }
        if(id==12) //如果为 1, 则进入查看编码模块
        {
            id=0;

```



```

writecmd(0x01);          //清屏指令
while(1)
{
    if(flag==1)
    {
        flag=0;
        sprintf(d1,"%02X %02X %02X %02X
",irdata[0],irdata[1],irdata[2],irdata[3]);
        strshow(3,0,d1);
        strshow(2,1,"CODE");
        strshow(10,1,"EXIT:9");
        AUXR2|=0x10;
    }

    if(irdata[2]==74)      //如果按下 9, 退出
    {
        writecmd(0x01);    //清屏指令
        break;
    }
}

else if(id==24)          //为 2, led 灯变化
{
    id=0;
    writecmd(0x01);        //清屏指令
    while(1)
    {
        if (flag==1)
        {
            flag=0;
            switch (irdata[2])
            {
                case 28:P46=!P46;break;
                case 90:P47=!P47;break;
            }
            AUXR2|=0x10;
        }
        strshow(2,0,"LED control");
        strshow(10,1,"EXIT:9");
        if(irdata[2]==74)    //如果按下 9, 退出
        {
            writecmd(0x01);    //清屏指令
            break;
        }
    }
}

```

```

    }
}

else if(id==94)    //为 3， 游戏
{
    id=0;
    writecmd(0x01);    //清屏指令
    strshow(6,0,"GAME");
    strshow(0,1,"begin:CH  EXIT:9");
    while(1)
    {
        if (flag==1)
        {
            flag=0;
            if(irdata[2]==70)
            {
                writecmd(0x01);    //清屏指令
                strshow(6,0,"BEGIN");
                TL1=0;    //给定时器 1 设置初值
                TH1=0;
                ET1=1;    //开启定时器 1 中断
                TR1=1;    //开始定时器 1
                EA=1;    //使能全局中断
                id1=suiji();    //把随机产生的数赋值给 id1
                count=0;    //count 归 0
                while(count<=20)
                {
                    time++;
                    if(time==500)
                        time=0;
                    if(flag==1)
                    {
                        flag=0;
                        if(irdata[2]==id1)    //如果接收到正确的数据
                        {
                            count=0;    //count 重新归 0
                            id1=suiji();    //重新给 id1 赋值
                        }
                    }
                }
                irdata[2]=0;
                strshow(4,0,"GAME OVER");    //如果规定时间没有收到正确数据

                strshow(10,1,"EXIT:9");    //游戏结束
                TR1=0;    //关闭定时器 1
            }
        }
    }
}

```

