

# 红外通信和遥控 的设计

---

讲解人：叶谋佳  
班级：信工1602班

# 目 录

## CONTENT

---

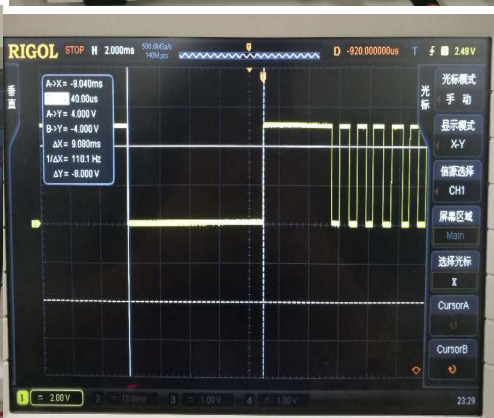
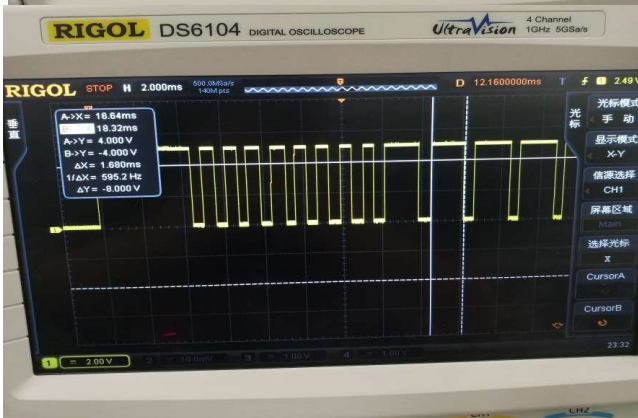
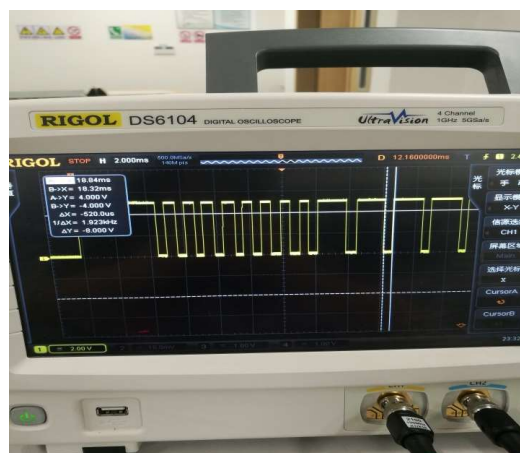
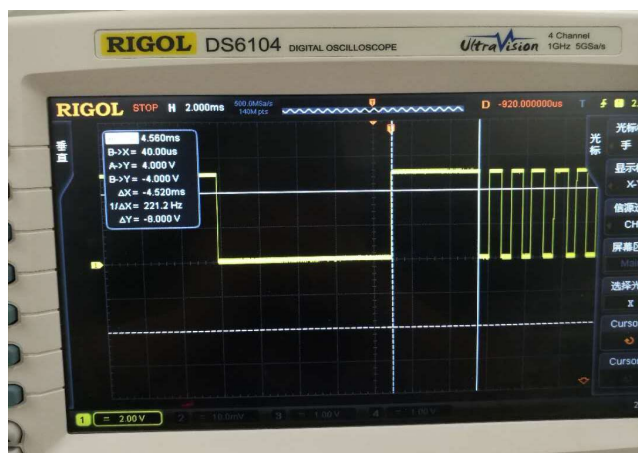
- ◆ 原理讲解
- ◆ 设计思路
- ◆ 代码实现
- ◆ 遇到的问题与解决过程



# PART 1

# PART 1

单片机接收到的波形



起始低电平的时间为9ms左右，起始高电平的时间为4.5ms左右，之后的高低电平时间分别为1.68ms和520us。

# PART 1

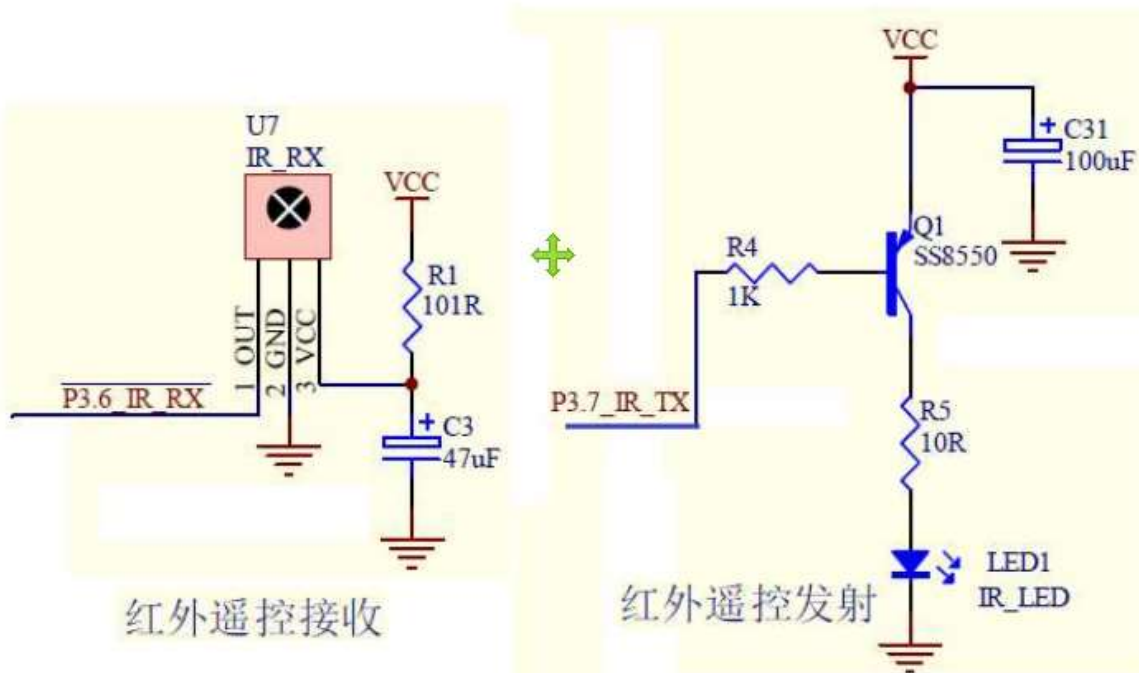
## NEC协议传输



为NEC协议传输格式，起始位（引导码）为9ms高+4.5ms低组成，有效数据为地址+地址反码+命令+命令反码。



# PART 1



用于接收红外线通信信息的红外接收器连接到单片机的P3.6引脚处。该接收器将红外光携带的信息，转换成电信号，通过P3.6引脚传给单片机进行处理。



PART

2

# PART 2



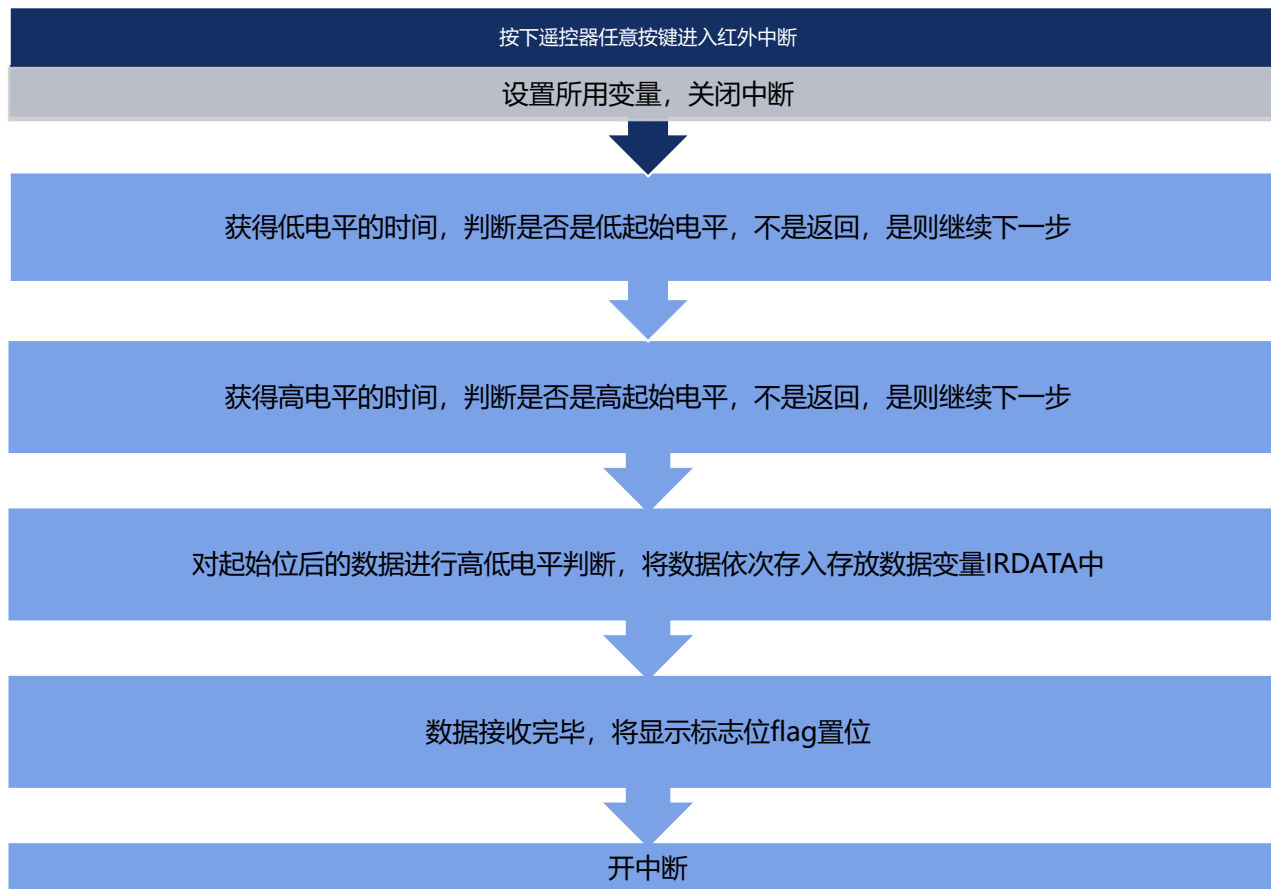
## 红外中断服务程序

当P3.6为0时, 表示低电平, 启动定时器0一直计数, 以此获得低电平的持续时间。  
当P3.6为1时, 表示高电平, 启动定时器0一直计数, 以此获得高电平的持续时间。



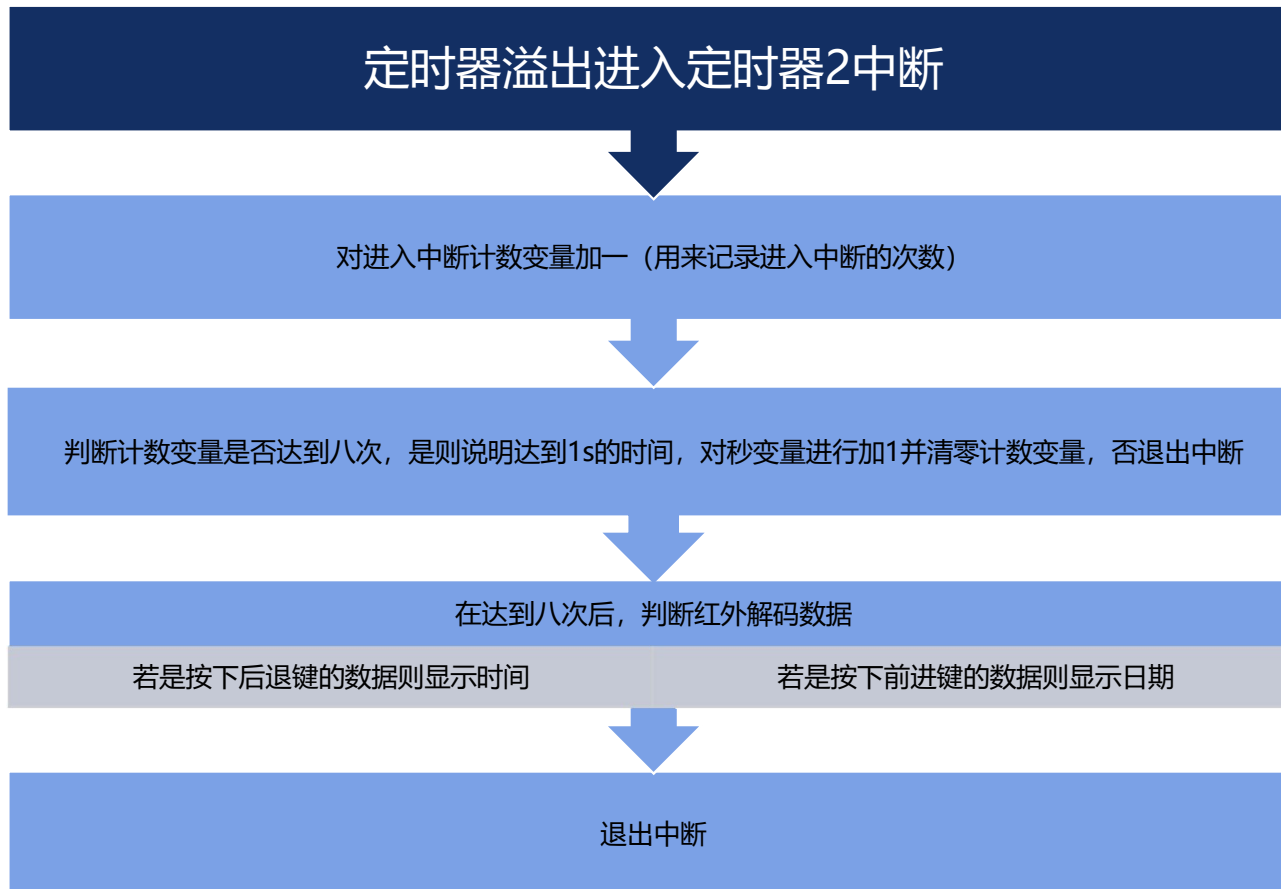
# PART 2

## 红外中断服务程序



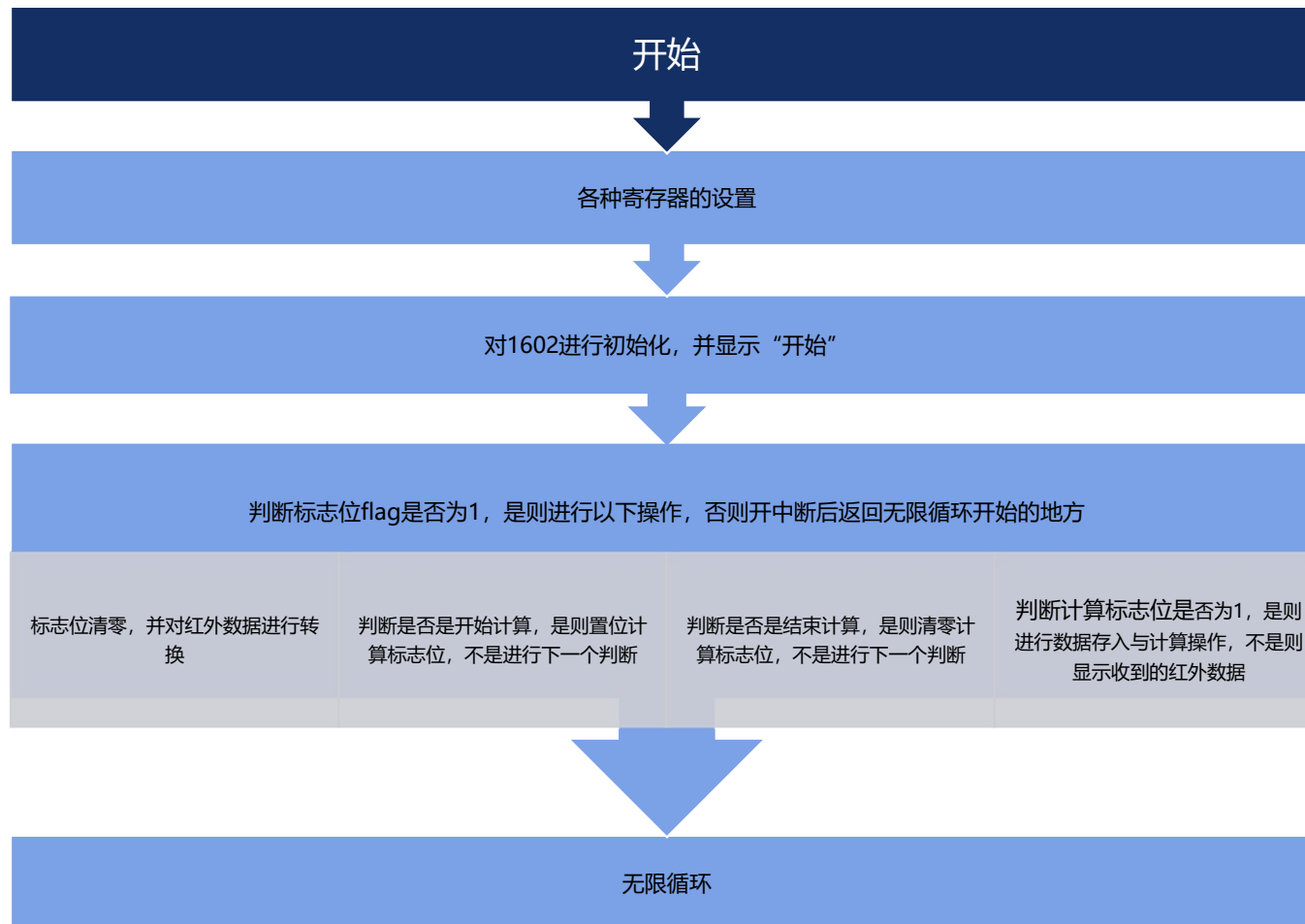
# PART 2

## 定时器 2服务 程序



# PART 2

## 主函数 程序





# PART 3

# PART 3

```
191 //红外中断
192 void int2() interrupt 10
193 {
194     unsigned char i,j;
195     unsigned int count=0;
196     unsigned char dat=0;
197
198     n++;
199     AUXR2&=0x00;
200     count=low_level_time();
201     // lcdshowstr(6,1,"DA");
202     if(count<4000 || count>5000)
203     {
204         return;
205     }
206     // lcdshowstr(6,1,"DATE");
207     count=high_level_time();
208     if(count<2000 || count>2500)
209     {
210         return;
211     }
212
213     for(i=0;i<4;i++)
214     {
215         P36=1;
216         dat=0;
217         for(j=0;j<8;j++)
218         {
219             count=low_level_time();
220             if(count<200 || count>350) //经过试验得到该值下可以接收高低电平
221                 return;
222             count=high_level_time();
223             if(count>200 && count<350)
224                 dat>>=1;
```

```
225         else if(count>700 && count<1000)
226         {
227             dat>>=1;
228             dat|=0x80;
229         }
230         else return;
231     }
232     irdata[i]=dat;
233 }
234     flag=1;
235     AUXR2|=0x10;
236 }
```

# PART 3

## 主函数

```
378 while(1)
379 {
380     if(flag==1)
381     {
382         flag=0;
383         //将接收到的红外数据变成对应数字和字符
384         irdatatoint();
385         switch (irdata[2])
386         { case 0x47: //进入计算模式
387             { sumf=1;
388                 lcdwritecmd(0x01); //清屏
389                 lcdshowstr(0,0,"CALCULATOR");
390                 break;
391             }
392             case 0x45: //退出计算模式
393             { sumf=0;
394
395                 lcdwritecmd(0x01); //清屏
396                 lcdshowstr(0,0,"EXIT");
397                 break;
398             }
399             default:;
400         }
401
402         if(sumf) //如果是计算模式
403         {
404             if(fqing) lcdwritecmd(0x01); //清屏
405             fqing=0;
406             switch (disda)
407             { case 11: //输入加号
408                 { datam=datan;
409                     datan=0;
410                     flags=1;
411                     lcdshowstr(4,1,discha);
```

```
411         lcdshowstr(4,1,discha);
412         flagc=1;
413         break;
414     }
415     case 12: //输入减号
416     { datam=datan;
417         datan=0;
418         flags=1;
419         lcdshowstr(4,1,discha);
420         flagc=0;
421         break;
422     }
423     case 13: //计算结果
424     {
425         if(flagc) sumdata=datan+datam;
426         else sumdata=datam-datan;
427         sprintf(sumcha,"%d",sumdata);
428         //lcdwritecmd(0x01); //清屏
429         lcdshowstr(9,1,"=");
430         lcdshowstr(10,1,sumcha);
431         datan=0;
432         sumdata=0;
433         flags=0;
434         fqing=1;
435         break;
436     }
437     case 14: //开始计算
438     {
439         lcdwritecmd(0x01); //清屏
440         lcdshowstr(0,0,"CALCULATOR");
441         flagc=0;
442         datan=0;
443         sumdata=0;
444         break;
```



# PART 3

## 主函数

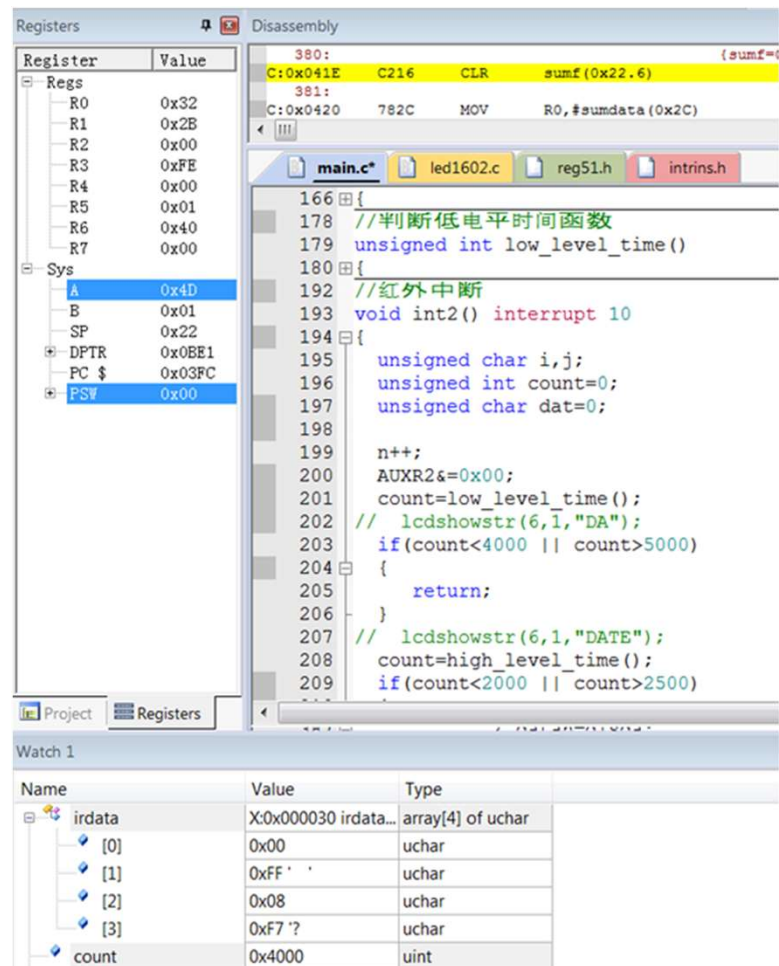
```
441         flagc=0;
442         datan=0;
443         sumdata=0;
444         break;
445     }
446     default:
447     {
448         if(!flags)
449         {datan=datan*10+disda;
450          sprintf(chadataan,"%d",datan);
451          lcdshowstr(0,1,chadataan); }
452         else
453         { datan=datan*10+disda;
454          sprintf(chadataan,"%d",datan);
455          lcdshowstr(5,1,chadataan);
456          }
457     }
458 }
459 }
460 else
461 {
462     lcdwait();
463     lcdwritecmd(0x01);//清屏
464     display();
465     if(irdata[2]==0x45)
466     {
467         lcdwritecmd(0x01);//清屏
468         lcdshowstr(0,0,"EXIT CALCULATOR");
469     }
470 }
471 }
472 AUXR2|=0x10;
473 }
474 }
```

The background features a large, abstract geometric pattern composed of overlapping triangles and polygons. The colors transition from a deep teal on the left to a bright yellow in the center, and then to a warm orange and red on the right. The pattern is layered, creating a sense of depth. In the bottom right corner, the text "PART 4" is displayed in a large, bold, green font. The word "PART" is in a lighter shade of green, while the number "4" is in a darker shade.

# PART 4

# PART 4

1.在进行红外显示的时候，我的红外数据一直不能显示，我思考之后发现可能有以下几个问题：（1）没有进入红外中断。（2）进入红外中断但是没有进行高低电平判断。（3）irdata有值但是不能显示。我通过在红外中断的不同位置调用1602显示函数，来观察是什么原因，经过判断，我发现是在判断完起始高低电平后的一个判断中出现了问题，是if判断的范围太窄了，我进入硬件调试查看irdata和count的值后给出了合适的判断区间，然后就能正确判断了。



Registers

Register	Value
R0	0x32
R1	0x2B
R2	0x00
R3	0xFE
R4	0x00
R5	0x01
R6	0x40
R7	0x00
Sys	
A	0x4D
B	0x01
SP	0x22
DPTR	0x0BE1
PC	0x03FC
PSW	0x00

Disassembly

```
380: 380: C216 CLR sumf(0x22.6)
381: 782C MOV R0,#sumdata(0x2C)

main.c led1602.c reg51.h intrins.h

166 {
178 //判断低电平时间函数
179 unsigned int low_level_time()
180 {
192 //红外中断
193 void int2() interrupt 10
194 {
195     unsigned char i,j;
196     unsigned int count=0;
197     unsigned char dat=0;
198
199     n++;
200     AUXR2&=0x00;
201     count=low_level_time();
202     // lcdshowstr(6,1,"DA");
203     if(count<4000 || count>5000)
204     {
205         return;
206     }
207     // lcdshowstr(6,1,"DATE");
208     count=high_level_time();
209     if(count<2000 || count>2500)
```

Watch 1

Name	Value	Type
irdata	X:0x000030 irdata...	array[4] of uchar
[0]	0x00	uchar
[1]	0xFF	uchar
[2]	0x08	uchar
[3]	0xF7	uchar
count	0x4000	uint

# PART 4

2. 得到正确的红外接收数据后还是不能正确的显示，我就对显示函数进行单独的调试，我发现在修改了数据类型之后就能正常显示了。因为没有负数的概念，我先将无符号字符型转换为整型，再转换为有符号字符型进行显示。

3. 在我将电子表与红外控制结合在一起时我发现两个定时器不能正常工作，我通过对代码的逐一筛查发现是我原来用的分16倍频后的系统时钟影响了定时器0的计数工作，导致计数值不正确，从而不能正确的判断接收到的红外数据。对于这个问题，我想了几个解决方案：（1）不改变定时器2的工作时钟，对判断计数值的区间进行重新设置，但这个方法付出代价太高，计数值的判断区间很多，如果逐一修改将要花费很多时间。（2）改变计数时钟，不改变定时器0的工作。经过多次调试，我发现在不改变初值的状态下，每进入8次溢出中断后对变量秒进行加1，得到的计数时间就是正确的。

4. 在进行计算模式的时候由于不能和显示模式很好的兼容，导致了乱码。我重新梳理了思路，运用标志位来判断，并加入清屏函数，最后得到了比较满意的效果。

**THANK YOU**

A horizontal banner spanning the width of the image. The left portion is a solid teal color, and the right portion features a complex geometric pattern of overlapping triangles in various shades of yellow, orange, and light green.