毕业设计

基于单片机的交通信号的灯控制系统

一．综合实训的主要内容

1.设计任务

设计一单片机控制的交通信号灯系统，模拟城市十字路口交通信号灯功能。

2.基本功能要求

2.1 交通信号控制

直行车道红黄绿灯控制、左行车道绿灯控制、人行横道红绿灯控制。

2.2 通行时间显示

数码管倒计时显示通行时间。

2.3 时间参数设置存储

按键实现通行时间的设置，并存储到EEPROM（24C02）芯片中。

二．硬件方案设计与论证

1. 显示模块设计

1.1倒计时时间显示

设计思想：由于该系统要求完成倒计时显示通行时间的功能，且考虑到实际的交通系统中车辆及行人通行时间不会超过一分钟，基于以上原因，我们考虑完全采用数码管显示，四个路口分别采用一个二位共阴极数码管进行显示。（其实物图见附录 1 图 5.3)

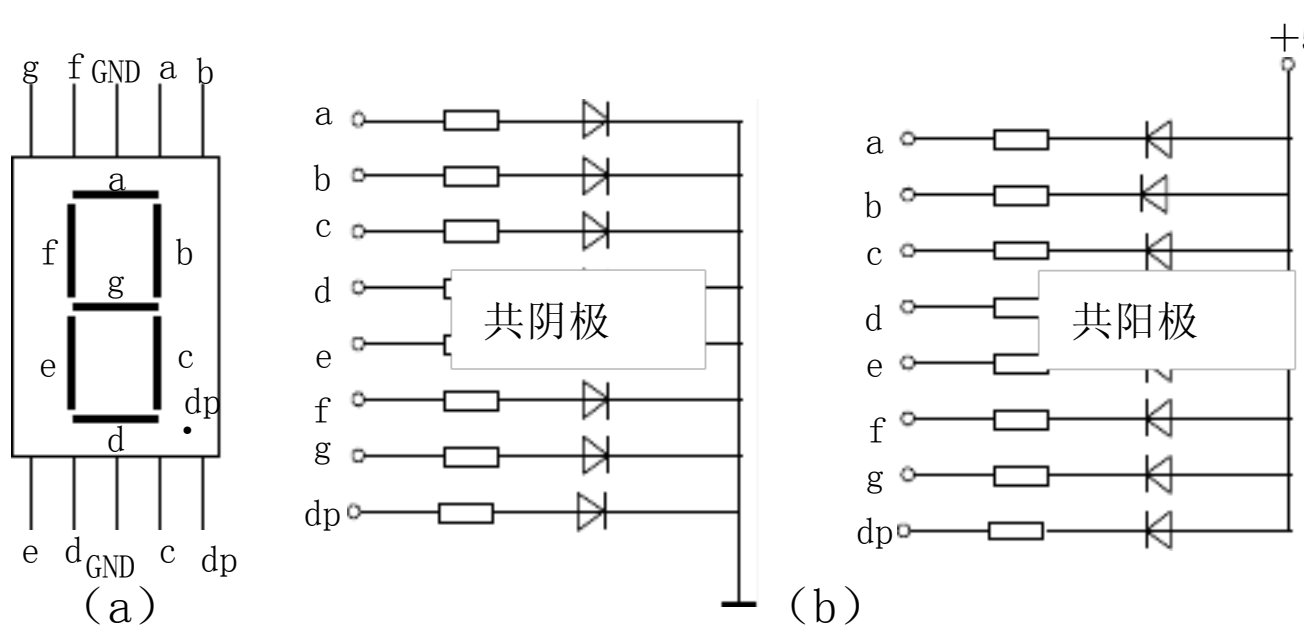


图 2.1 数码管原理图

原理图分析：

为了显示数字或字符，必须对数字或字符进行编码。七段数码管

（a,b,c,d,e,f,g）加上一个小数点(dp)，共计 8 段，构成一个字节，通过对这八段给予高低平使二极管导通或截止，从而显示不同的数字或字符。系统中所使用的是 2 位共阴数码管（实物图见附录），其管脚从左上方起顺时针依次为 1,a,b,e,d,2,g,f,dp,c

1.2状态灯显示

设计思想：由于该系统要求完成状态灯显示的功能，我们把各个路口的红灯和黄灯设成直行和左拐两个通行方式所共有，也就是说，一个路口只需四个状态灯，一个直行通行的绿灯，一个左拐通行的绿灯，一个共有的红灯，一个共有的黄灯，人行横道采用红绿灯控制，综上所述，我们共使用 16 个 LED 绿灯，12 个 LED 红灯，4 个 LED 黄灯来完成状态灯显示功能。

2.控制模块设计

2.1 设计思想

由于本系统结构简单，实现较容易，不需要大量的外围扩展，所以我们采用 STC89C51 单片机作为主控制器，STC89C51 单片机具有体积小，功耗低，控制能力强，价格低、扩展灵活，使用方便等特点，其最小系统由振荡电路、复位电路构成。

2.2 最小系统原理图

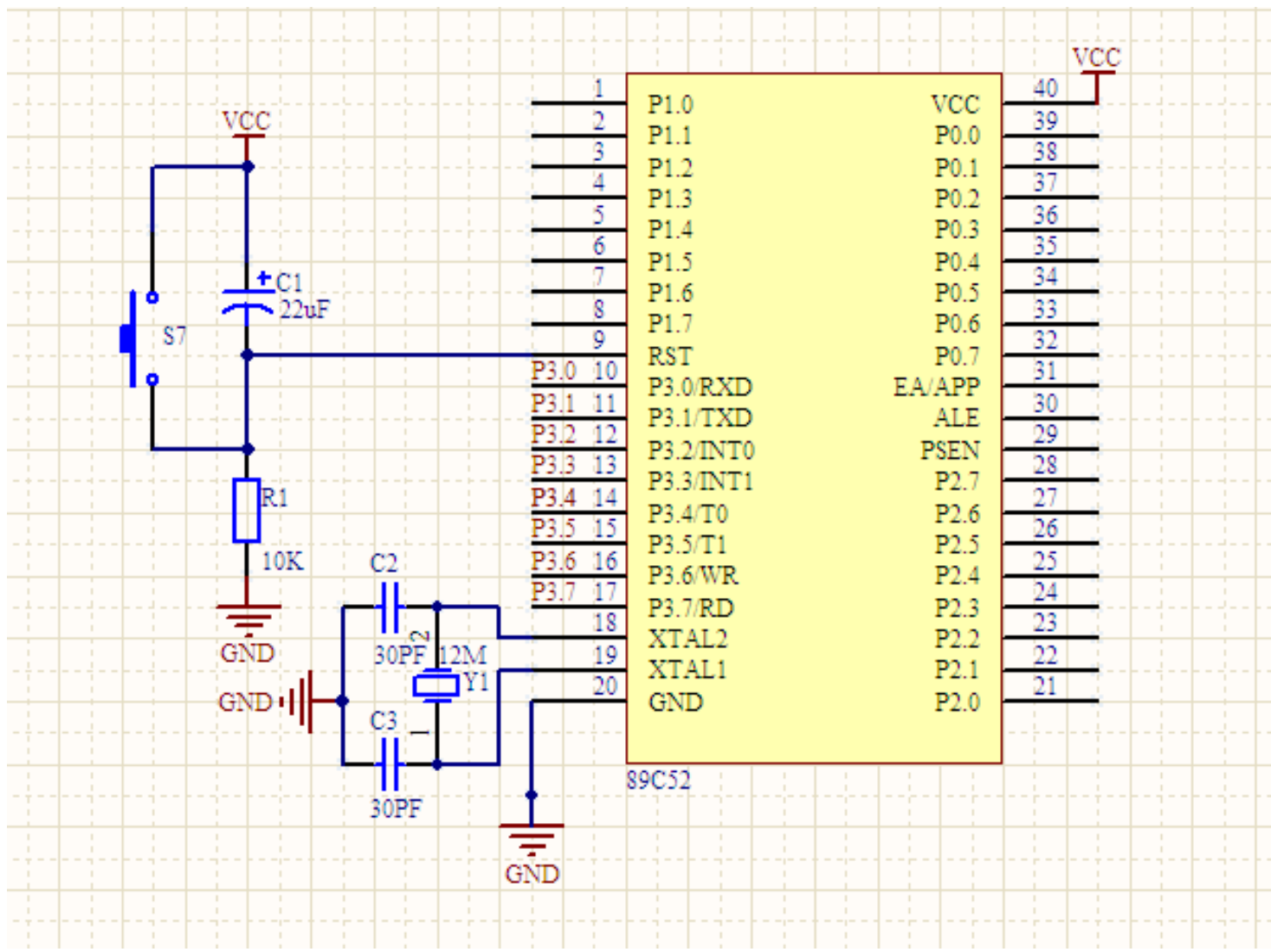


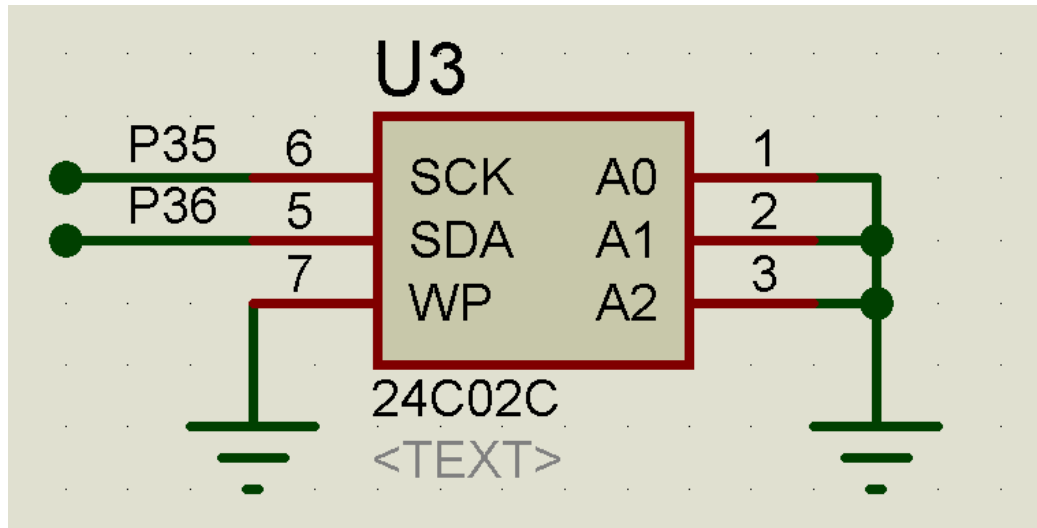
图 2.2 单片机最小系统原理图

原理图分析：51 单片机最小系统由复位电路，振荡电路组成。振荡电路使用 11.0592M Hz 高精度晶振，振荡电容选择 30pF 瓷片电容；复位电路采用 RC 电路。

3.存储模块

3.1 设计思想：系统掉电存储模块采用串行 E2PROM ，它是基于 IIC 总线的存储器件，遵循二线制协议，其具有接口方便，体积小，数据掉电不丢失等特点。

3.2 24C02芯片原理图



管脚描述 ： A0A1A2　引脚 器件地址选择
SDA　引脚 串行数据/地址
SCL 引脚串行时钟
WP　写保护
VCC　电源 1.8~6V
VSS　地

4. 系统理论分析

4.1交通灯显示时序的理论分析

依次循环共分 4 种状态：南北方向直行通行、南北方向左拐通行、东西方向直行通行，东西方向左拐通行。

南北方向直行：南北段直行通行（绿灯）、东西段禁止（红灯），,此时，南北段人行道通行（绿灯），东西段人行道禁止（红灯），同时南北段和东西段方向的数码管分别从 20s 和 30s（加上南北段左拐时间）开始倒计时，至最后 5s 时南北段绿灯变成黄灯闪烁；此后南北段左拐（左拐绿灯亮）通行、东西段禁止（红灯）10s，南北段、东西段人行道都禁止（红灯），同时南北段和东西段方向的数码管都从 10s 开始倒计时，至最后 5s 时南北段左拐灯变成黄灯闪烁；再后东西段直行通行（绿灯）、南北段禁止（红灯），东西段人行道通行（绿灯），南北段人行道禁止（红灯），同时东西段和南北段方向的数码管分别从 20s 和 30s 开始倒计时，至最后 5s 时东西段绿灯变成黄灯闪烁；最后东西段左拐（左拐灯亮）通行、南北段禁止（红灯）10s，东西段、南北段人行道都禁止（红灯），同时东西段和南北段方向的数码管都从 10s 开始倒计时，至最后 5s 时东西段左拐灯变成黄灯闪烁，即完成一次循环。

4.2 状态切换

系统中共设置了四种模式，分别为开始模式、延长通行时间模式、减少左拐时间模式、急停模式，这几种模式分别通过相应的按键进行切换。

开始模式：直行 20s,左拐 10s;

延长通行时间模式：直行 40s，左拐 20s;

减少左拐时间模式：直行 40s,左拐 10s

急停模式：当有紧急事故发生时，所有指示灯全变成红灯，禁止通行，数码管显示 00.

4.3倒计时显示的具体实现

利用定时器中断，设置 TH0=TH1＝(65536-50000)/256，即每 0.05 秒中断一次。每到第 20 次中断即过了 20*0.05 秒＝1 秒时，使时间的计数值减 1，便实现了倒计时的功能。

4.4状态灯显示的实现方法

黄灯闪烁利用定时器中断。每到第 10 次中断即过了 10*0.05 秒＝0.5 秒时，使黄灯标志位反置，即可让黄灯 1 秒闪烁一次。

4.5 状态切换的实现方法

状态切换在定时器中实现，定时器每 0.05 秒中断一次，完全可以检测按键的发生。考虑到实际的交通系统不可能立即切换状态，程序一个周期内检测两次状态，若在南北左拐前按键修改状态，则南北左拐结束后切换状态，若在南北左拐后修改状态，则在东西左拐后切换状态。

三．软件编程

1.程序流程图

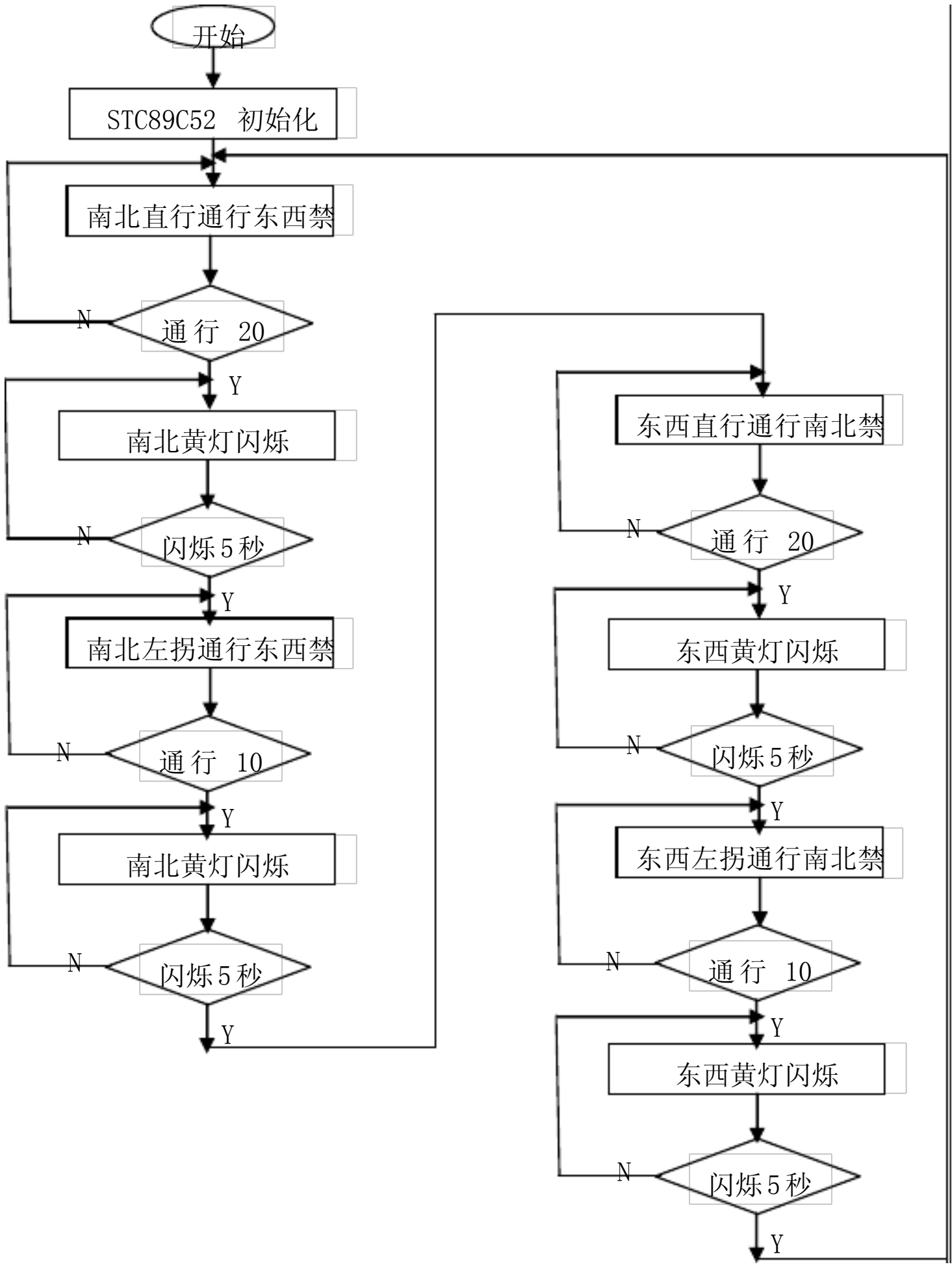


图 3.1 主程序流程图

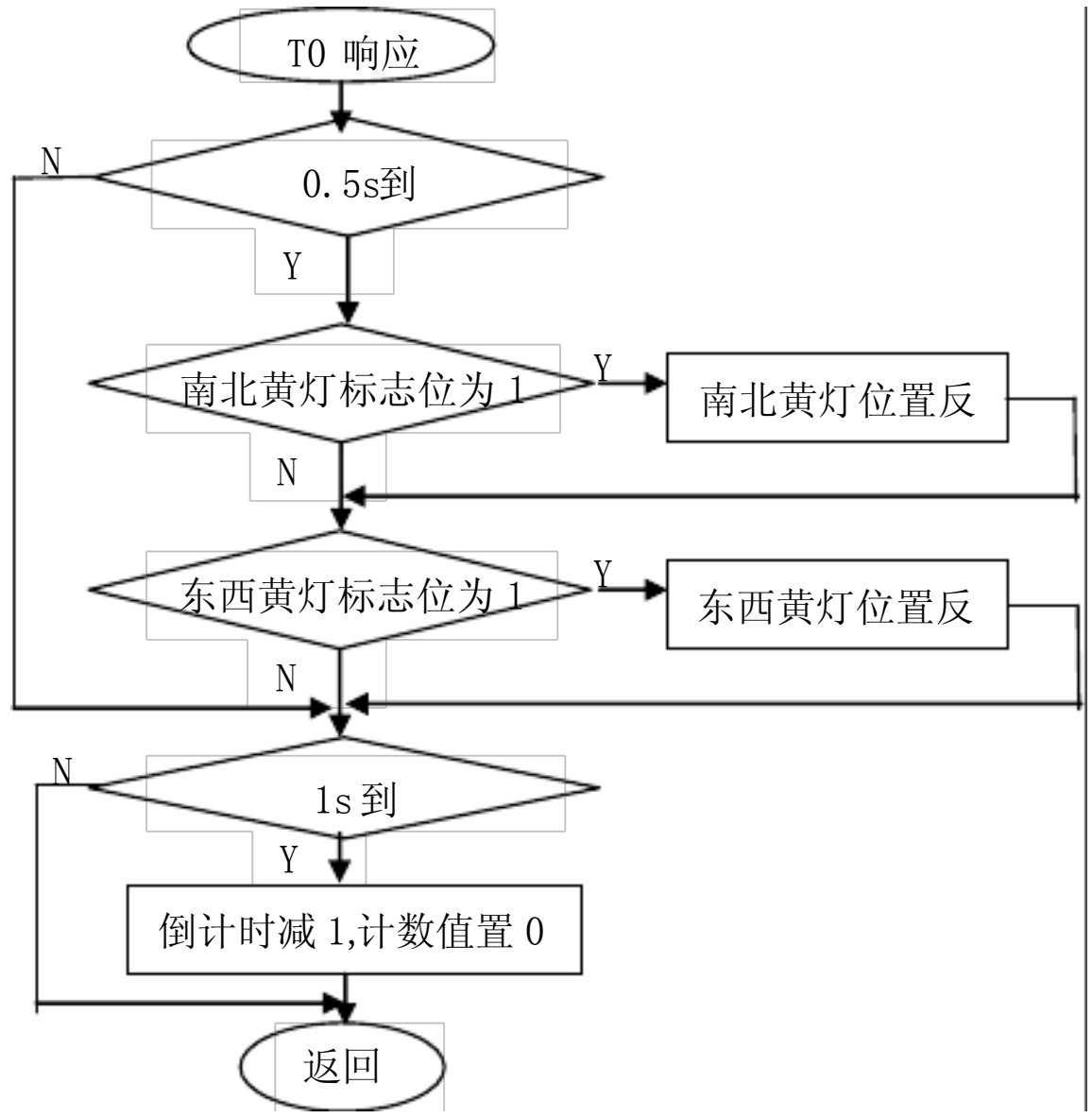


图 3.2 定时 0 中断流程图

2.程序

```
#include<reg52.h>
#include"24C02.h"

/*************端口定义*************/
sbit   EW_ShuMa2=P2^3;          //EW 方向低位数码管控制位
sbit   EW_ShuMa1=P2^2;          //EW 方向高位数码管控制位
sbit   SN_ShuMa2=P2^1;          //SN方向低位数码管控制位
sbit   SN_ShuMa1=P2^0;          //SN方向高位数码管控制位

sbit   SN_Yellow=P1^1;         //SN黄灯
sbit   EW_Yellow=P1^5;          //EW 黄灯
sbit   EW_ManGreen=P2^7;        //EW 人行道绿灯
sbit   SN_ManGreen=P2^4;       //SN人行道绿灯
sbit   EW_ManRed=P2^6;          //EW 人行道红灯
sbit   SN_ManRed=P2^5;          //SN人行道红灯
sbit   EW_Red=P1^6;             //EW 直行道红灯
```

```
sbit  SN_Red=P1^2;              //SN直行道红灯

sbit shezhi1=P3^0;          /模式设置键
sbit shezhi2=P3^1;          /模式设置键
sbit stop1=P3^7;            /紧急情况键

bit  Flag_SN_Yellow;         //SN黄灯标志位
bit  Flag_EW_Yellow;          //EW 黄灯标志位

char   Time_EW;                  /东西方向倒计时单元
char   Time_SN;                  /南北方向倒计时单元

    uchar  EW=30,SN=20,EWL=9,SNL=9;          /程序初始化赋值，正常模式
    uchar  EW1=30,SN1=20,EWL1=9,SNL1=9;
    uchar code table[10]={0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F};
//0-9段选码
    uchar code State[8]={0xBE,0xBD,0xB7,0xBD,0xEB,0xDB,0x7B,0xDB};
    /*************函数声明***************/
    void Delay_ms(uint z);
    void ShuMa_Display();
    void timer0_init();
    void state1();
    void state2();
    void state3();
    void state4();
    /**************主函数************/
    void main()
{
   timer0_init();    /定时器初始化
   EW1=read(10);    //24c02读操作
   SN1=read(11);
   EWL1=read(12);
   SNL1=read(13);
   EW=EW1;
   SN=SN1;
   EWL=EWL1;
   SNL=SNL1;
   //Int_init();
```

```
    while(1)
    {
        state1();
        state2();
        state3();
        state4();
    }
}

/*************延时函数*************/
/*void Delay_ms(uint z)
{  uint x,y;
    for(x=120;x>0;x--)
        for(y=z;y>0;y--);
}*/
/*************数码管显示函数*****************/
void ShuMa_Display()
{
    uchar H,L;
    H=Time_EW/10;
    L=Time_EW%10;

    P0=table[L];
    EW_ShuMa2=0;     /点亮 EW_LED2
    Delay_ms(1);
    EW_ShuMa2=1;         /熄灭 EW_LED2
    P0=table[H];
    EW_ShuMa1=0;          /点亮 EW_LED1
    Delay_ms(1);
    EW_ShuMa1=1;

    H=Time_SN/10;
    L=Time_SN%10;
    P0=table[L];
    SN_ShuMa2=0;          /点亮 SN_LED2
    Delay_ms(1);
    SN_ShuMa2=1;
    P0=table[H];
```

```
    SN_ShuMa1=0;          /点亮 SN_LED1
    Delay_ms(1);
    SN_ShuMa1=1;

}
/***************模式函数*****************/
void state1()
{
    /*********SN 通行 SW  禁止*********/
        SN_ManRed=1;
        SN_ManGreen=0;   //SN人行道通行
        EW_ManRed=0; //EW  人行道禁止
        EW_ManGreen=1;
        Flag_SN_Yellow=0; //SN关黄灯显示信号
        Flag_EW_Yellow=0; //EW 关黄灯显示信号
        Time_EW=EW;
        Time_SN=SN;
        while(Time_SN>=5)
        {
            P1=State[0]; //SN绿灯，EW  红灯
            ShuMa_Display();
        }
        P1=0xff;/关闭 P1 口所有灯
        while(Time_SN>=0)
        {
            Flag_SN_Yellow=1; //SN开黄灯信号位
            EW_Red=0;
            //P1=P1 |0XB0;      /保持 EW  红灯
            ShuMa_Display();
        }
}
 void state2()
 {
        /***********SN 方向左拐状态**********/
        SN_ManRed=0;          //SN人行道禁止
        SN_ManGreen=1;
        EW_ManRed=0;     //EW 人行道禁止
        EW_ManGreen=1;
```

```
Flag_SN_Yellow=0; //SN关黄灯显示信号
Flag_EW_Yellow=0; //EW 关黄灯显示信号
Time_SN=SNL;
Time_EW=EWL;
while(Time_SN>=5)
{
    P1=State[2];    //SN左拐绿灯亮，EW  红灯
    ShuMa_Display();
}
P1=0xFF;
while(Time_SN>=0)
{
        Flag_SN_Yellow=1; //SN开黄灯信号位
         Flag_EW_Yellow=1;
        ShuMa_Display();
      }
}
void state3()
{
  /**********赋值*********/
        EW=EW1;
        SN=SN1;
        EWL=EWL1;
        SNL=SNL1;

       /*******SN 禁止，EW  通行状态**********/
      SN_ManRed=0;        //SN人行道禁止
SN_ManGreen=1;
EW_ManRed=1;
EW_ManGreen=0;      //EW 人行道通行
Flag_SN_Yellow=0; //SN关黄灯显示信号
Flag_EW_Yellow=0;//EW 关黄灯显示信号
Time_EW=SN;
Time_SN=EW;
while(Time_EW>=5)
{
     P1=State[4];//EW通行，SN 红灯
     ShuMa_Display();
```

```
        }
        P1=0Xff;
        while(Time_EW>=0)
        {
            Flag_EW_Yellow=1;//EW 开黄灯信号位
            SN_Red=0;
            //P1=P1|0x0B;   /保持 SN 红灯
            ShuMa_Display();
        }
}

void state4()
{
  /***********EW 方向左拐状态**************/
        SN_ManRed=0;        //SN人行道禁止
        SN_ManGreen=1;
        EW_ManRed=0;    //EW 人行道禁止
        EW_ManGreen=1;
        Flag_SN_Yellow=0; //SN关黄灯显示信号
        Flag_EW_Yellow=0;//EW 关黄灯显示信号
        Time_EW=EWL;
        while(Time_EW>=5)
        {
            P1=State[6];    //EW 左拐绿灯亮，SN 红灯
            ShuMa_Display();
        }
        P1=0Xff;
        while(Time_EW>=0)
        {
            Flag_EW_Yellow=1; //EN开黄灯信号位
            Flag_SN_Yellow=1;
            ShuMa_Display();
        }
        /**********赋值*********/
        EW=EW1;
        SN=SN1;
        EWL=EWL1;
        SNL=SNL1;
```

```
}

/****************定时器 0 初始化函数***************/
void timer0_init()
{
    TMOD=0x01;       /定时器 0 工作方式 1
    TH0=(65536-10000)/256; /装初值
    TL0=(65536-10000)%256;
    EA=1; /开总中断
    ET0=1;/开定时器 0 中断
    TR0=1;/启动定时器
}

/*******************定时器 0 服务中断函数***********/
void timer() interrupt 1
{
        uchar t;
       TH0=(65536-50000)/256;
        TL0=(65536-50000)%256;
       t++;
       if(shezhi1==0)
       {
           EW1=60;
           SN1=40;
           EWL1=19;
           SNL1=19;
           write(10,60);//24C02写操作
           write(11,40);
           write(12,19);
           write(13,19);
      }
      if(shezhi2==0)
      {
           EW1=50;
           SN1=40;
           EWL1=9;
```

```
        SNL1=9;
        //write(10,60);
        //write(11,50);
        //write(12,9);
        //write(13,9);
    }
    if(stop1==0)
    {
        P0=table[0];
        EW_ShuMa2=0;
        EW_ShuMa1=0;
        SN_ShuMa2=0;
        SN_ShuMa1=0;
        P1=0XBB;
        EW_ManGreen=1;
        SN_ManGreen=1;
        EW_ManRed=0;
        SN_ManRed=0;
        while(1);
    }


    if(t==10)
     {
        if(Flag_SN_Yellow==1) /测试南北黄灯标志位
            {SN_Yellow=~SN_Yellow;}
        if(Flag_EW_Yellow==1) /测试东西黄灯标志位
            {EW_Yellow=~EW_Yellow;}

     }
    if(t==20)
    {
        Time_EW--;
        Time_SN--;
        if(Flag_SN_Yellow==1) /测试南北黄灯标志位
            {
               SN_Yellow=~SN_Yellow;
            }
```

```
            if(Flag_EW_Yellow==1) /测试东西黄灯标志位
                {
                   EW_Yellow=~EW_Yellow;
                }
            t=0;
        }
}
```

24c02的.h程序

```
#define uchar unsigned char
#define uintunsigned int

sbit SCL=P3^5;
sbit SDA=P3^6;
/************函数声明***********/
void delay();
void Delay_ms(uint z);
void init(void);
void start();
void stop();
void ack();
void noack();
void write_byte(uchar dat);
uchar read_byte(void);
void write(uchar add,uchar dat);
uchar read(uchar add);
void delay(短延时，大约5us
{
   ;  ;;;;;;
}
void Delay_ms(uint z)
{  uint x,y;
       for(y=z;y>0;y--)
       for(x=120;x>0;x--);
}
void init(void)
{
   SDA=1;
   SCL=1;
```

```
}
void start()/启动信号
{
   SDA=1;
   delay();
   SCL=1;
   delay();下降沿到来
   SDA=0;
   delay();
}
void stop()/停止信号
{
   SDA=0;
   delay();
   SCL=1;
   delay();
   SDA=1;//上升沿到来
   delay();
}
void ack()/应答信号
{
   uchar i;
   SCL=1;
   delay();
   while((SDA==1)&&(i<250)) /等待应答，用与语句防止一直不应答
    i++;
    SCL=0;  /若不应答，也将时钟拉低
    delay();
}
void noack()
{
   SDA=1;
   delay();
   SCL=1;
   delay();
   SCL=0;
   delay();
```