浙江大学

题目：《微机接口电路课程设计》

姓名：张钰清

学号：3160101143

教师：李宏娟　林勇刚　唐建中

目　　录

一、设计任务说明 1
二、硬件系统设计 1
2.1 器件选型 1
2.2 原理设计 2
2.2.1 CPU 2
2.2.2 电源供电电路 2
2.2.3 USB 转串口电路 2
2.2.4 共阳发光二极管电路 3
2.2.5 单片机复位电路 3
三、布局和布线设计 4
四、软件系统设计 5
4.1 软件系统结构 5
4.1.1 主函数框图 5
4.1.2 INT0 外部中断框图 5
3.1.3 INT1 外部中断框图 6
3.1.4 T0 定时器中断框图 6
4.2 软件的模块化实现 6
4.2.1 头文件 6
4.2.2 变量定义 7
4.2.3 延时子程序 7
4.2.4 手动控制模式 7
4.2.5 自动-手动模式转换 8
4.2.6 自动控制模式 8
五、系统测试 9
5.1 测试工具和软件 9
5.2 测试方法 9
六、课程设计总结 10
附件：
附件 1 硬件系统原理图（图片） 11
附件 2 程序及注释 11
附件 3 PCB 图（图片） 14

一、 设计任务说明

设计 PCB，模拟浙大玉泉校区正门交通灯的控制流程；其具体任务要求如下：

1、 利用红、绿、黄三色发光管代表红、绿、黄灯；

2、 PCB 上交通灯相对位置和实际的交通灯布置方位基本一致

3、 PCB 上用丝印标注道路和校门。

4、 具有交通灯手动/自动切换功能。手动方式下可通过按键手动设置通行方向。

技术关键：

1、 在于通过 I/O 口控制各交通灯的亮灭；

2、 通过外部中断使得交通灯进入手动状态；

3、 通过查询方式读取交通灯的方向。

成果要求：

1、 制作的 PCB 布局合理、标注完整；

2、 元件焊接牢固，无虚焊，漏焊；

3、 交通灯变换规律和时间周期与实物基本一致；

4、 可以通过串口对交通灯的变换规律和时间周期进行设置。

二、 硬件系统设计

2.1 器件选型

器件选择的总体原则，是根据 data sheet 以及各经典电路模块的接法，进行参考和选择。

最后得出器件清单如下表所示：

元件名	描述	数量	封装
USB-B 普通插座	普通插座/4PIN/250V/1.5A/双排/弯角/母座/2.5mm/USB-B	1	
6 脚自锁开关	6 脚自锁开关/8*8mm	1	
4 脚微动开关	4 脚微动开关/12VDC/30mA/6*6*4.3mm/插件	5	
STC-89C51	STC-89C51	1	
DIP40 插座	DIP40 插座	1	
CH340G	CH340G/USB 转串口 IC	1	SOP-16
22pf 瓷片电容	22pf/50V/瓷片电容	3	RAD0.1
0.1uf 瓷片电容	0.1uf/50V/瓷片电容	3	RAD0.1
30pf 瓷片电容	30pf/50V/瓷片电容	2	RAD0.1
220uF 电解电容	电解电容/105℃/220uF/16V/φ5*7mm/直脚/P=2.0mm	1	RB.2/.4
10uF 电解电容	电解电容/105℃/10uF/16V/φ5*11mm/直脚/P=2.0mm	1	RB.1/.2
330Ω 碳膜电阻	碳膜电阻/1/8W/300Ω±5%	12	AXIAL0.3
10KΩ 碳膜电阻	碳膜电阻/1/8W/10KΩ±5%	2	AXIAL0.3
510Ω 碳膜电阻	碳膜电阻/1/8W/510Ω±5%	2	AXIAL0.3
排阻	10K/排阻	1	A9-103

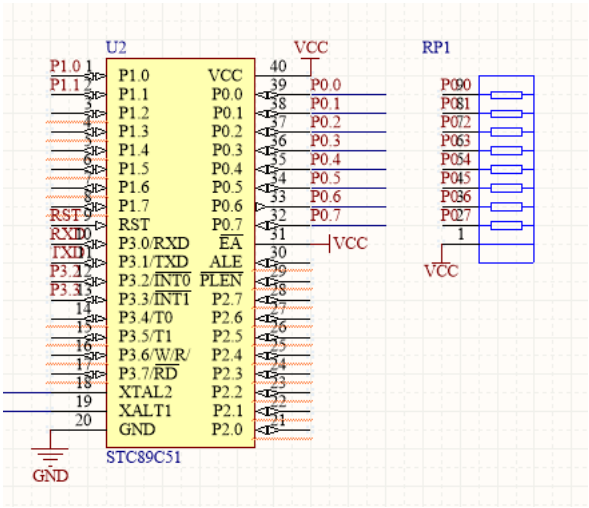
D1,D6,D7,D10,D13	LED 发光二极管/红色/Φ3/80mcd/20m	5	DIODE
D3,D4,D9,D12	LED 发光二极管/绿色/Φ3/16mcd/10mA	4	DIODE
D2,D5,D8,D11	LED 发光二极管/黄色/Φ3/200mcd/20mA	4	DIODE
D14	1N4148/DO35 封装	1	DO35
Y1,Y2	12M 晶振/DIP	2	
USB 下载线	USB 下载线	1	

2.2 原理设计

描述 CPU 单元及外围所有电路的设计方法，器件参数的选型计算

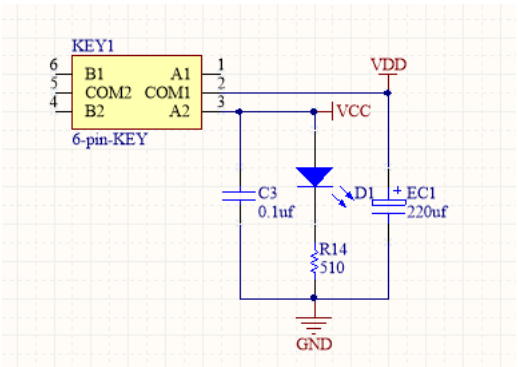
2.2.1 CPU

主芯片采用 STC89C51， P0 端口并接 10k 上拉电阻，从而赋予其驱动能力。



2.2.2 电源供电电路

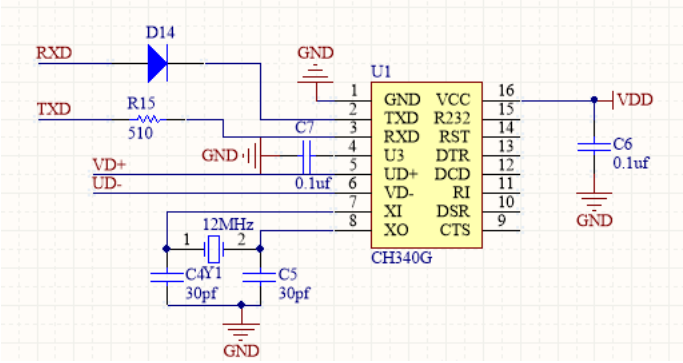
单片机采用 5V USB 供电，需要 USB 供电电路（未写入报告）与电源开关电路相配合。开关电路兼具亮灯指示、大电容滤波的功能。大电容选择 220Uf 电解电容，滤低频，以便滤除直流信号外的所有干扰信号。



2.2.3 USB 转串口电路

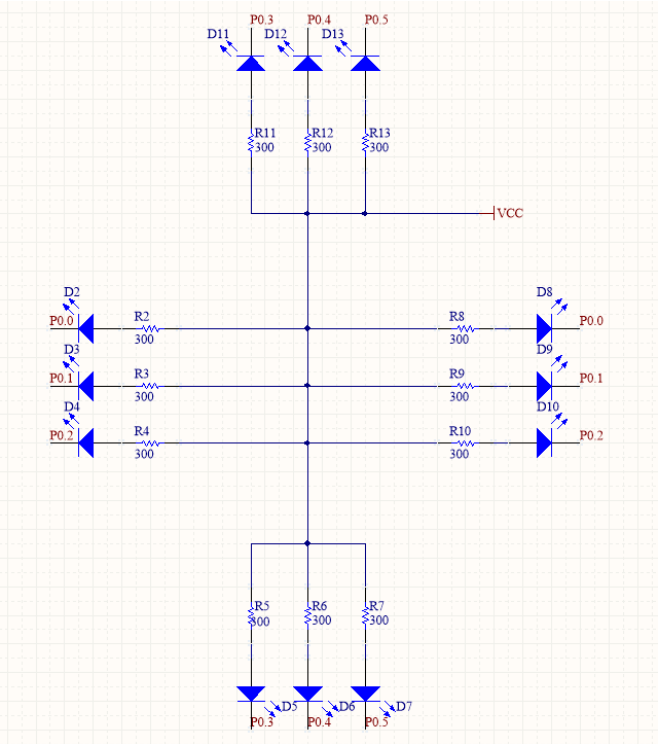
本设计采用 CH340G USB 总线转接芯片。该芯片需要电源持续供电，所以直接连接供电电源 VDD，另

外并接 0.1Uf 小电容,配合电源滤波电容 220Uf 达到全面滤波的效果。使用快速恢复二极管 1N4148 来防止单片机断点不彻底、程序无法下载的情况。



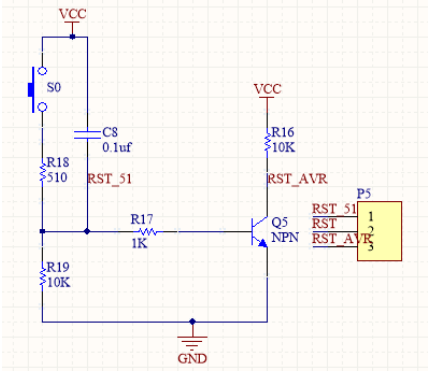
2.2.4 共阳发光二极管电路

将发光二极管经过 300Ω 的电阻连接到电源 VCC 端，发光二极管的负极分别接到 P0,0-P0,5 的管脚处，通过单片机使得 P0,n 变为低电平，使得对应的发光二极管点亮。



2.2.5 单片机复位电路

单片机的复位要求是：RST 引脚输入高电平，并持续 2us 以上。一般单片机都有两种复位方式，其一为上电复位，其二为按键复位。对于后者，我们只用一个开关就能赋予 RST 引脚高电平；对于前者，我们既要保证复位引脚上有大于 2us 的高电平，又要保证在单片机正常工作的时候没有高电平，所以我们采用简单的 RC 电路，使 RST 引脚的电压在一定时间内从高电平下降为低电平。



这里采用了 STC51 开发板中的电路。由于不需要用到其 RST_AVR 的功能，所以直接将 P5 中的 RST51 和 RST 连接起来即可。

三、 布局和布线设计

关于 PCB 板中各个元件的布局，我主要采用了模块化的思想，对其进行分区归集。例如把复位电路、发光二极管电路分区放置再一起，再使他们零件的方向朝着便于布线的方向。在次设计 PCB 按照以下几个规则进行：

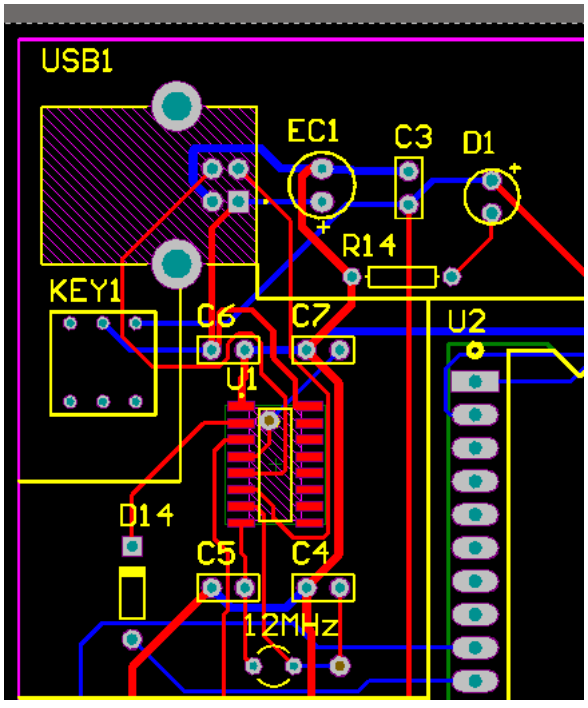
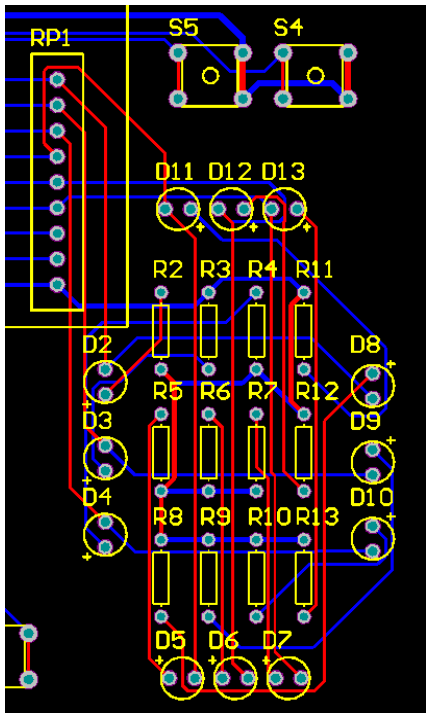
①CPU 放在中间位置；

②一个功能电路的元件尽量放在一起；

③相同类型的元件整齐排在一起；

④同一层线路线路尽量设计成相同走向，比如本设计中蓝线主要走横向，红线主要走纵向；

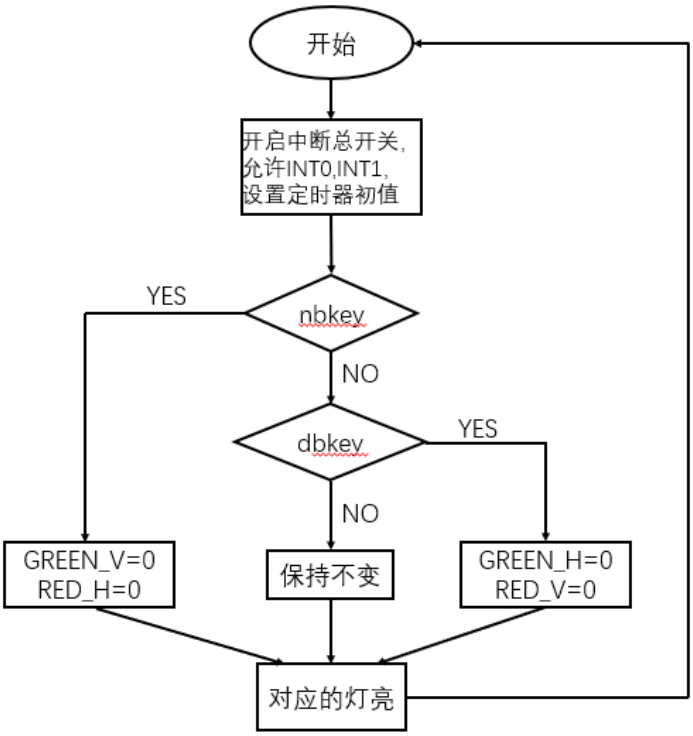
⑤布线完成后做滴泪并铺铜。

开关及烧录电路	共阳发光二极管电路
	
设计遵循的原则是 USB 口的摆放要方便 USB 线的插入，同时将电源开关与 USB 口放在一起，减少布线长度。	由于四面交通灯围城的方块中间还有很大的空间，所以恰好可以把连接在它们与地之间的电阻放在中间的位置处。

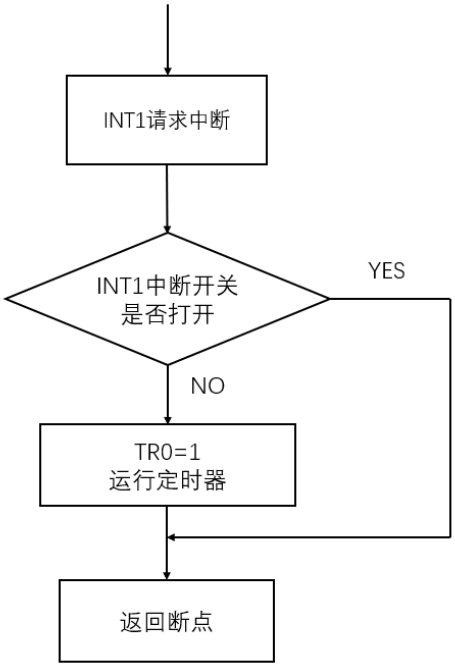
四、 软件系统设计

4.1 软件系统结构

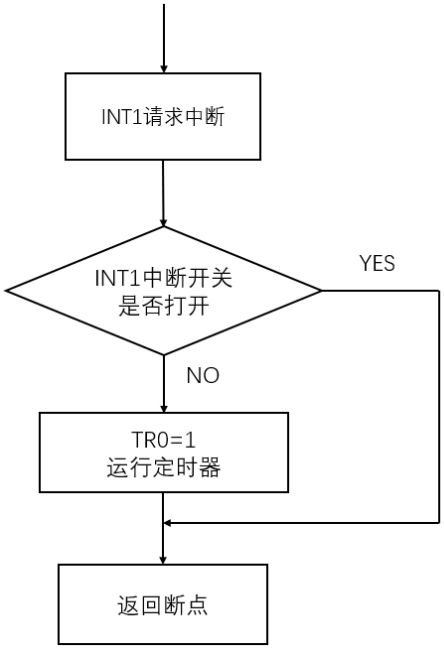
4.1.1 主函数框图



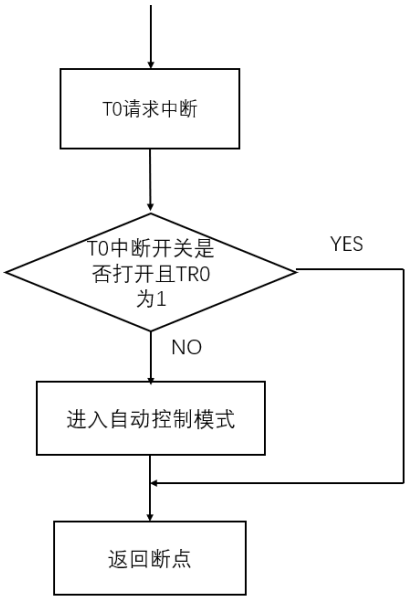
4.1.2 INT0 外部中断框图



3.1.3 INT1 外部中断框图



3.1.4 T0 定时器中断框图



4.2 软件的模块化实现

描述主要功能的实现流程和方法，关键代码的描述

4.2.1 头文件

C 语言作为高级语言，其很突出的一点优势就是能包含库，由于这些库在程序头几行被声明，所以一般也叫头文件。对于 51 单片机的 C 语言编程，一个很重要的头文件就是<reg51.h>。

```
#include <reg51.h>
#define uint unsigned int
#define uchar unsigned char                    //头文件定义
```

查阅资料，发现这个头文件对 51 单片机的诸多引脚、寄存器空间都进行了定义，并且分配了相应的存储空间空间。文件中对 P0~P3 口的定义如下所示：

sfr P0 = 0x80; sfr P1 = 0x90; sfr P2 = 0xA0; sfr P3 = 0xB0;

所以在编程是可以直接使用 P0~P3 来表示端口数值。但该头文件中并不包含对这些端口每个引脚的定义，所以我们要想直接用引脚的代号来表示引脚，则必须先使用 sbit 进行相关定义。

```
sbit GREEN_H = P0^0;                    // 定义水平方向horizontal对应的引脚
sbit YELLOW_H = P0^1;
sbit RED_H = P0^2;

sbit YELLOW_V = P0^3;                   // 定义竖直方向vertical对应的引脚
sbit GREEN_V = P0^4;
sbit RED_V = P0^5;
```

4.2.2 变量定义

由于交通灯的控制要求中，需要在手动模式下，通过按键来切换单行方向（即要在两种红绿灯模式下切换），所以要定义手动模式下的单行道控制变量。在程序中通过查询的方式对其进行读取。

```
sbit nbkey=P1^0;                    //定义手动模式下的单行道控制变量
sbit dxkey=P1^1;
```

在交通灯的自动控制模式中，会有四种亮灯组合轮流运行，因此要定义一个标志位，使其在 0-3 四个数字中循环，然后对应的四个亮灯组合也进行循环。

```
uchar Operation_Type = 1;           //定义自动模式下的模式切换变量
```

4.2.3 延时子程序

因为交通灯在自动控制模式下需要循环显示，每个通行状态延时 5s，每个等待状态延时 2s。因为该程序所用的延时时间固定，所以定义一个延时子程序，方便直接调用。

```
void Delays(unsigned int s)         //定义延时子程序
{
    unsigned int i,j,k;
    for(i=0;i<ms;i++)
        for(j=0;j<15000;j++)
        for (k=0;k<9;k++);
}
```

如上所示，这个程序能够以 1ms 位单位延时。如果要延时 5s 或 2s，只需在调用该子函数时，使得局部变量分别为 5s 或 2s 即可。

4.2.4 手动控制模式

由于题目需要在手动控制模式下，对交通灯允许的通向进行切换。因此这里采用查询的方法进行。在 main 主函数中，会进入 while（1）的死循环，然后在该循环中就不断地对 nbkey 和 dxkey 两个标志变量进行查询。当 nbkey 按下时，其值变成 0，使得竖直方向通车，水平方向禁行；当 dxkey 按下时，其值变成 0，使得水平方向通车，竖直方向禁行。

```
   while(1)                                  //手动切换模块，用查询进行方式切换
 if(nbkey==0)                                    //按键nbkey按下，竖直方向通车，水平方向禁行
        { TR0=0;
        YELLOW_H=1;
        YELLOW_V=1;
        GREEN_H=1;
        GREEN_V=0;
        RED_H=0;
        RED_V=1;
        }

 if(dxkey==0)                                    //按键dxkey按下，水平方向通车，竖直方向禁行

        { TR0=0;
        YELLOW_H=1;
        YELLOW_V=1;
        GREEN_H=0;
        GREEN_V=1;
        RED_H=1;
        RED_V=0;
        }
```

4.2.5 自动-手动模式转换

自动-手动模式间的转换通过外部中断来实现。对应两个模式切换的按键连接单片机的 INT0 和 INT1 脚。INT1 按下使得定时计数器 T0 的运行位置 1，定时计数器开始工作，单片机进入自动切换模式。INT0 按下使得定时计数器 T0 的运行位置 0，定时器停止工作，单片机进入手动控制模式。

```
void External_Interrupt_0() interrupt 0     //自动-手动模式转换，通过中断方式进行
     {TR0=0;

      pause=1;
     }

void External_Interrupt_1() interrupt 2
{  TR0=1;
    pause=0;
      }
```

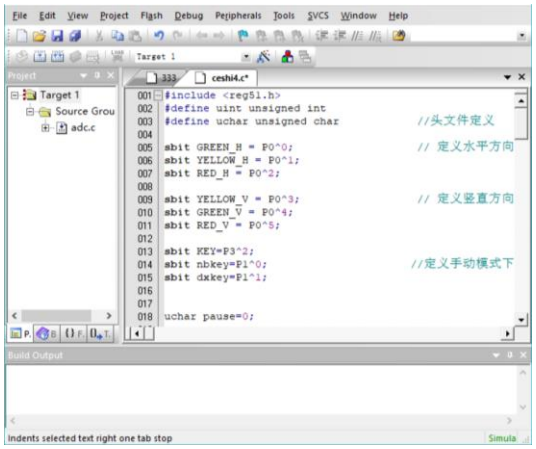
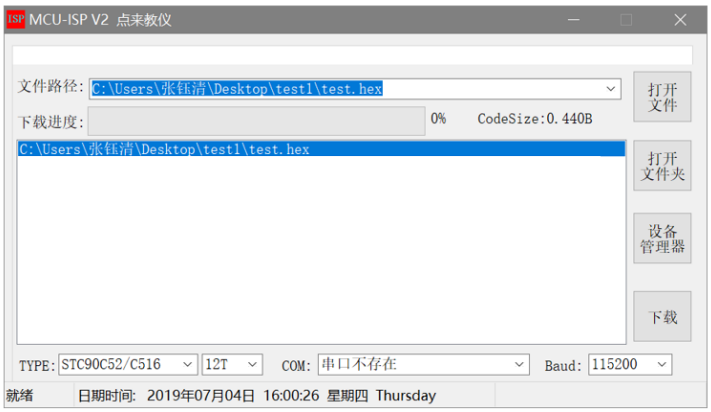
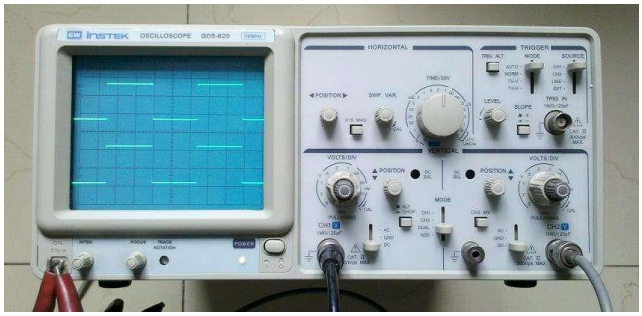
4.2.6 自动控制模式

在交通灯的自动控制模式中，两组交通灯需要在 4 个亮灯组合中切换。且在通行-禁止状态持续 5s，在等待-通行状态持续 2s。因此需要对 Operation Type 进行分支函数的编写。每进入一个 Operation Type 时，该标志位指向下一个状态，使得能在这四个状态间循环。

```
        void T0_INT() interrupt 1                  //自动切换模式
        {    if(pause==1) return;
            switch(Operation_Type)
            {   case 1:                                  //水平方向通行，竖直方向禁行，持续5秒
                    RED_H=1;YELLOW_H=1;GREEN_H=0;
                    RED_V=0;YELLOW_V=1;GREEN_V=1;
                    Delays(5);
                    Operation_Type=2;
                        return;
                case 2:                                  //水平方向黄灯等待，竖直方向禁行，持续2秒
                    RED_H=1;YELLOW_H=0;GREEN_H=1;
                    RED_V=0;YELLOW_V=1;GREEN_V=1;
                    Delays(2);
                    Operation_Type=3;
                        return;
                case 3:                                  //水平方向禁行，竖直方向通行，持续5秒
                    RED_H=0;YELLOW_H=1;GREEN_H=1;
                    RED_V=1;YELLOW_V=1;GREEN_V=0;
                    Delays(5);
                    Operation_Type=4;
                         return;
                case 4:                                  //水平方向禁行，竖直方向黄灯等待，持续2秒
                    RED_H=0;YELLOW_H=1;GREEN_H=1;
                    RED_V=1;YELLOW_V=0;GREEN_V=1;
                    Delays(2);
                    Operation_Type=1;
                         return;
```

五、 系统测试

5.1 测试工具和软件

Keil 中对 C 语言源代码进行调试	ISP 中用 HEX 文件对单片机进行烧录
	
用万用表检测焊点是否有虚焊、断点	用示波器检查各关键点的波形传递情况
	

5.2 测试方法

1.Keil 中对 C 语言源代码进行调试；
2.ISP 中用 HEX 文件对单片机进行烧录；
3.用万用表检测焊点是否有虚焊、断点；
4.用示波器检查各关键点的波形传递情况。

六、 课程设计总结

此次微机原理课程设计其实对于我本人是一次不大不小的挑战，课程设计的整个过程涉及了选型、画原理图、画 PCB 图等硬件设计的过程，也有在 Keil 中编写程序、以及在 SPI 中烧录的软件开发的过程。还有手工焊接电路板的动手操作环节，使得我们有机会把以前学的知识都使用了一遍。

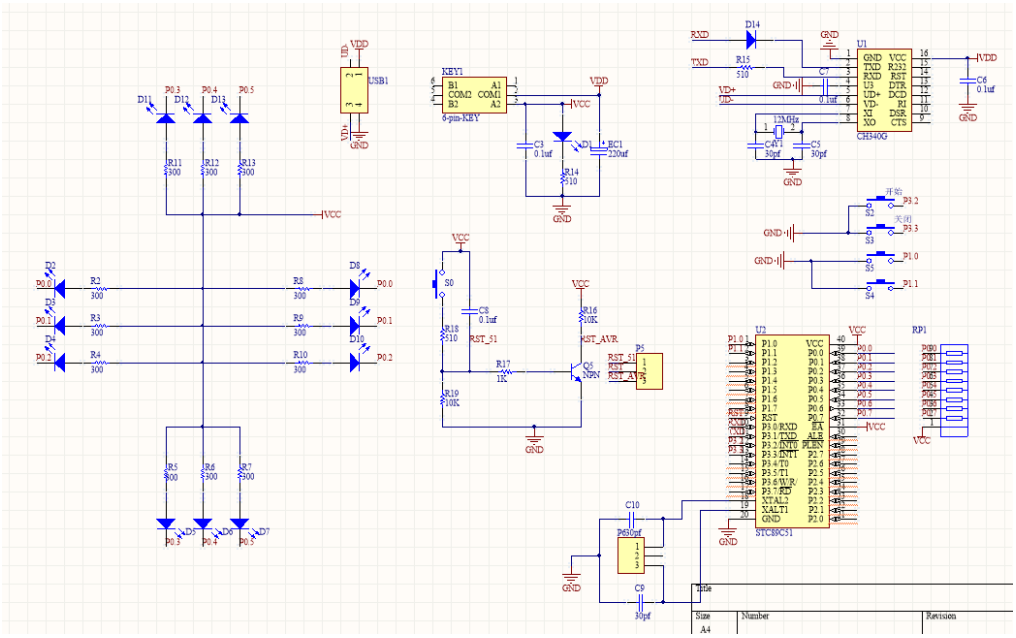
这次的课程设计中最重要的一点是不断的学习、试错、查找别人优秀的解决方案、模仿、学习后加以利用的过程。我特别感受到，其实很多问题早就都有了前人去试错，也都有了成熟的结局方案，重要的是能不能找到这些优秀的经验，如果找到了就是事半功倍，比自己闭门造车半天要好很多。

在程序编写的过程中，我进一步熟悉了查询和中断两种外部信息读取的方法。通过查询方式，我得以在手动控制状态下切换交通灯的方向。一般来说，中断是较大范围的外界信息扫描和读取，而查询是小范围的外界信息扫描和读取。查询方式常常是嵌套在中断方式里面的。

虽然这不是我第一次画板子、第一次做电路设计，但是还是遇到了很多问题，比如很多库都是软件中没有的，需要自己建立库，而且还有很多画板子时和元器件的封装要对应，比如我对于几个开关的通断引脚在设计时候就弄错了，导致最后焊接电路板的时候需要把开关焊在下面，微动开关甚至要转 90°才能焊，回想起来李老师提醒我的开关也是有方向的……非常汗颜，还是栽倒在这上面以至于焊了两次。这些宝贵的经验都被我记在自己的手册上了，作为日后更多设计工作的经验之谈。总之这次微机原理课程设计还是非常值得能够学到相当多东西的。

附件：

附件 1 硬件系统原理图（图片）



附件 2 程序及注释

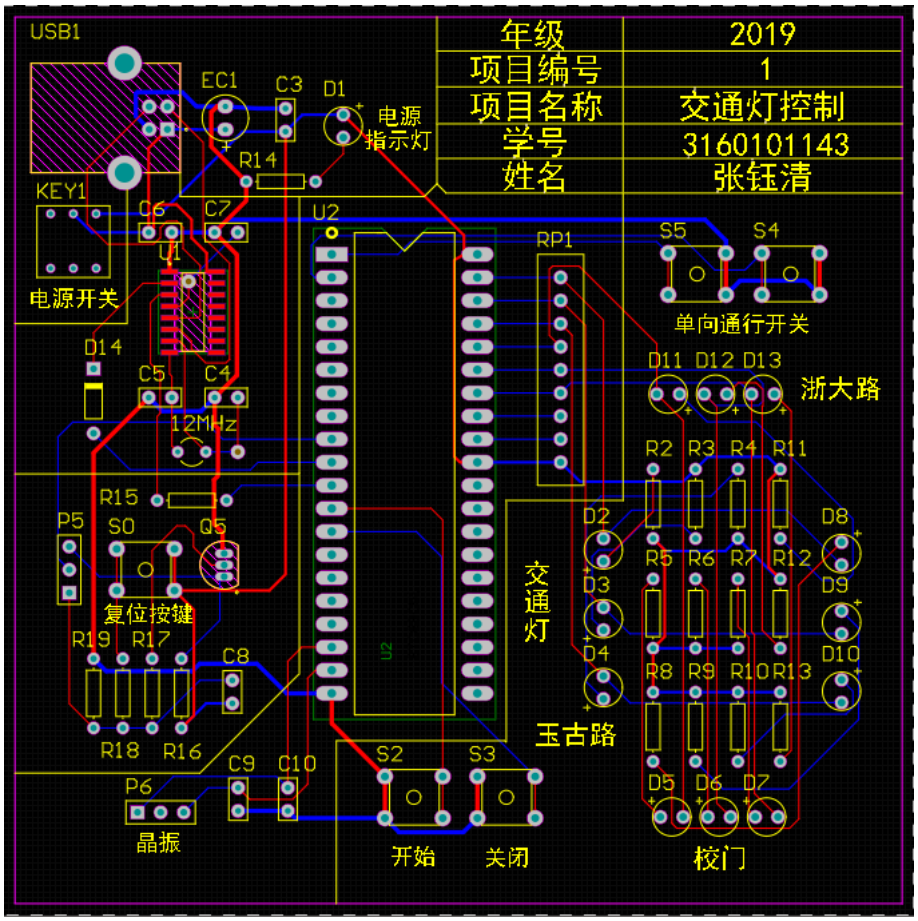
```
#include <reg51.h>
#define uint unsigned int
#define uchar unsigned char                    //头文件定义

sbit GREEN_H = P0^0;                           // 定义水平方向 horizontal 对应的引脚
sbit YELLOW_H = P0^1;
sbit RED_H = P0^2;

sbit YELLOW_V = P0^3;                          // 定义竖直方向 vertical 对应的引脚
sbit GREEN_V = P0^4;
sbit RED_V = P0^5;

sbit KEY=P3^2;
sbit nbkey=P1^0;                               //定义手动模式下的单行道控制变量
sbit dxkey=P1^1;
```

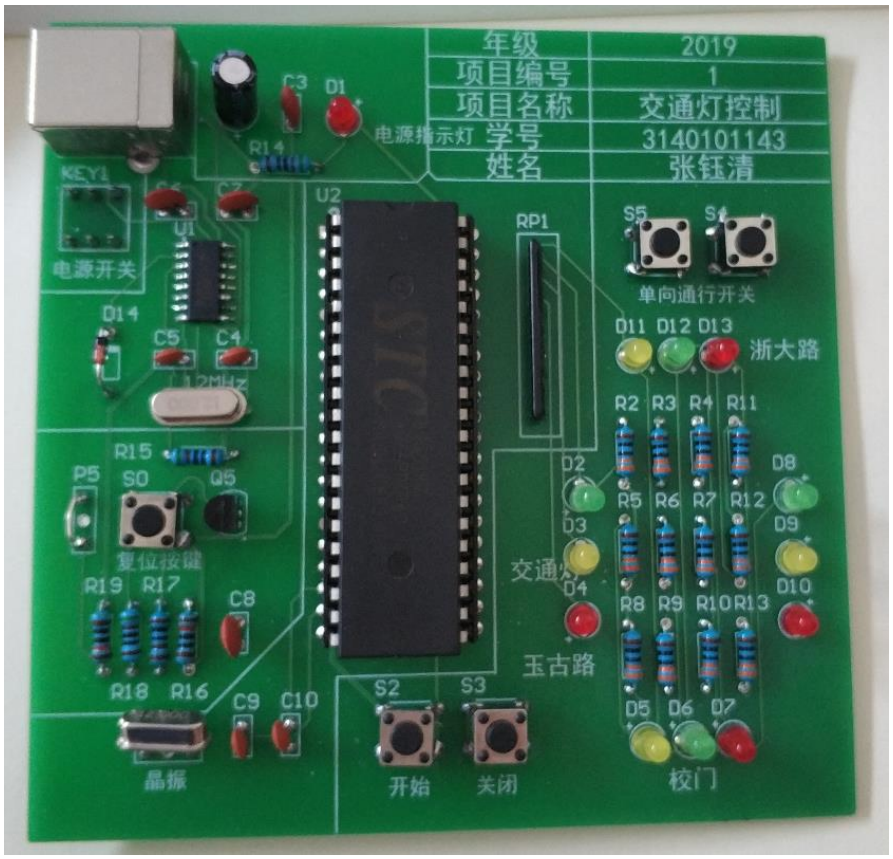
```
uchar pause=0;

uchar Operation_Type = 1;                    //定义自动模式下的模式切换变量
uchar Time_Count=0;

 void Delays(unsigned int s)        //定义延时子程序
{
      unsigned int i,j,k;
      for(i=0;i<ms;i++)
            for(j=0;j<15000;j++)
            for (k=0;k<9;k++);
}

 void main()
{           TH0=h;                                    //计时器及外部中断初始化定义
      TL0=l ;

      IP=    0X04;
      IE = 0x87;
      TMOD = 0x01;
      TCON = 0x11;
            nbkey=1;
            dxkey=1;
TH0=h;
TL0=l;


      while(1)                                  //手动切换模块，用查询进行方式切换
{ if(nbkey==0)                                        //按键 nbkey 按下，竖直方向通车，水平
方向禁行
            { TR0=0;
            YELLOW_H=1;
            YELLOW_V=1;
            GREEN_H=1;
            GREEN_V=0;
            RED_H=0;
            RED_V=1;
            }

 if(dxkey==0)                                          //按键 dxkey 按下，水平方向通车，
竖直方向禁行

            { TR0=0;
            YELLOW_H=1;
```

```
            YELLOW_V=1;
            GREEN_H=0;
            GREEN_V=1;
            RED_H=1;
            RED_V=0;
            }


}
            }

void External_Interrupt_0() interrupt 0       //自动-手动模式转换，通过中断方式进行
        {TR0=0;

          pause=1;
        }

void External_Interrupt_1() interrupt 2
{    TR0=1;
      pause=0;
         }

void T0_INT() interrupt 1                          //自动切换模式
{
        if(pause==1) return;
      switch(Operation_Type)
      {
           case 1:                                          //水平方向通行，竖直方向禁行，持续 5
秒
                RED_H=1;YELLOW_H=1;GREEN_H=0;
                RED_V=0;YELLOW_V=1;GREEN_V=1;
                Delays(5);
                Operation_Type=2;
                     return;

           case 2:                                          //水平方向黄灯等待，竖直方向禁行，
持续 2 秒
                RED_H=1;YELLOW_H=0;GREEN_H=1;
                RED_V=0;YELLOW_V=1;GREEN_V=1;
                Delays(2);
                Operation_Type=3;
             return;
```

```
            case 3:                                    //水平方向禁行，竖直方向通行，持续
5 秒
                RED_H=0;YELLOW_H=1;GREEN_H=1;
                RED_V=1;YELLOW_V=1;GREEN_V=0;
                Delays(5);
                Operation_Type=4;
                    return;

            case 4:                                    //水平方向禁行，竖直方向黄灯等待，
持续 2 秒
                RED_H=0;YELLOW_H=1;GREEN_H=1;
                RED_V=1;YELLOW_V=0;GREEN_V=1;
                Delays(2);
                Operation_Type=1;
                    return;

    }

}
```

附件 3 PCB 图（图片）




年级 2019
项目编号 1
项目名称 交通灯控制
学号 3140101143
姓名 张钰清
C3
D1
电源指示灯
R14
KEY1
电源开关
C2
U1
U2
RP1
S5
S4
单向通行开关
D14
C5
C4
12MHz
D11 D12 D13
浙大路
R2 R3 R4 R11
R15
P5
S0
Q5
复位按键
D2
R5 R6 R7 R12
D8
D3
交通灯
D9
R19 R17
C8
D4
R8 R9 R10 R13
D10
玉古路
R18 R16
C9 C10
晶振
S2
S3
开始
关闭
D5 D6 D7
校门