Lab Report

School of Manufacturing Science and Engineering

Course Name	**Microcontroller: Principles and Interfacing Technology**						
Name of the Experiment	**Traffic Lights**						
Name	尹思凯 **Yin Sikai**	Student No.	**2014141441168**	Department	**Wu Yuzhang Honors College**	Class No.	**NO.1**
Date	**2016.11.30**			Location	**C204, Basic Teaching Building**		
Teacher	**Pro. Yu Deping**			Grade			

1. Problem description and analysis

The main objective of this experiment is to write program simulate the control of traffic light. Based on the resources of the main board, we are required to write an assembly program to simulate the traffic light (Green/Red light: 30s; Yellow light: 5s) and debug it in Keil μvision. In fact, each LED light of the 10 traffic lights has already been connected to one pin respectively. As a result, we can control the light logic through sending rights control signals to these pins.

2. Experimental procedure

2.1 Hardware connection

As mentioned above, the hardware connection between port pins and LED lights has already been accomplished so we just need to use appropriate wires to link pins with the microcontroller pins. The Proteus graph below illustrates the hardware design of this LED traffic light system based on a simulated crossway.

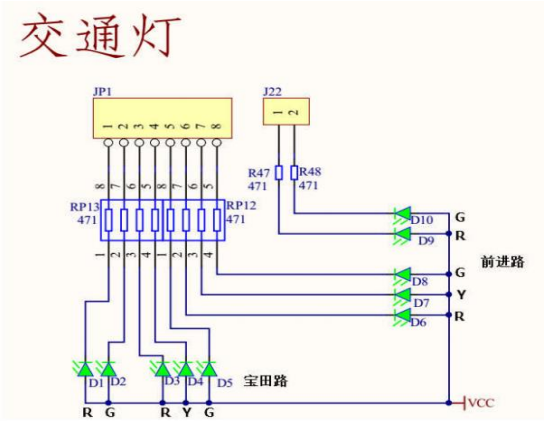

Fig. 1 Hardware design and connection

2.2 Wire connection

Based on the task requirements, we can just utilize ten pins to control the ten LED traffic lights sequentially. As a result, only 10 wires are need. The 8 pins of Port2 are utilized, connecting to eight control pins. The other two lights are controlled by Pin3.0 and Pin3.1 respectively. Consequently, we shall connect JP11(P2) with JP1, connect P3.0 and P3.1 with JP22. The final connection effect is illustrated as below:

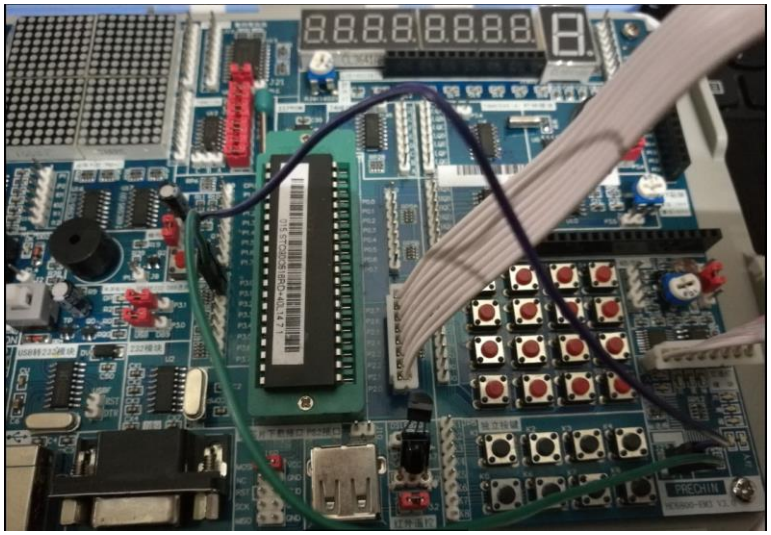

Fig. 2 Wire connection

2.3 Write assembly program to simulate the traffic light

In the next step, we write the assembly program based on Keil uvision platform to simulate the operation conditions of the crossroads based on the knowledge learnt about interrupt, timer/counter. Here we set the timer 0 to record 5000us each time. The address 30H is designated to count until 20 rolls(1s). For each second recorded, the number stored in address 31H would be increased by 1. Then when the time reaches certain boundaries that needs operation, which are 30s, 35s(30+5), 65s(35+30) and 70s, the traffic lights would all be switched off first, then implement the control signal by the microcontroller to light desired ones. After completing the whole circle, the program would return to its beginning. The code programmed is listed in the following section.

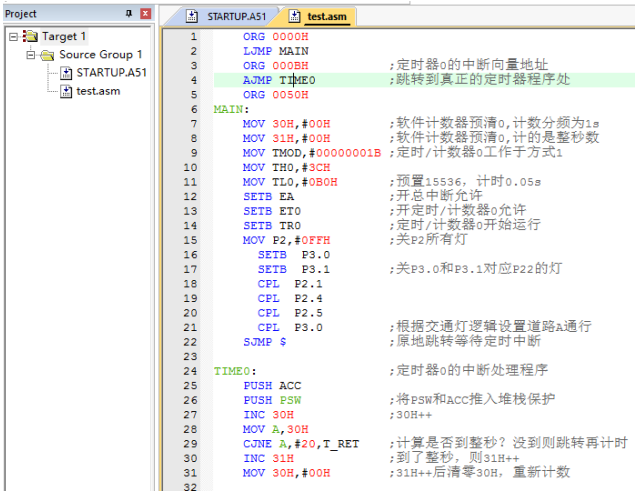

Fig. 3 Programming the desired functions

2.4 Debug and Test the Program Designed

Finally we compiled the code in Keil and debugged it. As this program requires the function of interrupt and timer/counter, it cannot be realized simply by clicking step running button. After setting the timer/counter and interrupt in Keil debug session window, we can test the result of our program by watching the output of Port 2 and Port 3 simultaneously.

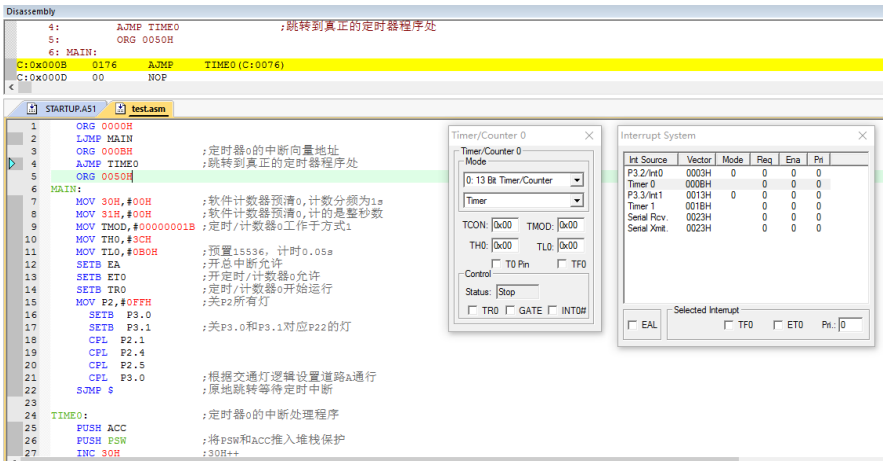

Fig. 4 Testing and Debugging Process

3. Experimental results

The final results of this experiment meet the problem requirement to simulate the traffic lights of a crossroad. The green/red lights last for 30 seconds and are followed by 5-second interval with yellow lights. Altogether, there are four lighting states, which are illustrated below:

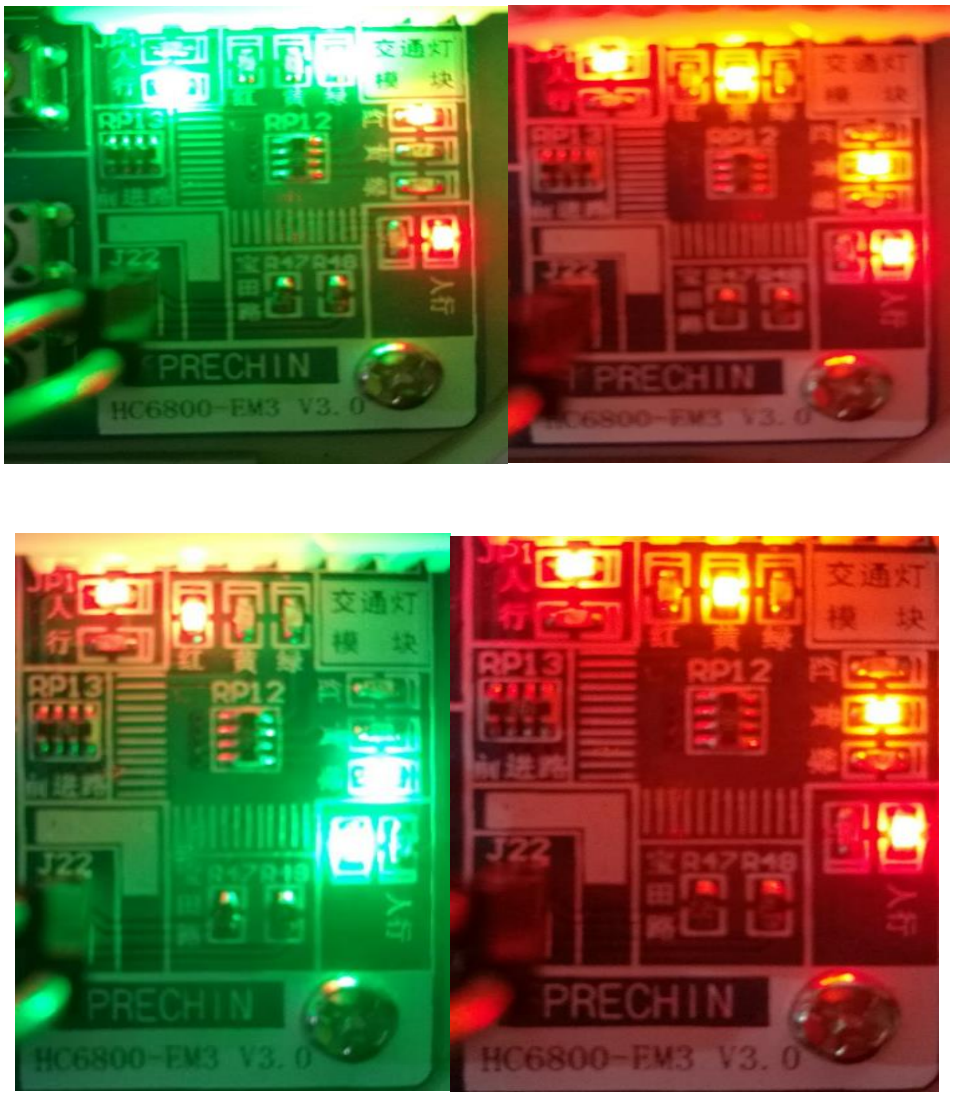

Fig. 5 Test Results for four states of traffic lights

4. Source code with comments(Assembly language and Chinese comments)

```
        ORG 0000H
        LJMP MAIN
        ORG 000BH                ;定时器 0 的中断向量地址
        AJMP TIME0               ;跳转到真正的定时器程序处
        ORG 0050H
MAIN:
        MOV 30H,#00H        ;软件计数器预清 0,计数分频为 1s
        MOV 31H,#00H        ;软件计数器预清 0,计的是整秒数
        MOV TMOD,#00000001B ;定时/计数器 0 工作于方式 1
        MOV TH0,#3CH
```

```
        MOV TL0,#0B0H       ;预置 15536，计时 0.05s
        SETB EA             ;开总中断允许
        SETB ET0            ;开定时/计数器 0 允许
        SETB TR0            ;定时/计数器 0 开始运行
        MOV P2,#0FFH        ;关 P2 所有灯
          SETB  P3.0
          SETB  P3.1        ;关 P3.0 和 P3.1 对应 P22 的灯
          CPL   P2.1
          CPL   P2.4
          CPL   P2.5
          CPL   P3.0                ;根据交通灯逻辑设置道路 A 通行
       SJMP $                       ;原地跳转等待定时中断

TIME0:                              ;定时器 0 的中断处理程序
        PUSH ACC
        PUSH PSW                    ;将 PSW 和 ACC 推入堆栈保护
        INC 30H                     ;30H++
        MOV A,30H
       CJNE A,#20,T_RET             ;计算是否到整秒？没到则跳转再计时
        INC 31H                     ;到了整秒，则 31H++
        MOV 30H,#00H                ;31H++后清零 30H，重新计数

        MOV A,31H
        CJNE A,#30,T_NEXT32 ;31H 单元中的值到了 30S 了吗？没到，跳转到 T_NEXT32 判断下一个数字计数器的值
       MOV P2,#0FFH
          SETB  P3.0
          SETB  P3.1                ;关所有灯
          CPL   P2.3
          CPL   P2.0
          CPL   P2.6
          CPL   P3.0                ;黄灯阶段

T_NEXT32:
          CJNE A,#35,T_NEXT33;31H 单元中的值到了 35S 了吗？没到，跳转到 T_NEXT33 判断下一个数字计数器的值
          MOV P2,#0FFH
          SETB  P3.0
          SETB  P3.1                  ;关所有灯
          CPL   P2.2
          CPL   P2.0
          CPL   P2.7
          CPL   P3.1                ;根据交通灯逻辑设置道路 A 通行

T_NEXT33:
```

```
        CJNE A,#65,T_NEXT34;31H 单元中的值到了 65S 了吗？没到，跳转到 T_NEXT34 判断下一个数字计数器的值
        MOV P2,#0FFH
        SETB   P3.0
        SETB   P3.1                ;关所有灯
        CPL    P2.3
        CPL    P2.0
        CPL    P2.6
        CPL    P3.0                ;黄灯阶段

T_NEXT34:
        CJNE A,#70,T_RET           ;31H 单元中的值到了 70S 了吗？没到，跳转到 T_RET 重置定时
      LJMP MAIN                    ;到了 70s,跳回 MAIN 程序，实现循环
T_RET:
        MOV TH0,#3CH
        MOV TL0,#0B0H              ;重置定时常数
        POP PSW
        POP ACC
        RETI                       ;中断返回
        END
```

5. Conclusions, suggestions and comments on the experiments

Through this interesting lab about traffic lights, I got much more familiar with the 8051 microcontrollers. Through writing the basic assembly language codes about timer/counter interrupt, I successfully controlled the logic of these ten LED lights. The function of timer/counter provides us the programmer with more possibilities to fulfill our desires. The electronic modules can thus be controlled in real time sequence, which is really cool. The microcontroller is really significant in converting the program into the control signals to electronic devices and let them function as the designer wanted. The experiment greatly improved my programming ability and computer skills. More importantly, it greatly aroused my interest towards programming. I wish to accomplish more interesting and inspiring projects like this. As for the suggestions and comments on the experiments, I think we can prolong the time of experiment and achieve more mathematical functions.

That was all the knowledge and ideas that I obtained through today's experiment, actually I can not wait for the next one. I will do more active individual learning and interdisciplinary studies based on my major afterwards with the platform and I hope I can get more interesting results. Also, I want to express my gratitude to our tutor for your patient guidance on this experiment. Thank you very much!