**《单片机原理及应用》**

**实验报告**

实验题目：

专业班级： 物联网工程17-1

学生学号： 541707090128

学生姓名： 刘亚炜

指导教师： 王丽萍

成 绩：

1、实验目的

实验目的：

1. 掌握单片机C语言程序设计和调试方法；

2. 掌握Proteus硬件仿真与单片机开发板的使用。

任务：

1. 根据要求编写程序，并写出原理性注释；

2. 将检查程序运行的结果，分析一下是否正确；

3. 完成所建工程的仿真及调试；

4. 将程序下载至开发板验证实验结果。

2、实验内容及要求

设计8路节日彩灯控制器，要求在Protues中设计接口电路图，并编程实现节日彩灯闪烁方式的控制：

1. 通过P1.0到P1.2来控制3个按键。3个按键分别对应3种彩灯闪烁方式。

2. 彩灯闪烁方式分别为：从上往下的循环、从下往上以及从中间向两边再从两边到中间三种流水方式。

3、实验步骤

（1）软件实现：

实验步骤：

1. 运行Keil uVision4 开发环境，按照“模块化单片机教学实验平台配套的实验指导书1.2.3 节”介绍的方法建立程esimlab3. uvproj，CPU 为AT89S52，包含启动文件STARTUP.A51。
2. 输入源程序，在Keil uVision4 开发环境中，建立源程序esimlab3.c，将上述程序加入该程序文件。并将该文件加入工程esimlab3.uvproj。
3. 设置工程esimlab3. uvproj 属性，将其晶振频率设置为11.0592MHz，选择输出可执行文件（HEX 文件），仿真方式为“Use Simulator”。
4. 构造（Build）工程esimlab3.uvproj。如果输入有误进行修改，直至构造正确，生成可执行程序 esimlab3.hex 为止。

（2）Proteus仿真：

实验步骤：

1. 运行Proteus 开发环境，建立工程，新建原理图设计文件esimlab3.DSN。
2. 设置单片机属性，将其晶振频率设置为11.0592MHz，载入KEIL生成的文件esimlab3.hex。
3. 运行仿真并观察结果。

（3）开发板实验验证：

实验步骤：

1. 关掉实验箱电源。将CPU 板插接在JK1、JK2 上，注意CPU 板的插接方向。按照表1连接关系表将硬件连接好。

表1

|  |  |
| --- | --- |
| **杜邦线连接关系** | |
| CPU\_CORE\_52 | MAIN\_BOARD |
| P2.0-P2.7 | J48:LED1-LED8 |
| P1.0 | SKEY1 |
| P1.1 | SKEY2 |
| P1.2 | SKEY3 |

1. 将AT89S52 芯片插在CPU 板的CPU 插座上（注意不要插反）。母板上电。
2. 通过PROGISP 烧写软件，利用ISP 下载线（见“模块化单片机教学实验平台配套的实验指导书1.3 节”）把HEX （esimlab3.hex）文件下载到单片机中。
3. 下载成功后，通过按键，观察母板上的LED1~LED8。

4、实验小结

### Keil程序源代码

#include<reg52.h>

#include<intrins.h> //包含循环左、右移位函数的头文件（\_crol\_, \_cror\_）

sbit S1=P1^0;

sbit S2=P1^1;

sbit S3=P1^2;

unsigned char keyval;

void key\_scan(void); //按键扫描，取键值

void operation1(); //从上到下闪烁

void operation2(); //从下到上闪烁

void operation3(); //从中间到两边，再从两边到中间

void delay10ms(void); //按键扫描延时，防抖动

void initial();

void led\_delay();

void main(void)

{

keyval=0;

while(1)

{

key\_scan();

switch(keyval)

{

case 1:

operation1();

break;

case 2:

operation2();

break;

case 3:

operation3();

}

}

}

void key\_scan(void)

{

P1=0xff;

if((P1&0x0f)==0x0e)

{

delay10ms();

if(S1==0)

keyval=1;

}

else if((P1&0x0f)==0x0d){

delay10ms();

if(S2==0)

keyval=2;

}

else if((P1&0x0f)==0x0b){

delay10ms();

if(S3==0)

keyval=3;

}

}

void operation1()

{

P2 = 0xfe;

keyval=0;

while(1){

key\_scan();

if(keyval>0){

initial();

return;

}

P2 = \_crol\_(P2, 1);

led\_delay();

}

}

void operation2()

{

P2 = 0xfe;

keyval=0;

while(1){

key\_scan();

if(keyval>0){

initial();

return;

}

P2 = \_cror\_(P2, 1);

led\_delay();

}

}

void operation3()//两边到中间

{

P2 = 0xfe;

keyval=0;

while(1){

key\_scan();

if(keyval>0){

initial();

return;

}

led\_delay();

P2 = 0x7e;

key\_scan();

if(keyval>0){

initial();

return;

}

led\_delay();

P2 = 0xbd;

key\_scan();

if(keyval>0){

initial();

return;

}

led\_delay();

P2 = 0xdb;

key\_scan();

if(keyval>0){

initial();

return;

}

led\_delay();

P2 = 0xe7;

}

}

void led\_delay(void)

{

unsigned char i,j;

for(i=0;i<220;i++)

for(j=0;j<220;j++);

}

void delay10ms(void)

{

unsigned char i,j;

for(i=0;i<100;i++)

for(j=0;j<100;j++);

}

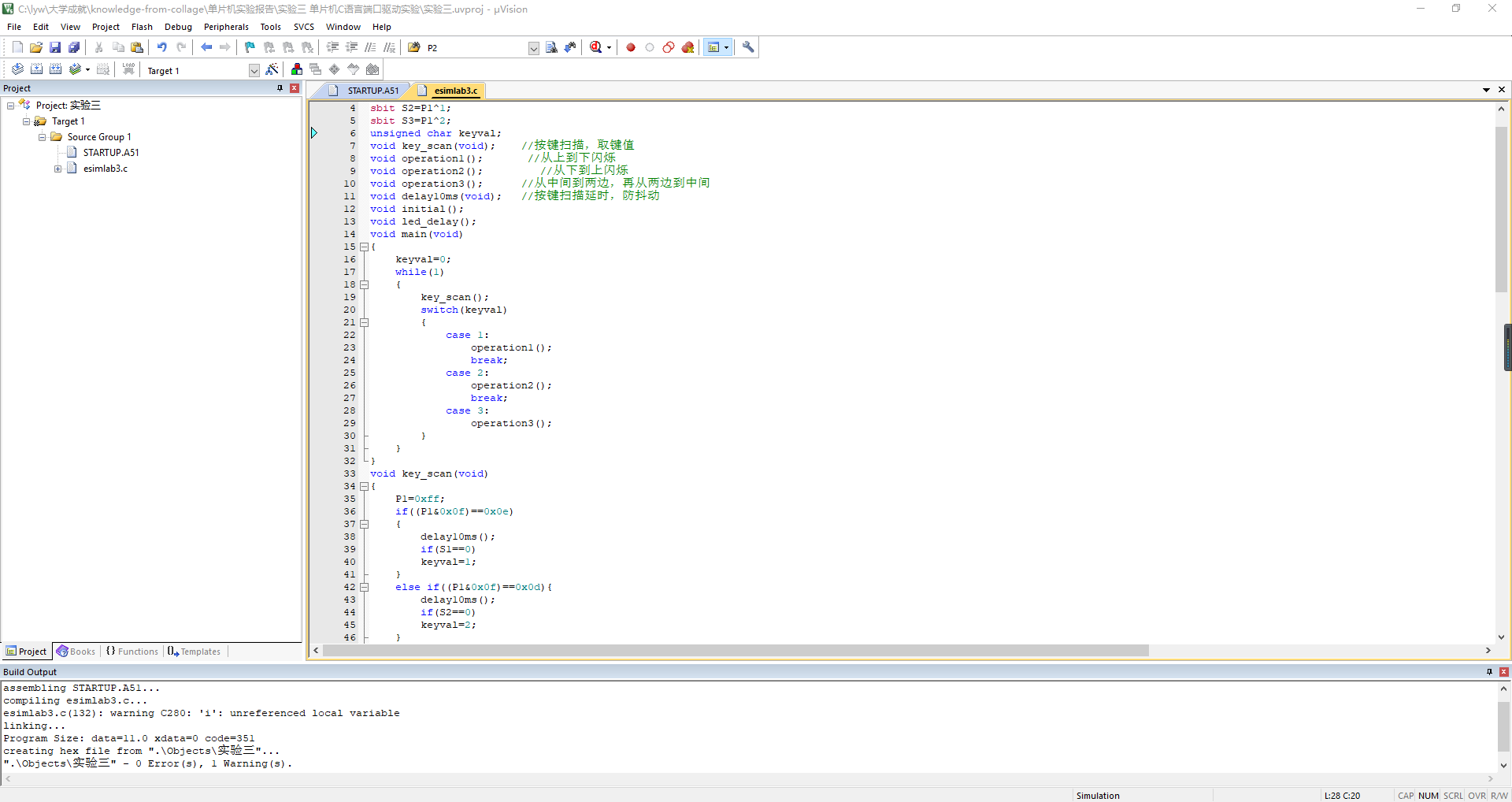
void initial(){

unsigned char i;

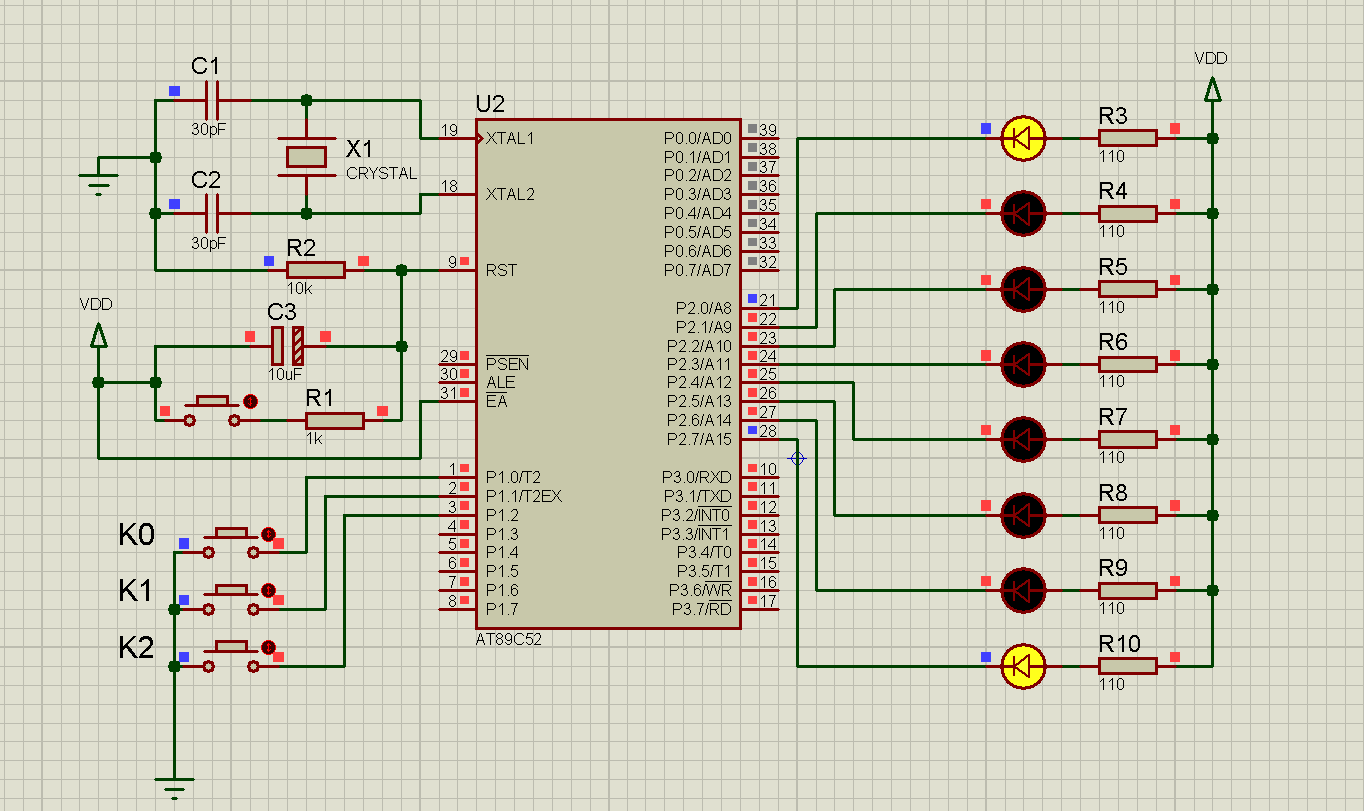
P2 = 0xff;

}

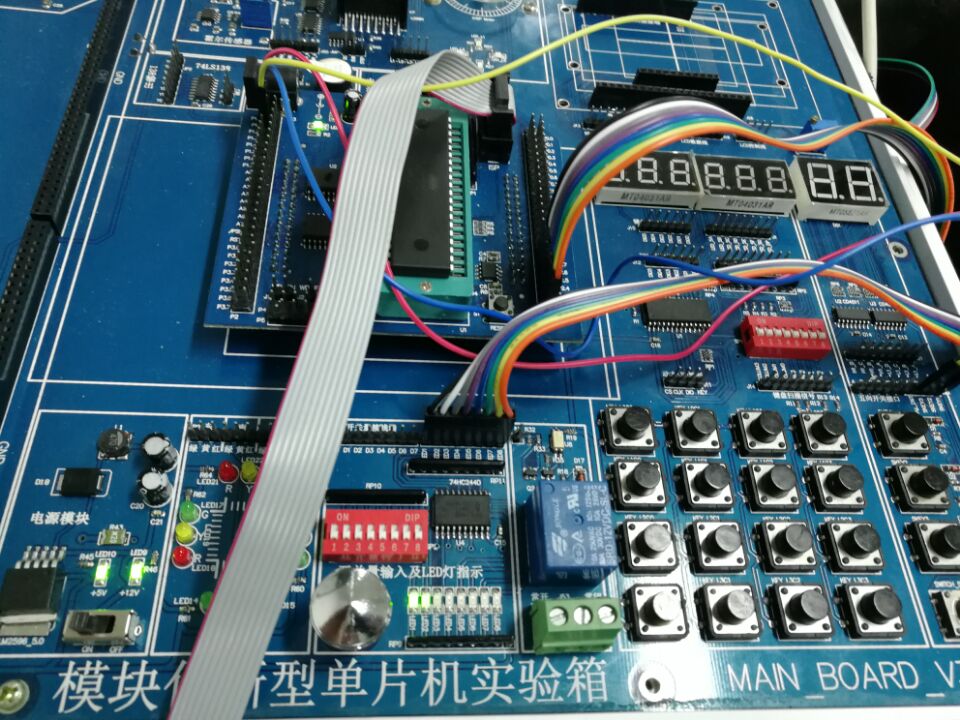
### Keil程序截图



### Proteus运行截图



### 单片机烧写程序截图



### 总结

本次实验的逻辑是逐一进行点亮并将上一个灯弄灭，有多种方法，既可以对P2口进行预置位的操作，又可以对P2口进行函数移位，也可以将每个引脚的地址组成数组进行使用，这种方法在使用过程中还可以使用三目运算符方便编写程序，最终生成hex文件，并将hex文件导入Proteus工程内，仿真正常，再将hex文件使用下载器下载到单片机中，单片机使用正常。