西北工業大學

单片机与嵌入式系统
课程设计

题目 基于单片机红外接收器和光敏电阻的 LED 点阵动画设计

分　数 _____

一、 实验内容

本实验选用了 AT89C51 单片机当红外线遥控器的核心，主要选用了单片机中断系统、定时器、计数器等单元，利用红外光的特点。本文先介绍了红外遥控的基本原理和应用范围，再对 AT89C2051 单片机的构造和性能给出简单的说明，接着给出了遥控器的编码方式，及遥控发射器，遥控接受器的电路设计。根据遥控操作的差别，遥控发射器经过对红外光发射频率的控制来区别不相同的操作；遥控接收器通过对红外光接收频率的识别，判断出控制操作，来完成整个红外遥控发射、接收过程。最后分别具体介绍遥控系统的发射部分和接收部分的电路原理图和程序流程图。

二、 实验环境

单片机介绍：

STC89C51RC/RD+ 系列单片机是宏晶科技推出的新一代超强抗干扰/ 高速/ 低功耗的单片机，指令代码完全兼容传统 8051 单片机，12 时钟/ 机器周期和 6 时钟/ 机器周期可任意选择。

特点：

1. 增强型 6 时钟/ 机器周期，12 时钟/ 机器周期 8051 CPU

2. 工作电压：5.5V - 3.4V（5V 单片机） / 3.8V - 2.0V（3V 单片机）

3. 工作频率范围：0 - 40 MHz，相当于普通 8051 的 0～80MHz.实际工作频率可达 48MHz.

4. 用户应用程序空间 4K / 8K / 13K / 16K / 20K / 32K / 64K 字节

5. 片上集成 1280 字节 / 512 字节 RAM

6. 通用 I/O 口（32/36 个），复位后为： P1/P2/P3/P4 是准双向口/ 弱上拉（普通 8051 传统 I/O 口）

P0 口是开漏输出，作为总线扩展用时，不用加上拉电阻，作为 I/O 口用时，需加上拉电阻。

7 . I S P （在系统可编程）/ I A P （在应用可编程），无需专用编程器/ 仿真器

可通过串口（P3.0/P3.1）直接下载用户程序，8K 程序 3 秒即可完成一片

8. EEPROM 功能

9. 看门狗

10.内部集成 MAX810 专用复位电路（D 版本才有），外部晶体 20M 以下时，可省外部复位电路

11.共 3 个 16 位定时器/ 计数器，其中定时器 0 还可以当成 2 个 8 位定时器使用

12.外部中断 4 路,下降沿中断或低电平触发中断,Power Down 模式可由外部中断低电平触发中断方式唤醒

13. 通用异步串行口(UART)，还可用定时器软件实现多个 UART

14.工作温度范围： 0 - 75℃ / -40 - +85℃

15.封装： LQFP-44,PDIP-40， PLCC-44,PQFP-44,如选择 STC89 系列,请优先选择 LQFP-44 封

编程系统介绍：

软件：keil μVision4

操作系统：Microsoft Windows 10 专业版 (BUILD:17763) (64 位)

CPU 信息：AMD Ryzen 5 2600 Six-Core Processor

步进：2 型号：8 系列：F 扩展型号：08 扩展系列：80F

主板信息：Gigabyte (技嘉) B450M DS3H-CF

内存信息：16.0 GB (8.0 GB / 8.0 GB /)

显卡信息：NVIDIA GeForce GTX 760

三、 硬件设计

Vcc：接+5V 电源正端

GND：接+5V 电源地端

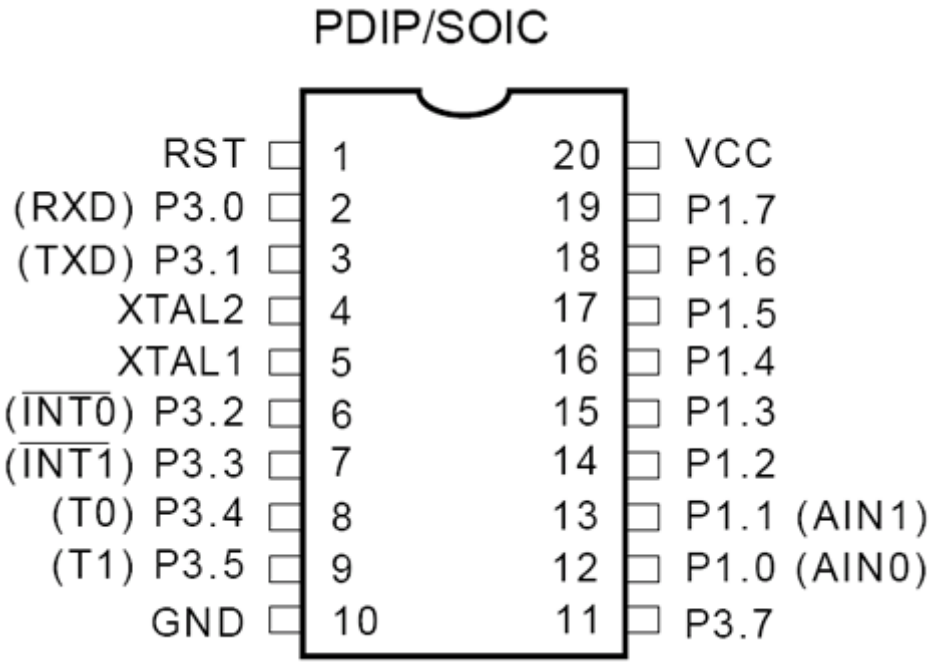
P1.0—P1.7：完整的双向串行通讯接口，P1.0 与 P1.1 另有第二种功能

P3.0—P3.7：除 P3.6 外，双向 I/O 口，除 P3.7 外，均有第二功能，与 MCS-51 系列单片机基本相同

XTAL1：震荡器反向放大器里面运行时钟输入端

XTAL2：震荡器反向放大器的输出端

RST：复位引脚，震荡器工作时，该引脚上两个机器周期的高电平复位



AT89C2051 引脚图

红外线遥控电路设计

发射部分包含键盘矩阵、编码调制、LED 红外发送器。

发射采用脉宽调制的串行码，以脉宽为 0.565ms、间距 0.56ms、周期为 1.125ms 的组合表示二进制的“0”；以脉宽为 0.565ms、间距 1.685ms、周期为 2.25ms 的组合表示二进制的“1”，其波形如图 3-2 所示。

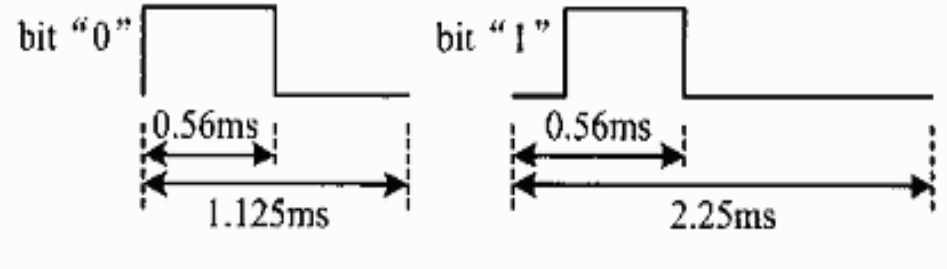


图 遥控码的“0”和“1”

上述“0”和“1”构成的 32 位二进制码经 38kHz 的载频实行二次调制以提升发射效率，实现降低电源功耗的目的。然后通过红外发射二极管产生的红外发射到太空。

编码器产生的遥控编码是 32 位二进制码组，其中前 16 位为用户识别码，能区分不相同的电器设备，避免不同机种遥控码互相干预。芯片的用户识别码固定为十六进制 01H；后 16 位为 8 位操作码（功能码）及其反码。遥控信号编码波形图如图 3-3 所示。

图 遥控信号编码波形图

遥控器在按键按下后，周期性地发出一种 32 位二进制码，周期约为 108ms。一组码本身的持续时间随它包括的二进制“0”和“1”的个数不同而分歧，大约在 45～63ms 之间，图 3-4 为遥控信号的周期性波形图。

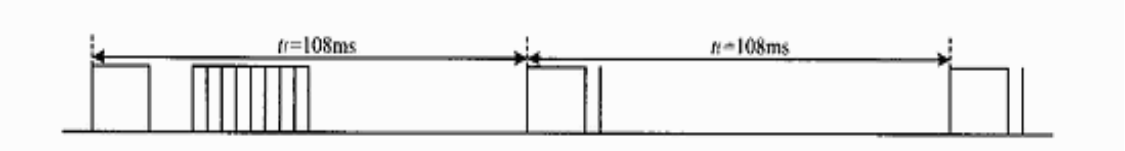


图 遥控信号的周期性波形

当一个键按下超出 36ms，振荡器使芯片激活，将发射一组 108ms 的编码脉冲,这 108ms 发射代码由一个起始码（9ms）,一个结果码（4.5ms）,低 8 位地址码（9ms~18ms）,高 8 位地址码（9ms~18ms）,8 位数据码（9ms~18ms）和这 8 位数据的反码（9ms~18ms）构成。假如键按下超出 108ms 仍未放开，接下来发射的代码（连发代码）将仅由起始码（9ms）和结束码（2.5ms）构成。

按照红外发射管本身的物理特征，必需要有载波信号与将要发射的信号相“与”，而后将相“与”后的信号送发射管，才进行红外信号的发射传递，而在频率为 38KHz 的载波信号下，发射管的性能最佳，发射间隔最远，故本论文选用 38KHz 的晶振发出载波信号，与发射信号进行逻辑“与”运算后，经过三极管的功率启动到红外发光二极管上。

红外发送电路由 4001MOS 或非门 38KHz 振荡器，单片机发送控制电路和红外发送管驱动输出电路构成，当单片机 P3.4 口输出为“0”时，发射管不发光，当单片机 P3.4 口输出为“1”时，红外发送管发出 38KHz 调制红外线。

具体的发射波形如下图所示。

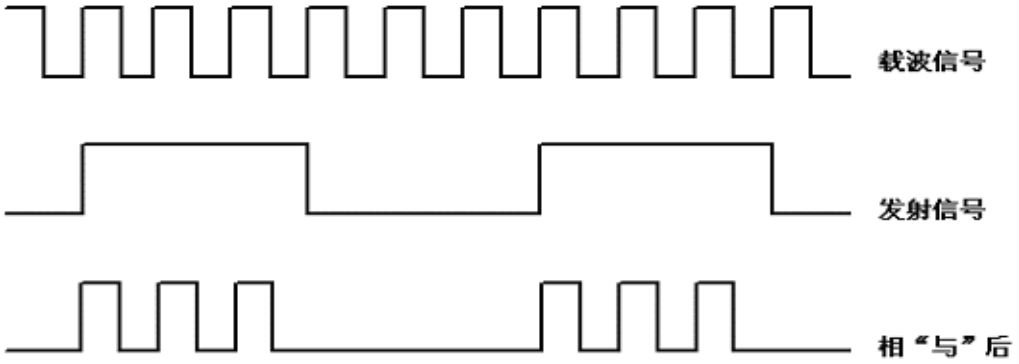


图 调制过程中的波形

红外线经过红外发光二极管发射出去，红外发光二极管是特别的发光二极管，其内部材料和普通发光二极管有差别，因而在其两头施加一定电压时，它发出的是红外线而不是可见光。当今大批利用的红外发光二极管发出的红外线波长约 940nm，外形与普通发光二极管一样。

四、 框图或原理图

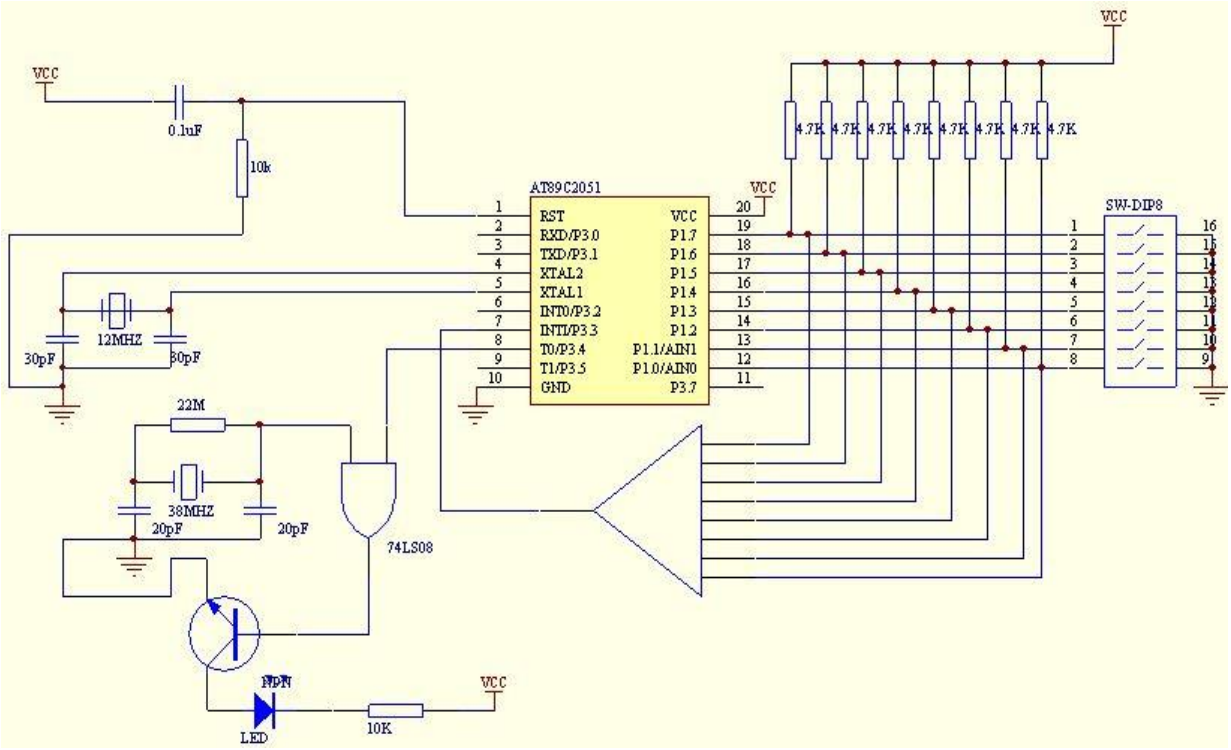


图 红外发射电路图

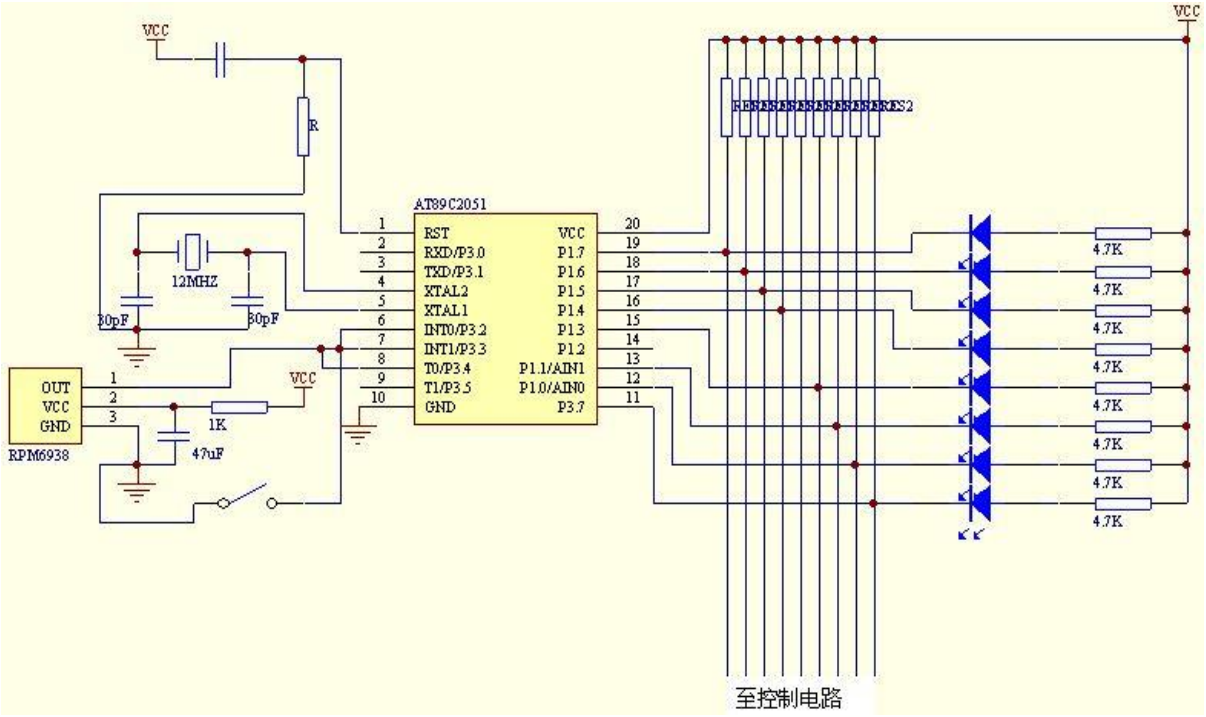


图 红外接收电路图

五、 设计说明

1. 遥控发射部分

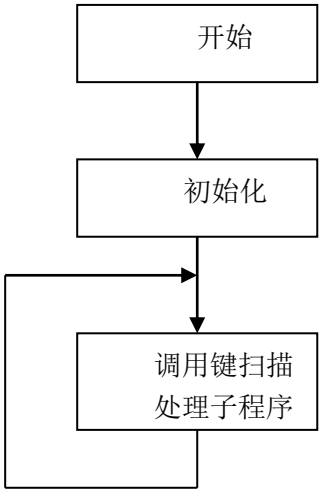


图 遥控发射主程序

上图是遥控发射的主程序，首先初始化程序,然后调用键扫描处理子程序。

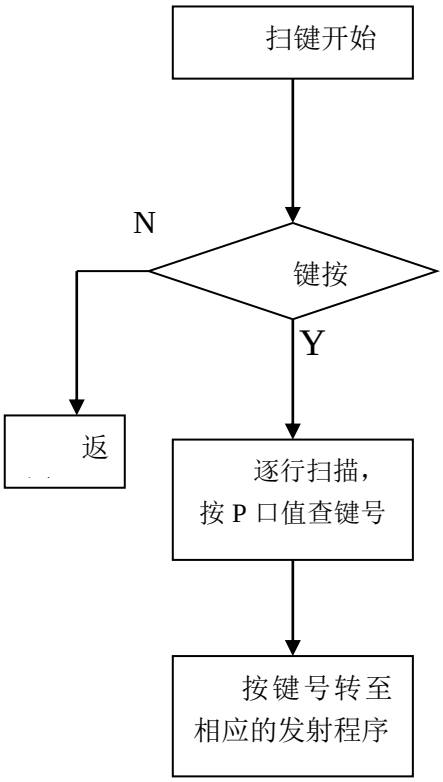


图 扫键过程流程图

扫键过程：首先判断控制键是否按下，若有控制键按下则进行逐行扫描，按照 P 口值查找键号，最后按照键号转至相应的发射程序如下所示。

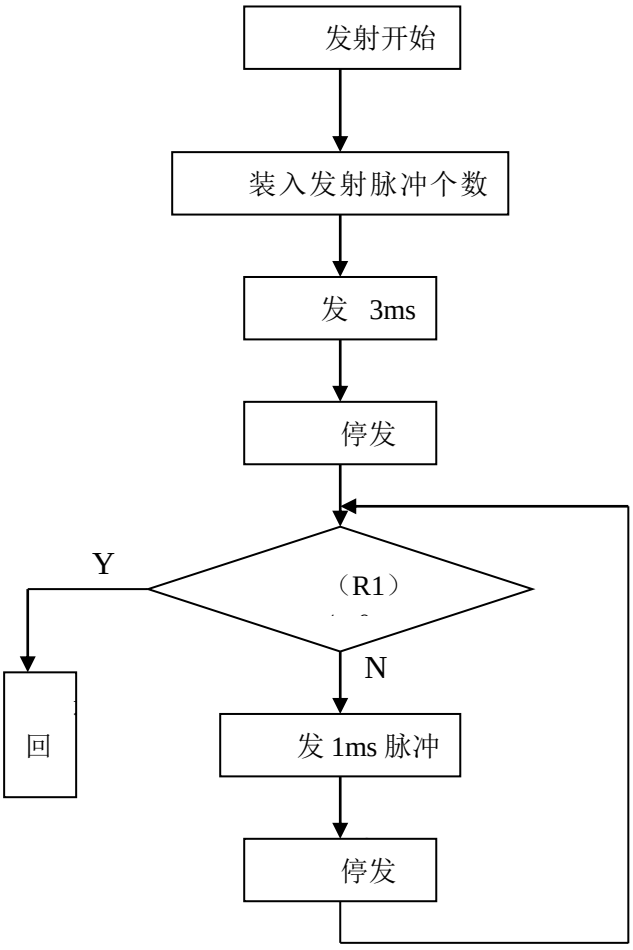


图 红外信号发射程序

红外信号发射过程：首先装入发射脉冲个数(发射时为 3ms 脉冲，停发时为 1ms 脉冲)，此时若发射脉冲个数为 1 则返回主程序，若不为 1 则发 1ms 脉冲，然后停发 1ms 脉冲，这样便结束整个发射过程。

在实践中，采用红外线遥控方式时，由于受遥控距离，角度等影响，使用效果不是很好，如采用调频或调幅发射接收码，可提高遥控距离，并且没有角度影响。

2. 遥控接收部分

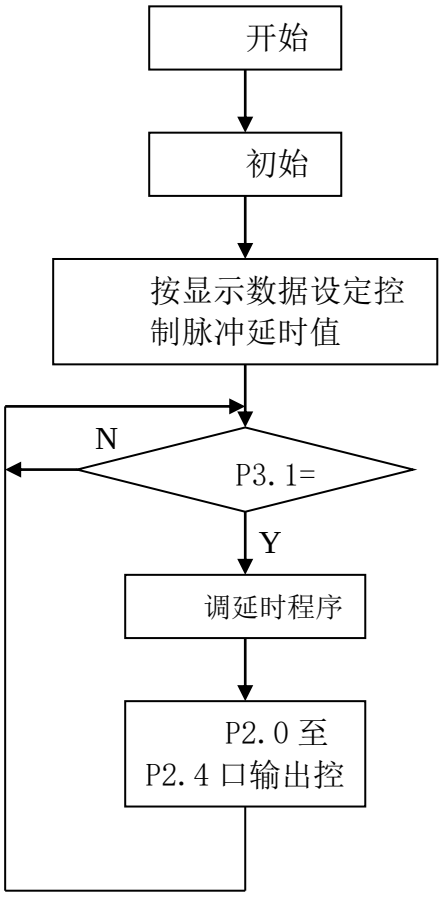


图 遥控接收部分主程序

遥控接收部分的主程序及初始化及延时过程如上：首先初始化，然后按照显示数据设定控制脉冲延时值，看 P3.1 口的脉冲是否为 0，若不为 0 则调入延时程序，此时 P2.7 口输出控制脉冲然后返回；若为 0 则直接返回。

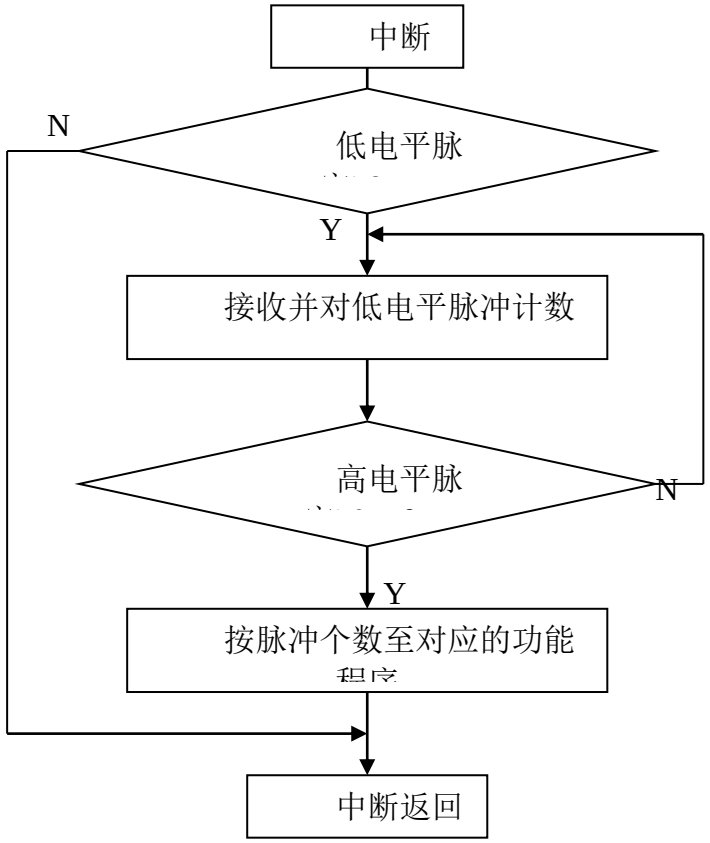
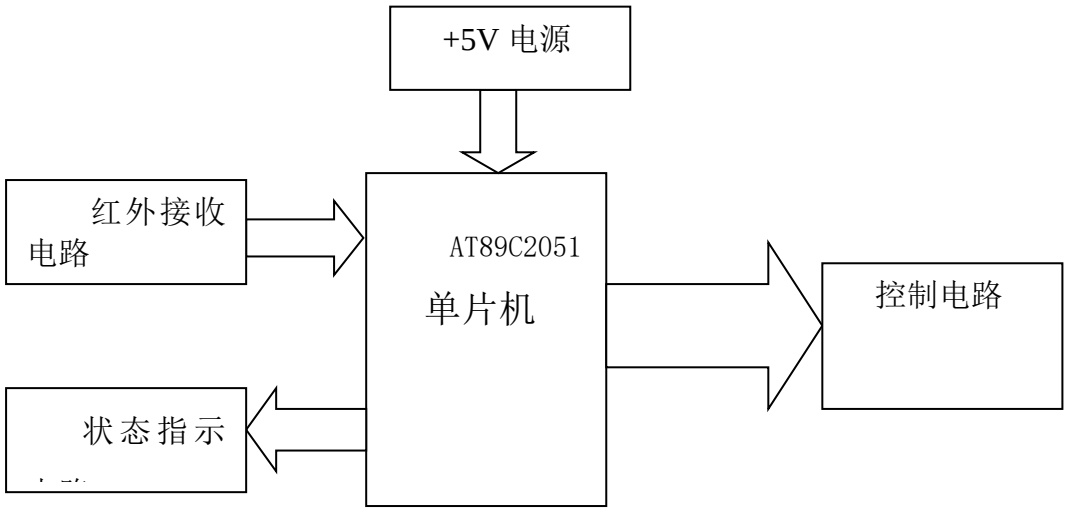


图 中断过程程序

中断过程：首先判断低电平脉宽度是否大于 2ms,若脉宽不到 2ms，则中断返回；若低电平大于 2ms，则接收并地低电平脉冲计数，接下来看判断高电平脉宽度冲是否大于 3ms，若脉宽不到 3ms，则返回上一接收计数过程；若高电平脉宽大于 3ms，则按照脉冲个数至对应功能程序.此时中断返回。

六、 主要流程图



七、 关键代码+注释

主函数代码展示

```
#include<reg51.h>
#include<intrins.h>
#include"XPT2046.h"

sbit IRIN=P3^2;
sbit SRCLK=P3^6;
sbit RCLK=P3^5;
sbit SER=P3^4;

#define COMMONPORTS          P0


unsigned char IrValue[6];
unsigned char Time;
void IrInit();
void DelayMs(unsigned int );
/************************************************************************************
*****
* 函数名             : main
* 函数功能             ： 主函数
* 输入               ： 无
* 输出               ： 无
************************************************************************************
****/

//--列选通控制--//
unsigned char code TAB[8]    = {0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};

//--点阵字码--//
unsigned char code CHARCODE1[18][8]=
{

{0x00,0x00,0x00,0x18,0x18,0x00,0x00,0x00}, //0

{0x00,0x00,0x3C,0x3C,0x3C,0x3C,0x00,0x00}, //1
```

```
{0x00,0x7E,0x7E,0x7E,0x7E,0x7E,0x7E,0x00}, //2

{0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF}, //3

};
unsigned char code CHARCODE2[18][8]=
{
{0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF},

{0x00,0x7E,0x7E,0x7E,0x7E,0x7E,0x7E,0x00},

{0x00,0x00,0x3C,0x3C,0x3C,0x3C,0x00,0x00},

{0x00,0x00,0x00,0x18,0x18,0x00,0x00,0x00}, //0

};
unsigned char code CHARCODE3[65][8]=
{

{0x00,0x00,0x00,0x00,0x08,0x00,0x00,0x00}, //0

{0x00,0x00,0x00,0x08,0x08,0x00,0x00,0x00}, //1

{0x00,0x00,0x00,0x18,0x08,0x00,0x00,0x00}, //2

{0x00,0x00,0x00,0x18,0x18,0x00,0x00,0x00}, //3

{0x00,0x00,0x00,0x18,0x18,0x10,0x00,0x00}, //3

{0x00,0x00,0x00,0x18,0x18,0x18,0x00,0x00}, //3

{0x00,0x00,0x00,0x18,0x18,0x1C,0x00,0x00}, //3

{0x00,0x00,0x00,0x18,0x1C,0x1C,0x00,0x00}, //3

{0x00,0x00,0x00,0x1C,0x1C,0x1C,0x00,0x00}, //3

{0x00,0x00,0x04,0x1C,0x1C,0x1C,0x00,0x00}, //3

{0x00,0x00,0x0C,0x1C,0x1C,0x1C,0x00,0x00}, //3

{0x00,0x00,0x1C,0x1C,0x1C,0x1C,0x00,0x00}, //3
```

```
{0x00,0x00,0x3C,0x1C,0x1C,0x1C,0x00,0x00}, //3

{0x00,0x00,0x3C,0x3C,0x1C,0x1C,0x00,0x00}, //3

{0x00,0x00,0x3C,0x3C,0x3C,0x1C,0x00,0x00}, //3

{0x00,0x00,0x3C,0x3C,0x3C,0x3C,0x00,0x00}, //3

{0x00,0x00,0x3C,0x3C,0x3C,0x3C,0x20,0x00}, //3

{0x00,0x00,0x3C,0x3C,0x3C,0x3C,0x30,0x00}, //3

};

unsigned char code CHARCODE4[18][8]=
{
{0x00,0x00,0x00,0x18,0x18,0x00,0x00,0x00}, //0

{0x00,0x00,0x3C,0x3C,0x3C,0x3C,0x00,0x00},


};

unsigned char code CHARCODE5[18][8]=
{
{0x00,0x00,0x3C,0x3C,0x3C,0x3C,0x00,0x00},

{0x00,0x00,0x00,0x18,0x18,0x00,0x00,0x00},


};

void delay(unsigned int time)
{
   unsigned int i,j;
   for(i=0;i<time;i++)
      for(j=0;j<121;j++);
}

/*****************************************************************************
*****
* 函数名             : Hc595SendByte(unsigned char dat)
* 函数功能             ： 想 74H595 发送一个字节的数据
* 输入               ： 无
```

```
    * 输出                  : 无
    ******************************************************************************
****/
    void Hc595SendByte(unsigned char dat)
    {
        unsigned char a;
        SRCLK=0;
        RCLK=0;
        for(a=0;a<8;a++)
        {
            SER=dat>>7;
            dat<<=1;

            SRCLK=1;
            _nop_();
            _nop_();
            SRCLK=0;
        }

        RCLK=1;
        _nop_();
        _nop_();
        RCLK=0;
    }

    void main()
    {
        unsigned char i;
        unsigned char tab, j;
        uint temp;
        IrInit();
        while(1)
        {
            temp = Read_AD_Data(0xA4);          //     AIN2  光敏电阻

            if(temp > 70){
                IrValue[5]=IrValue[2]&0x0f;       //低位
                if(IrValue[5] == 6) {
                    for(j = 0;j < 4; j++){
                        for(i=0;i<30;i++){
                            for(tab=0;tab<8;tab++)
                                {
                                Hc595SendByte(0x00);                    //消隐
```

```
                    COMMONPORTS = TAB[tab];                    // 输 出
字码
                    Hc595SendByte(CHARCODE1[j][tab]);
                    delay(2);
                    }
                }
            }
        }
        if(IrValue[5] == 12) {
            for(j = 0;j < 4; j++){
                for(i=0;i<30;i++){
                    for(tab=0;tab<8;tab++)
                    {
                    Hc595SendByte(0x00);                    //消隐

                    COMMONPORTS = TAB[tab];                    // 输 出
字码
                    Hc595SendByte(CHARCODE2[j][tab]);
                    delay(2);
                    }
                }
            }
        }
        if(IrValue[5] == 8) {
            for(j = 0;j < 65; j++){
                for(i=0;i<5;i++){
                    for(tab=0;tab<8;tab++)
                    {
                    Hc595SendByte(0x00);                    //消隐

                    COMMONPORTS = TAB[tab];                    // 输 出
字码
                    Hc595SendByte(CHARCODE3[j][tab]);
                    delay(2);
                    }
                }
            }
        }
        if(IrValue[5] == 14) {
            for(j = 65;j > 0; j--){
                for(i=0;i<5;i++){
                    for(tab=0;tab<8;tab++)
                    {
                    Hc595SendByte(0x00);                    //消隐
```

```
                                COMMONPORTS = TAB[tab];                    // 输 出
字码
                                Hc595SendByte(CHARCODE3[j-1][tab]);
                                delay(2);
                                }
                            }
                        }
                    }

                }
                else{
                    IrValue[5]=IrValue[2]&0x0f;        //低位
                    if(IrValue[5] == 6) {
                        for(j = 0;j < 2; j++){
                            for(i=0;i<30;i++){
                                for(tab=0;tab<8;tab++)
                                {
                                Hc595SendByte(0x00);                    //消隐

                                COMMONPORTS = TAB[tab];                    // 输 出
字码
                                Hc595SendByte(CHARCODE4[j][tab]);
                                delay(2);
                                }
                            }
                        }
                    }
                    if(IrValue[5] == 12) {
                        for(j = 0;j < 2; j++){
                            for(i=0;i<30;i++){
                                for(tab=0;tab<8;tab++)
                                {
                                Hc595SendByte(0x00);                    //消隐

                                COMMONPORTS = TAB[tab];                    // 输 出
字码
                                Hc595SendByte(CHARCODE5[j][tab]);
                                delay(2);
                                }
                            }
                        }
                    }
                    if(IrValue[5] == 8) {
```

```
                    for(j = 0;j < 16; j++){
                        for(i=0;i<5;i++){
                            for(tab=0;tab<8;tab++)
                                {
                                Hc595SendByte(0x00);                    //消隐

                                COMMONPORTS = TAB[tab];                 // 输 出
字码
                                Hc595SendByte(CHARCODE3[j][tab]);
                                delay(2);
                                }
                            }
                        }
                    if(IrValue[5] == 14) {
                        for(j = 16;j > 0; j--){
                            for(i=0;i<5;i++){
                                for(tab=0;tab<8;tab++)
                                {
                                Hc595SendByte(0x00);                    //消隐

                                COMMONPORTS = TAB[tab];                 // 输 出
字码
                                Hc595SendByte(CHARCODE3[j-1][tab]);
                                delay(2);
                                }
                            }
                        }
                    }

                }


        }
    }
    /*****************************************************************************
*****
    * 函数名          : DelayMs()
    * 函数功能          ： 延时
    * 输入            : x
    * 输出            ： 无
    *****************************************************************************
****/
```

```
    void DelayMs(unsigned int x)     //0.14ms 误差  0us
    {
     unsigned char i;
       while(x--)
     {
       for (i = 0; i<13; i++)
     {}
     }
    }
    /**************************************************************************
*****
    * 函数名             : IrInit()
    * 函数功能             ：初始化红外线接收
    * 输入               ：无
    * 输出               ：无
    **************************************************************************
****/

    void IrInit()
    {
        IT0=1;//下降沿触发
        EX0=1;//打开中断 0 允许
        EA=1;    //打开总中断

        IRIN=1;//初始化端口
    }
    /**************************************************************************
*****
    * 函数名             : ReadIr()
    * 函数功能             ：读取红外数值的中断函数
    * 输入               ：无
    * 输出               ：无
    **************************************************************************
****/

    void ReadIr() interrupt 0
    {
        unsigned char j,k;
        unsigned int err;
        Time=0;
        DelayMs(70);

        if(IRIN==0)         //确认是否真的接收到正确的信号
        {
```

```
            err=1000;                   //1000*10us=10ms,超过说明接收到错误的信号
            /*当两个条件都为真是循环，如果有一个条件为假的时候跳出循环，免得程序
出错的时
            侯，程序死在这里*/
            while((IRIN==0)&&(err>0)) //等待前面 9ms 的低电平过去
            {
                DelayMs(1);
                err--;
            }
            if(IRIN==1)                 //如果正确等到 9ms 低电平
            {
                err=500;
                while((IRIN==1)&&(err>0))         //等待 4.5ms 的起始高电平过去
                {
                    DelayMs(1);
                    err--;
                }
                for(k=0;k<4;k++)          //共有 4 组数据
                {
                    for(j=0;j<8;j++)     //接收一组数据
                    {

                        err=60;
                        while((IRIN==0)&&(err>0))//等待信号前面的 560us 低电平过去
//                      while (!IRIN)
                        {
                            DelayMs(1);
                            err--;
                        }
                        err=500;
                        while((IRIN==1)&&(err>0))   //计算高电平的时间长度。
                        {
                            DelayMs(1);//0.14ms
                            Time++;
                            err--;
                            if(Time>30)
                            {
                                EX0=1;
                                return;
                            }
                        }
                        IrValue[k]>>=1;       //k 表示第几组数据
                        if(Time>=8)               //如果高电平出现大于 565us，那么是 1
```

```
                {
                    IrValue[k]|=0x80;
                }
                Time=0;         //用完时间要重新赋值

            }
        }
    }
    if(IrValue[2]!=~IrValue[3])
    {
        return;
    }
  }
}
```

八、 实验结果

通过红外遥控器可以使用 0~3 来控制 LED 点阵从内到外和从外到内，螺旋扩大和螺旋缩小的动画效果，同时，我们使用手遮挡住光敏传感器可以使点阵从 8 乘 8 缩小成 4 乘 4 的小方阵。

九、 结果说明和操作方法说明

结果说明我们想要的效果已经实现，我们通过使用按键定义，中断和定时器的组合实现了可以人机交互的 LED 点阵动画展示。

十、 实验结果截图或照片