

**太原学院计算机科学与技术系**

**实验报告**

|  |  |
| --- | --- |
| 课程名称 | 单片机原理与应用 |
| 姓 名 |  |
| 学 号 |  |
| 专业班级 | 计科21-4班 |
| 指导教师 | 任凤琴 |

2023年09月

**目 录**

**[实验一：LED流水灯](#_Toc22732_WPSOffice_Level1)**

[一． 实验目的](#_Toc22732_WPSOffice_Level2) [3](#_Toc22732_WPSOffice_Level2)

[二． 实验设备](#_Toc7708_WPSOffice_Level2) [3](#_Toc7708_WPSOffice_Level2)

[三． 实验内容](#_Toc11783_WPSOffice_Level2) [3](#_Toc11783_WPSOffice_Level2)

[四． 实验步骤](#_Toc32164_WPSOffice_Level2) [4](#_Toc32164_WPSOffice_Level2)

[五． 实验结论及分析](#_Toc29664_WPSOffice_Level2) [6](#_Toc29664_WPSOffice_Level2)

**[实验二：LED数码管的静态显示](#_Toc11783_WPSOffice_Level1)** **[7](#_Toc11783_WPSOffice_Level1)**

**[实验三：LED数码管的动态显示](#_Toc25572_WPSOffice_Level1)** **[9](#_Toc25572_WPSOffice_Level1)**

**[实验四：键盘接口设计](#_Toc9711_WPSOffice_Level1)** **[12](#_Toc9711_WPSOffice_Level1)**

**[实验五：中断系统的应用](#_Toc1165_WPSOffice_Level1)** **[16](#_Toc1165_WPSOffice_Level1)**

**[实验六：定时/计数器的应用](#_Toc15037_WPSOffice_Level1)** **[20](#_Toc15037_WPSOffice_Level1)**

**[实验七：串行口的应用](#_Toc12977_WPSOffice_Level1)** **[23](#_Toc12977_WPSOffice_Level1)**

**[实验八：串行扩展应用](#_Toc22771_WPSOffice_Level1)** **[27](#_Toc22771_WPSOffice_Level1)**

计算机科学与技术系实验报告

课程名称：单片机原理与应用

班级：计科 21-4 姓名：赵泽辉 学号：4202150101443 成绩：

实验项目名称：LED流水灯

**一、实验目的**

通过对四个开关的操作，实现对八个led流水灯状态的控制。

**二、实验设备**

1.pc机器一台

2.keil编译软件与Proteus仿真软件

**三、实验内容**

通过对四个开关的操作，实现对八个led流水灯状态的控制。

功能一：上-下流水灯

功能二：下-上流水灯

功能三：循环闪动

功能四：四亮四灭循环

**四、实验步骤**

**（1）硬件图**

**Proteus仿真图：**

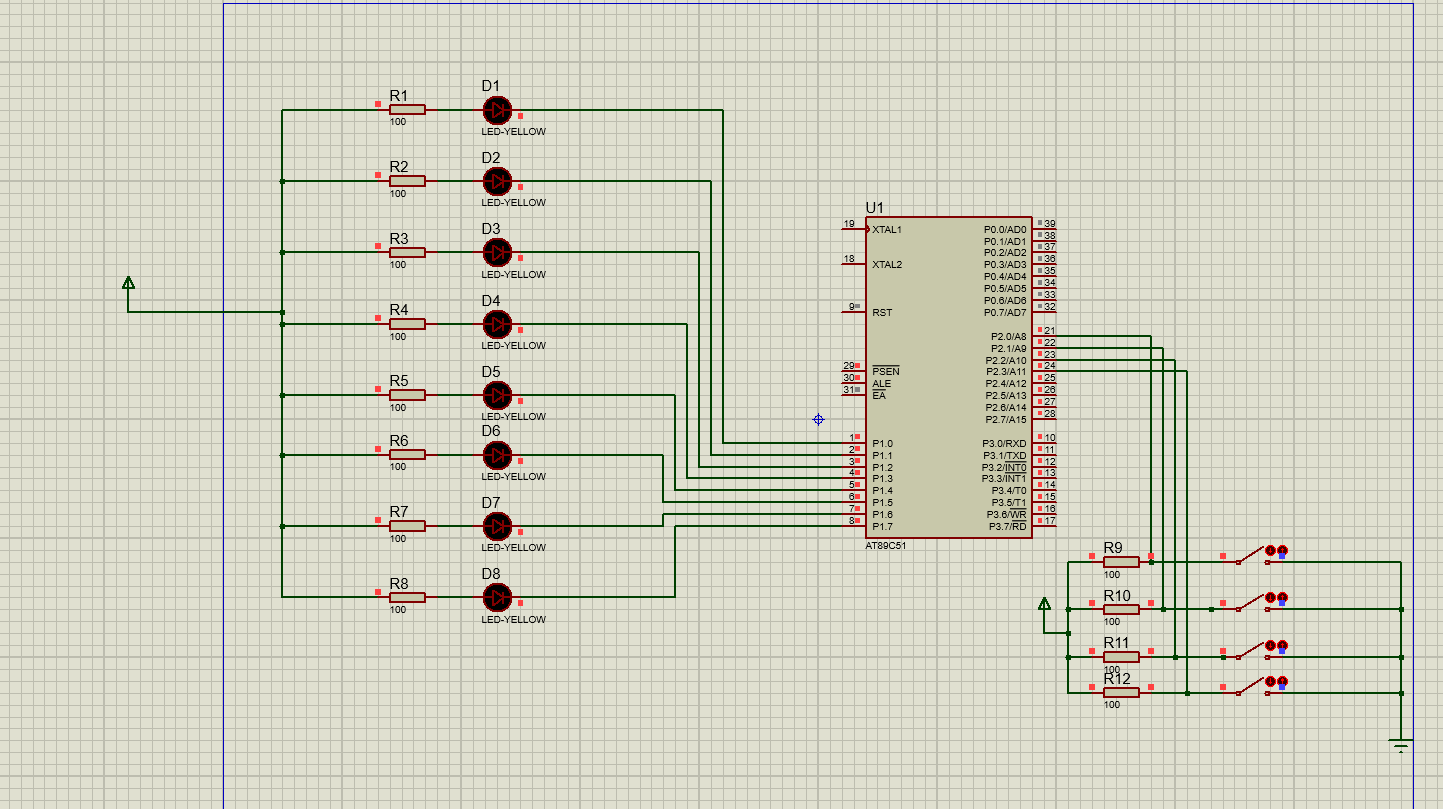


图1 实验仿真图

1. **软件设计**
2. #include <reg51.h>
3. #define uchar unsigned char
4. sbit P20 = P2^0 ;
5. sbit P21 = P2^1 ;
6. sbit P22 = P2^2 ;
7. sbit P23 = P2^3 ;
8. **void** delay()
9. {
10. uchar i,j;
11. **for**(i=0;i<100;i++)
12. **for**(j=0;j<150;j++);
13. }
15. **void** main ()
16. {
17. **if** (P20==0)     /\*\* 控制P2^0端口的开关实现上-下流水灯 \*/
18. {
19. **int** m,i=1;
20. P1 = 0xff;
21. **while**(i)
22. {
23. **for**(m = 0; m < 8; m++)
24. {
25. P1 = ~(1 << m);
26. delay();
27. **if**(P20==1)
28. {
29. P1=0xff;
30. i=0;
31. **break**;
32. }
33. }
34. }
35. }
37. **if** (P21==0)     /\*\* 控制P2^1端口的开关实现下-上流水灯  \*/
38. {
39. **int** m,i=1;
40. P1 = 0xff;
41. **while**(i)
42. {
43. **for**(m = 7; m >=0; m--)
44. {
45. P1 = ~(1 << m);
46. delay();
47. **if**(P21==1)
48. {
49. P1=0xff;
50. i=0;
51. **break**;
52. }
53. }
54. }
55. }
57. **if** (P22==0)         /\*\*  循环闪动 \*/
58. {   **int** i=1;
59. **while**(i)
60. {
61. P1=0x00;
62. delay();
63. P1=0xff;
64. delay();
65. **if**(P22==1)
66. {
67. P1=0xff;
68. i=0;
70. }
71. }
72. }
74. **if** (P23==0)         /\*\*  四亮四灭循环 \*/
75. {   **int** i=1;
76. **while**(i)
77. {
78. P1=0xf0;
79. delay();
80. P1=0x0f;
81. delay();
82. **if**(P23==1)
83. {
84. P1=0xff;
85. i=0;
87. }
88. }
89. }
90. }

**（3）效果图**

功能1演示：

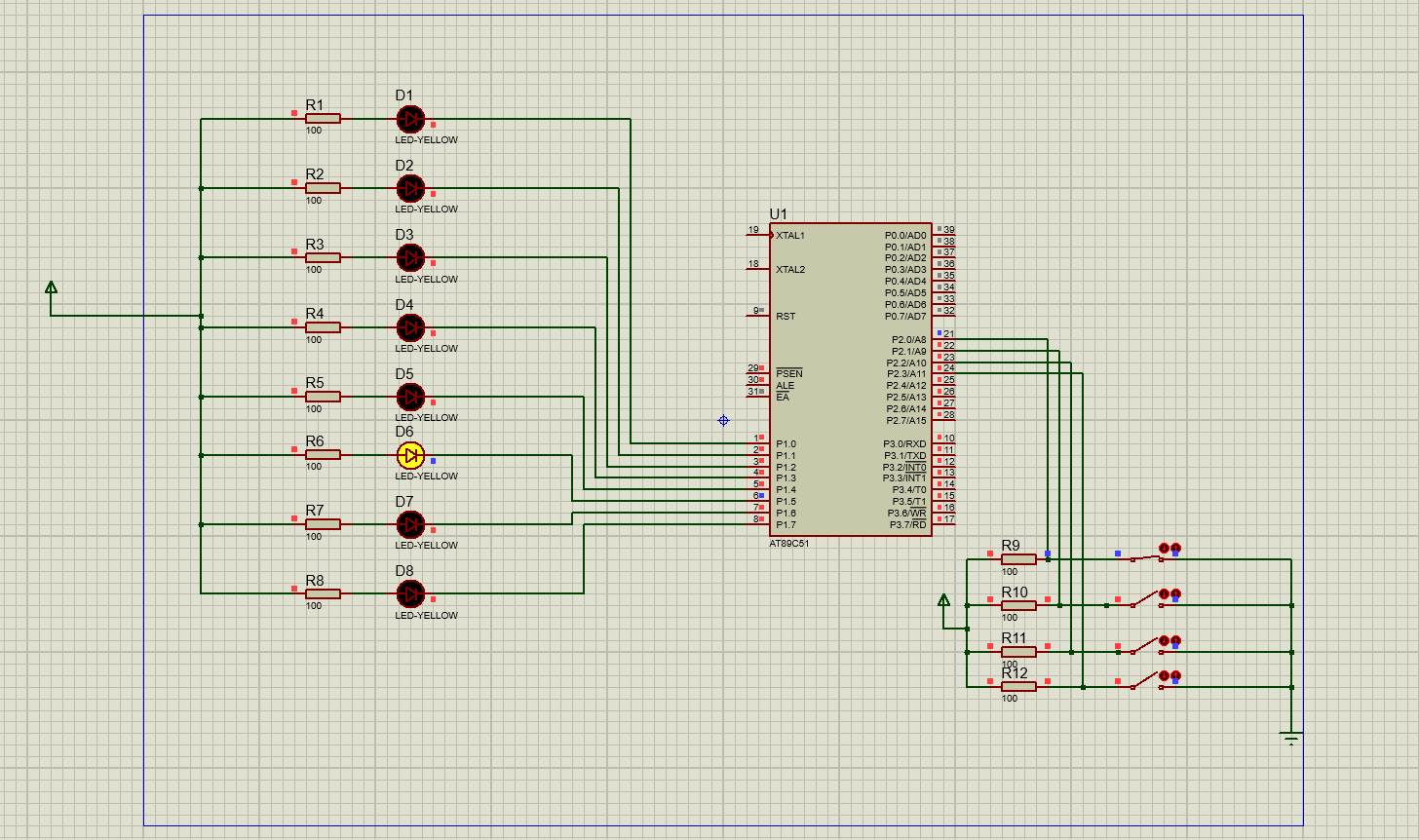


图2 功能演示图

功能2演示：

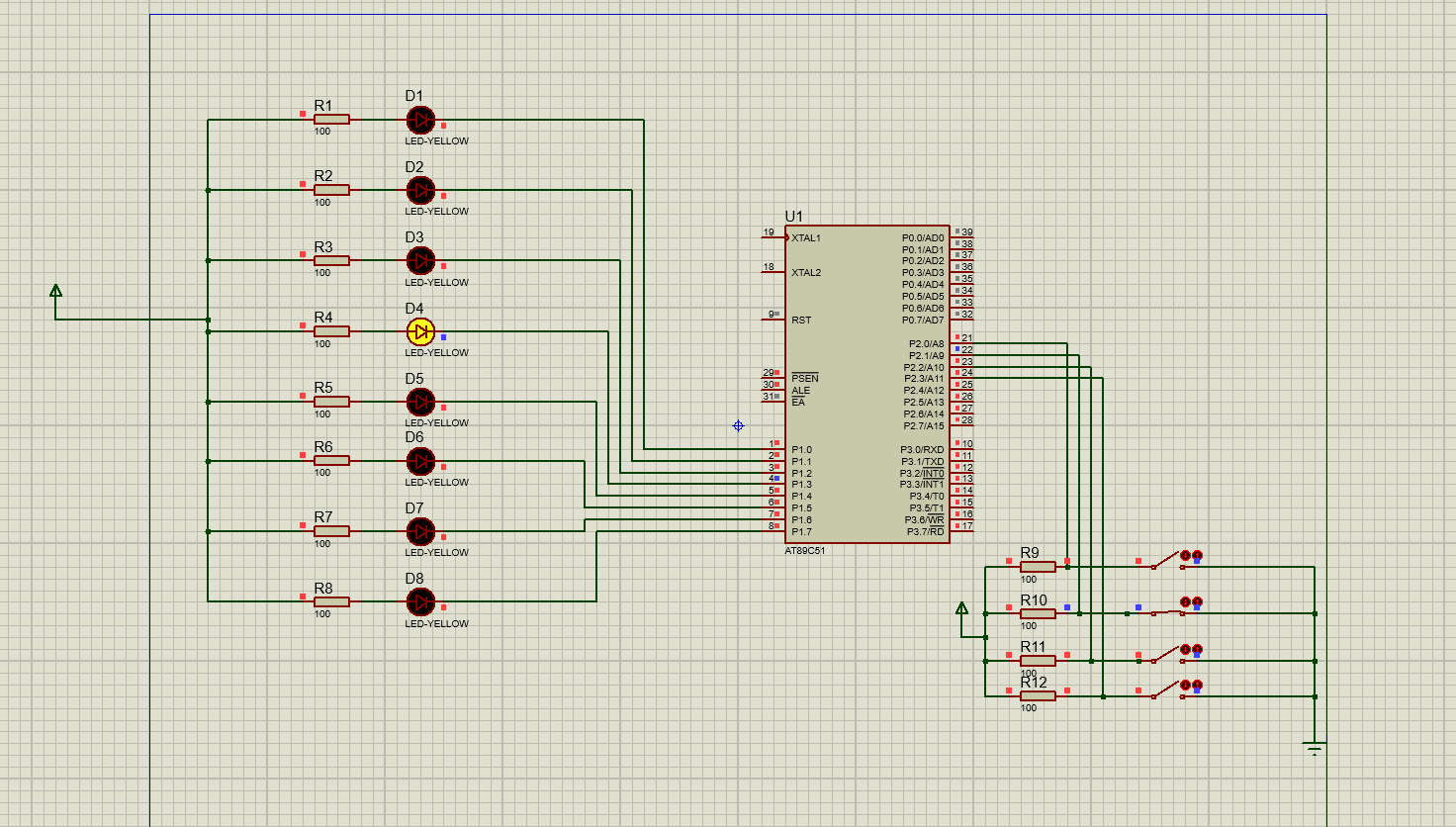


图3 功能演示图

功能3演示：

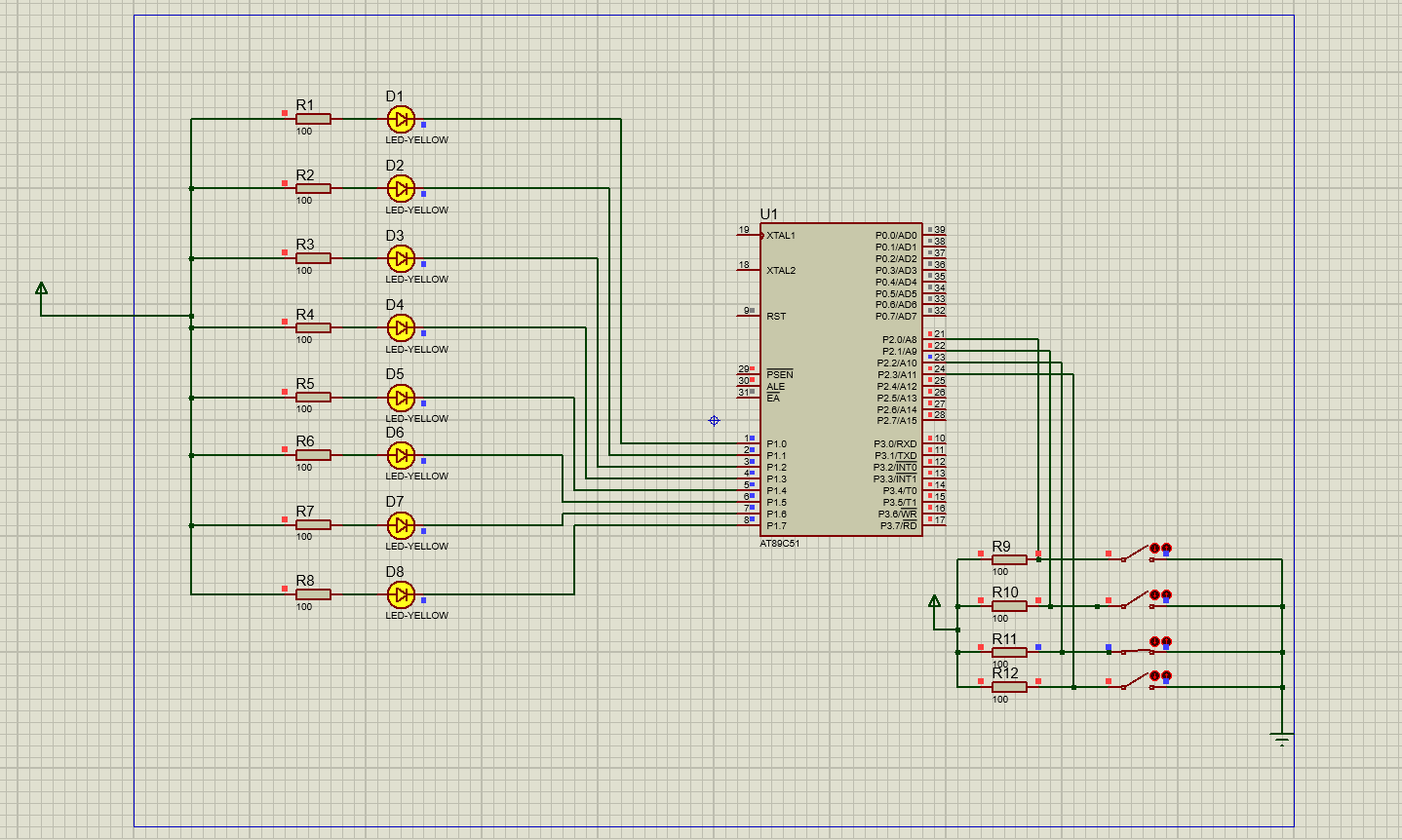


图4 功能演示图

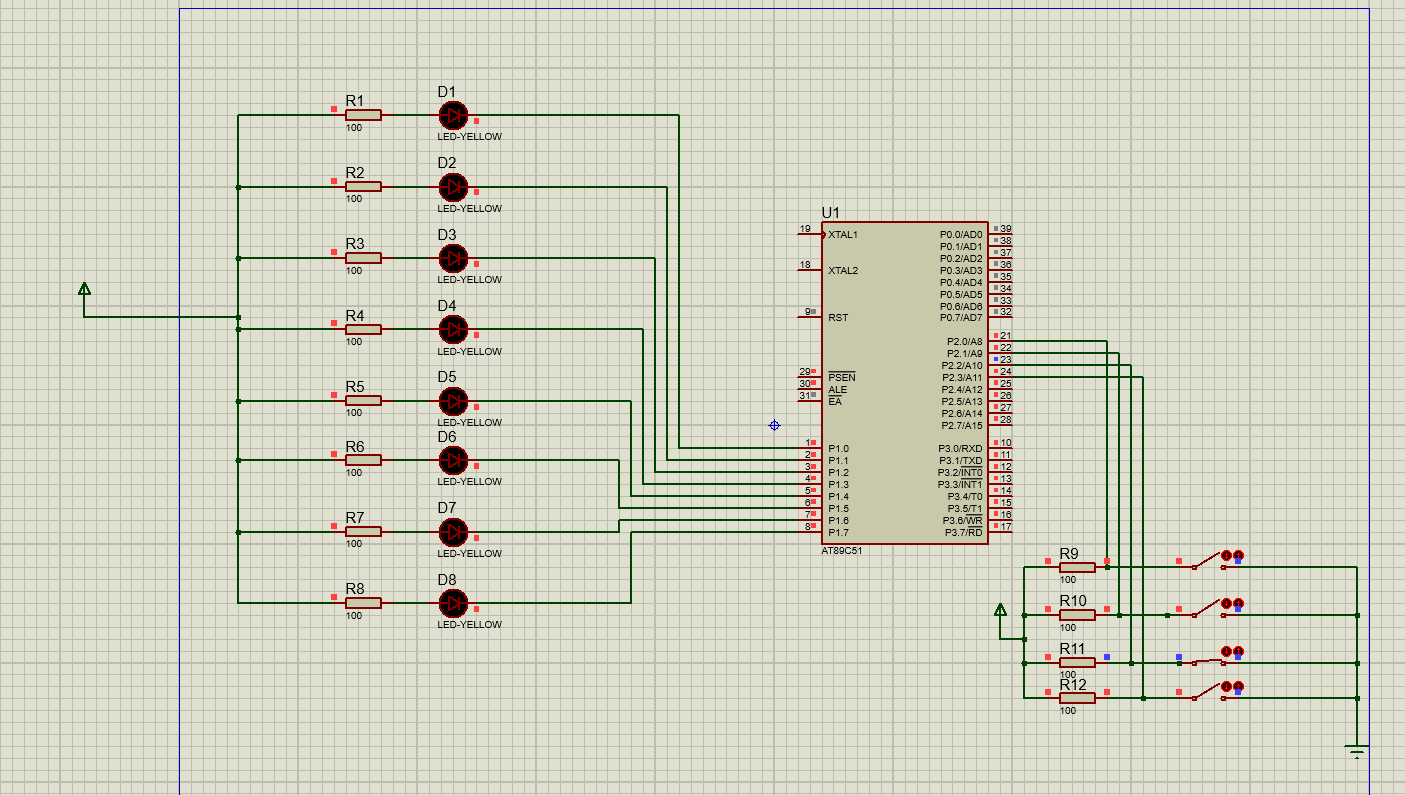


图5 功能演示图

功能4演示：

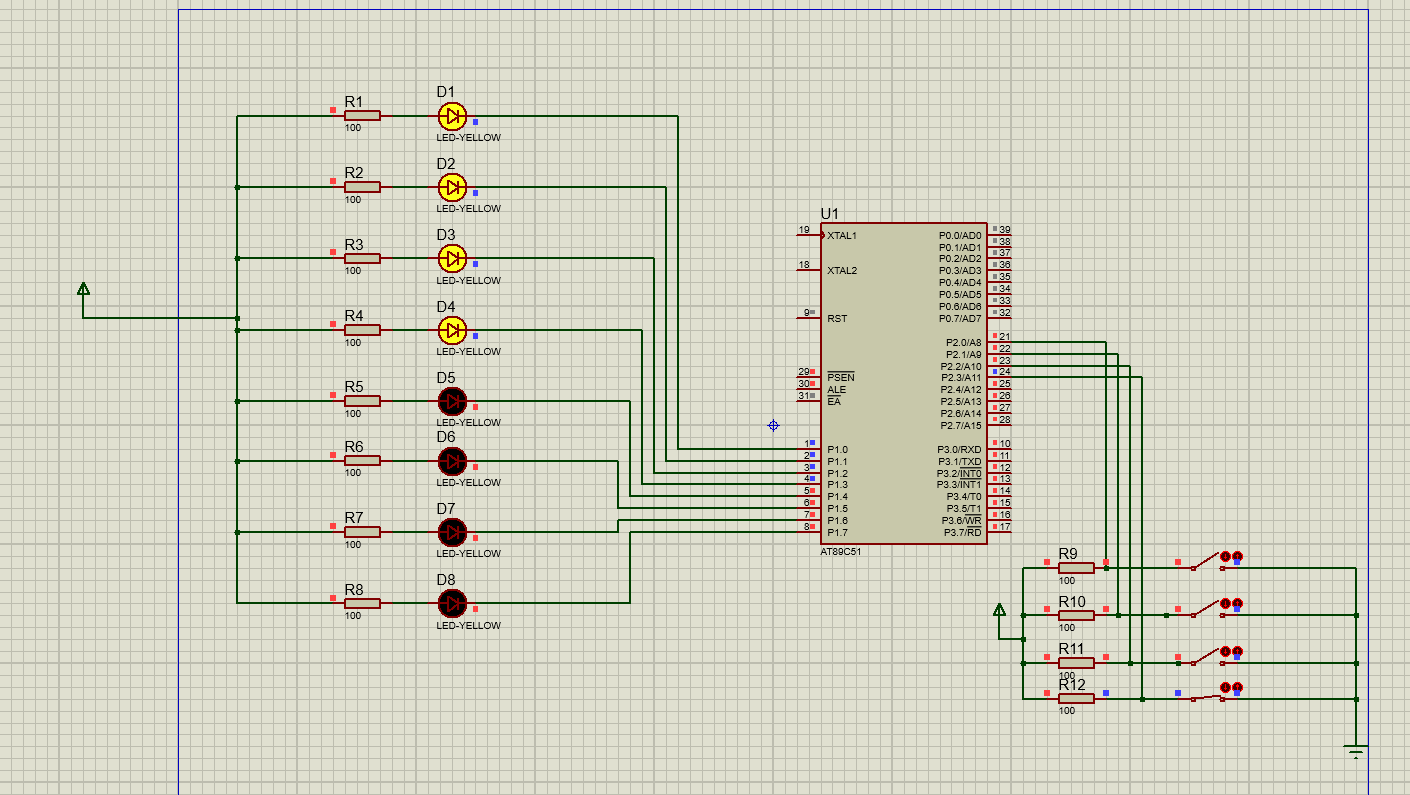


图6 功能演示图

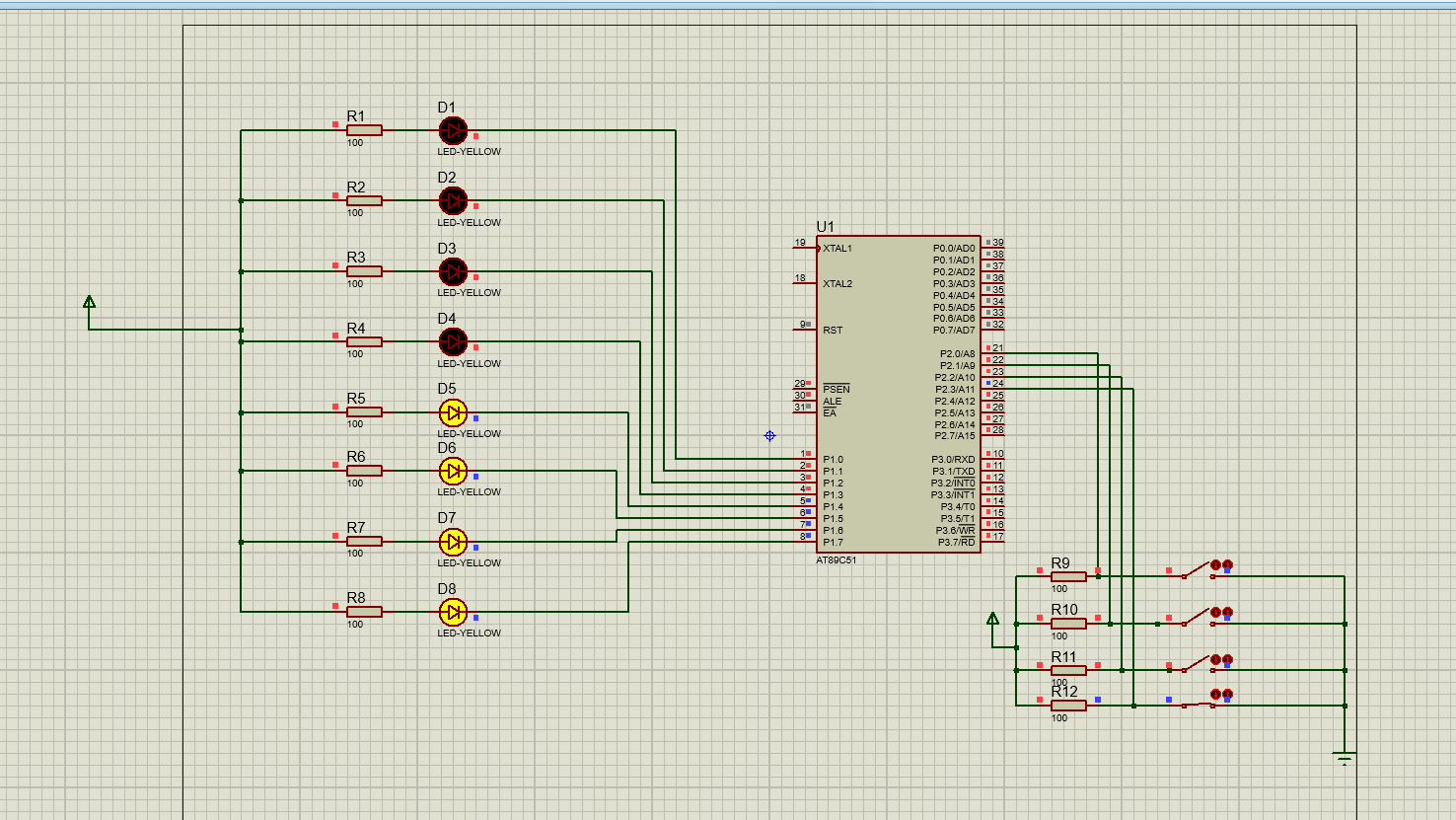


图7 功能演示图

**五、实验结论及分析**

1.总体分析：

（1）想要实现通过多个开关控制LED灯闪烁的效果，我选择P1端口作为LED灯的接口，P2口作为开关控制端口

（2）在电路中，我放置多个100欧姆的电阻用来保护电路

（3）想让LED灯发光，那么需要使连接接口的一端为低电平

（4）我通过if函数实现对开关的检测，当开关合上时，P2端口为低电平，否则为高电平

2.功能1分析：

（1）针对从上至下的流水灯效果，我选择使用移位来实现： P1 = ~(1 << m);

（2）增加延时函数，使LED流水效果更加明显

（3）利用while实现流水灯的反复运行，当if(P20==1)时，跳出while循环

3.功能2分析：

（1）针对从下至上的流水灯效果，我同样选择使用移位来实现：P1 = ~(1 << m);

，但是改变for循环实现相反的效果

（2）其余同功能1相同

4.功能3分析：

（1）通过在while循环中不断执行低电平和高电平的代码控制LED灯闪烁，增加延时函数使其效果更加明显

（2）本功能通过P2^3端口的开关来实现

5.功能4分析：

（1）本效果通过对P1端口进行更细致的控制来实现， P1=0xf0; P1=0x0f;

（2）其余同功能三相同

6.总结与心得体会

（1）通过本次实验，我深入了解了单片机的基本原理与应用，以及如何通过编程控制硬件来实现不同的功能。通过对四个开关的操作，我成功实现了四种不同的LED流水灯效果，包括上-下流水灯、下-上流水灯、循环闪动和四亮四灭循环。在实现这些功能的过程中，我学会了如何使用Keil编译软件编写程序，并通过Proteus仿真软件进行验证和调试。

（2）在硬件设计方面，我了解了如何使用电阻来保护电路，并将LED连接到P1端口，将开关连接到P2端口，通过检测P2端口的状态来控制LED的状态。通过掌握这些基本原理，我能够更好地理解单片机的工作原理和编程过程。

（3）在功能实现方面，我学会了如何使用移位操作来控制LED的流水效果，以及如何利用延时函数来调整LED的闪烁速度。我还学会了如何使用循环和条件语句来实现不同的LED效果，这些都是在单片机编程中非常重要的技能。

总的来说，通过这次实验，我对单片机原理与应用有了更深入的理解，提高了我的编程能力和硬件设计能力。这将对我的未来学习和工作都大有裨益。

计算机科学与技术系实验报告

课程名称：单片机原理与应用

班级：计科21-4 姓名：赵泽辉 学号：4202150101443 成绩：

实验项目名称：LED数码管的静态显示

**一、实验目的**

LED数码管的静态显示（1个SWITCH控制生日的年，月，日），1个数码管。

**二、实验设备**

1.pc机器一台

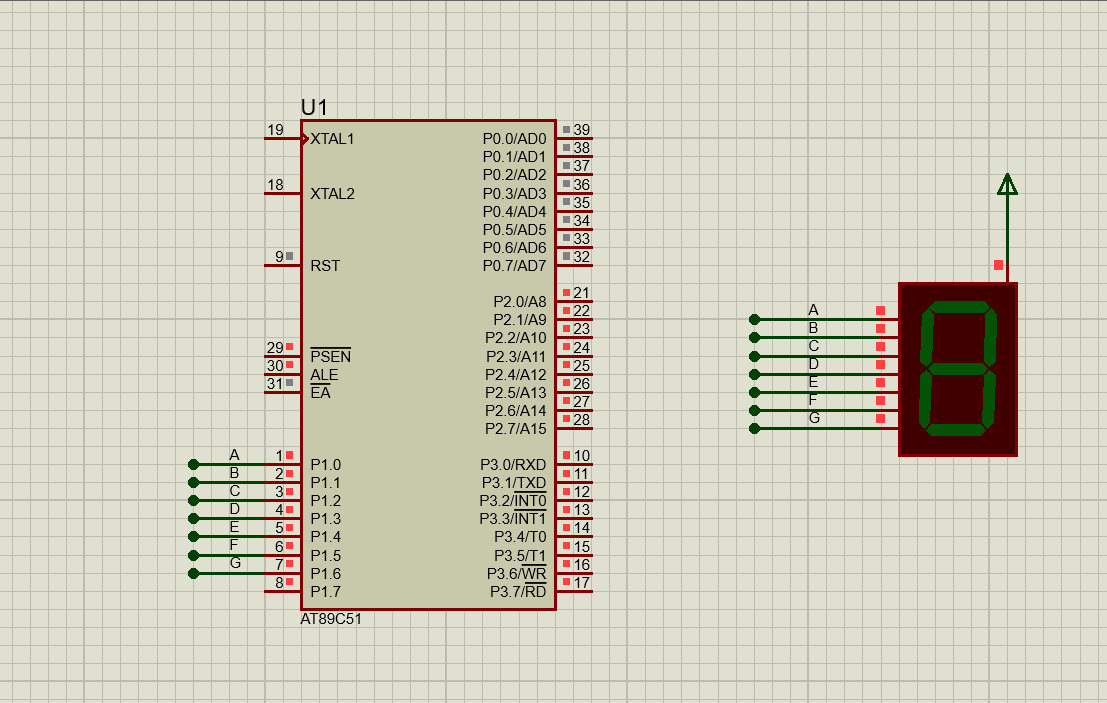
2.keil编译软件与Proteus仿真软件

**三、实验内容**

生日通过数码管实现静态显示

**四、实验步骤**

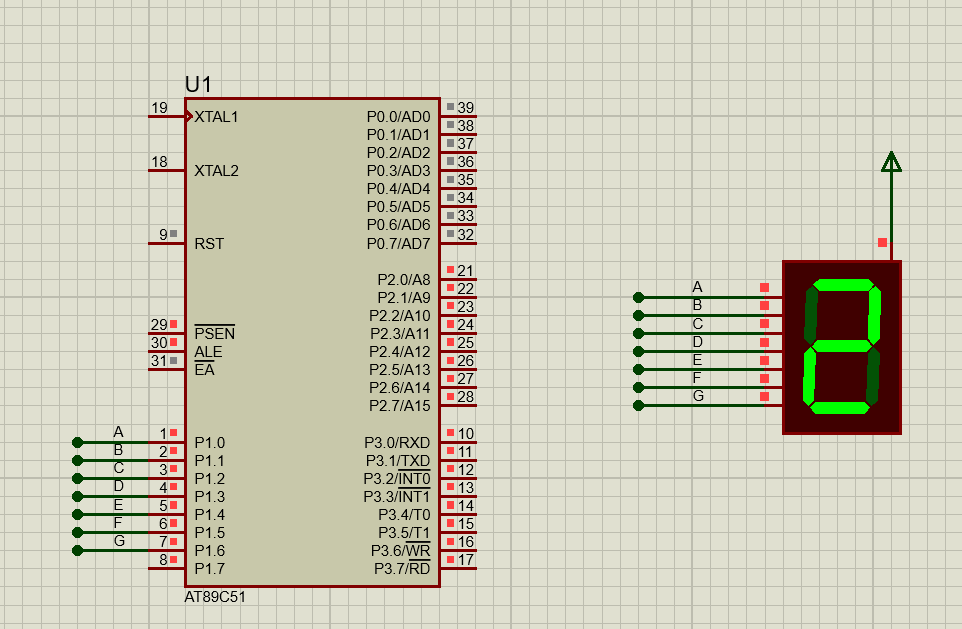
**（1）Proteus仿真图：**



**图1 仿真图**

1. **软件设计**
2. #include <reg51.h>
3. #define uint unsigned int
4. #define uchar unsigned char
5. uchar seg[]={0xa4,0xc0,0xc0,0xa4,0xf9,0xc0,0xa4,0xa4};
6. **void** delay()
7. {
8. uint i,j;
9. **for**(i=0;i<255;i++)
10. **for**(j=0;j<255;j++);
11. }
12. **void** main ()
13. {
14. uchar n;
15. **while** (1)
16. {
17. **for**(n=0;n<8;n++)
18. {
19. P1=seg[n];
20. delay();
21. P1=0xff;
22. delay();
23. **if** (n==10)
24. n=0;
25. }
26. }
27. }

**（3）效果图**



图二 仿真效果图

**五、实验结论及分析**

1.实验结果总结：

（1）我成功实现了使用LED数码管的生日年、月、日的静态显示，

（2）数码管按照预定的模式循环显示，通过编写的程序，每次显示一个数字，然后短暂延时后再切换到下一个数字，达到了静态显示的效果。

（3）在Proteus仿真软件中，验证了程序的正确性，并通过实物连接演示进一步确认了实验的有效性。

2.实验过程分析：

（1）在软件设计阶段，我们采用了Keil编译软件，通过编写程序代码实现LED数码管的控制。代码中使用了延时函数，确保每个数字的显示时间，同时通过循环控制实现了数字的连续显示。

（2）Proteus仿真图显示了程序在虚拟环境中的运行情况，这有助于在实际连接硬件之前验证程序的正确性。

（3）在实物连接及演示中，通过实际硬件连接和演示，验证了程序在实际单片机上的运行情况，确保了软硬件的协同工作。

3.实验问题与改进：

（1）在实验过程中，遇到了一些问题，例如电路连接错误、程序逻辑错误等，及时发现并解决这些问题。

（2）为了进一步完善实验，可以考虑增加一些异常处理机制，例如当输入的年月日超出合理范围时，给予相应的提示或默认值。

4.结论：

（1）通过本次实验，我们深刻理解了单片机在LED数码管静态显示方面的原理与应用，进一步学习了如何在单片机中实现静态的显示模式，提高了对单片机编程和硬件控制的实际操作能力。

（2）实验结果表明，成功实现了预期的功能，验证了程序的正确性，为今后在单片机应用领域中的更复杂任务奠定了基础。

综上所述，本实验对我们深入了解单片机的工作原理、软硬件协同开发具有积极的意义，为今后的学习和应用打下了坚实的基础。

计算机科学与技术系实验报告

课程名称：单片机原理与应用

班级：计科 21-4 姓名：赵泽辉 学号：4202150101443 成绩：

实验项目名称：LED数码管的动态显示

**[一． 实验目的](#_Toc22732_WPSOffice_Level2)**

用单片机控制8数码管显示，先从左至右慢速动态扫描数字“生日”

**[二． 实验设备](#_Toc7708_WPSOffice_Level2)**

1.pc机器一台

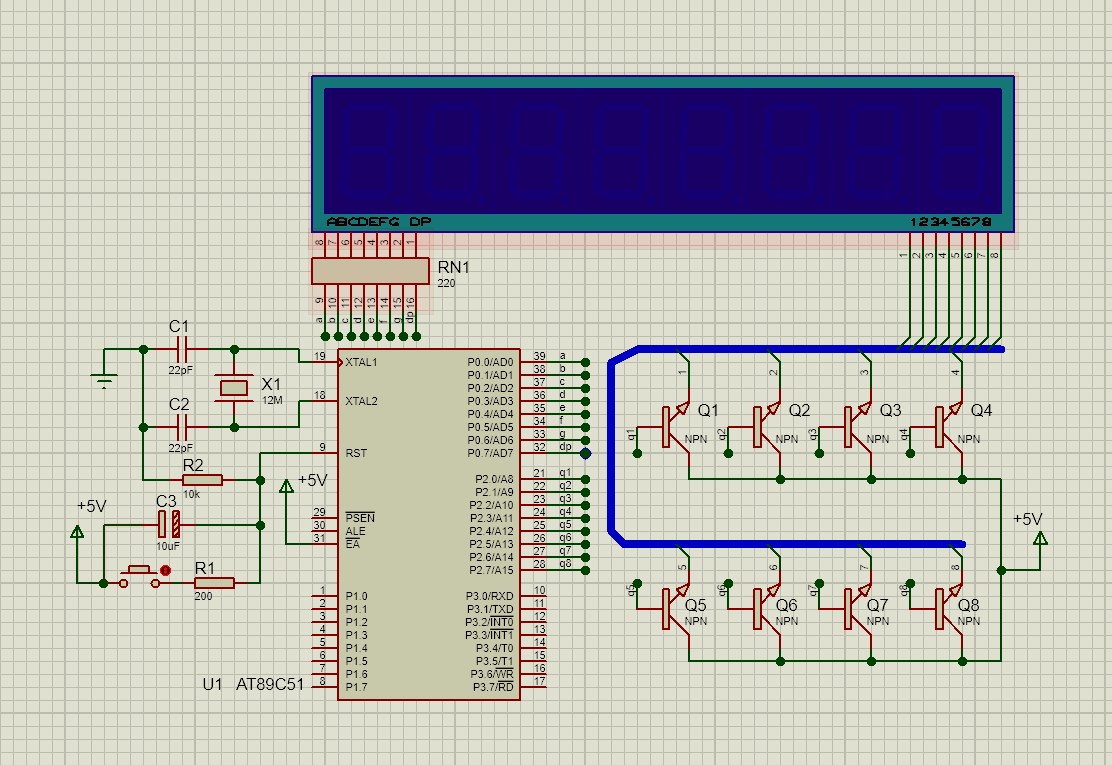
2.keil编译软件与Proteus仿真软件

**[三． 实验内容](#_Toc11783_WPSOffice_Level2)**

生日通过数码管动态显示

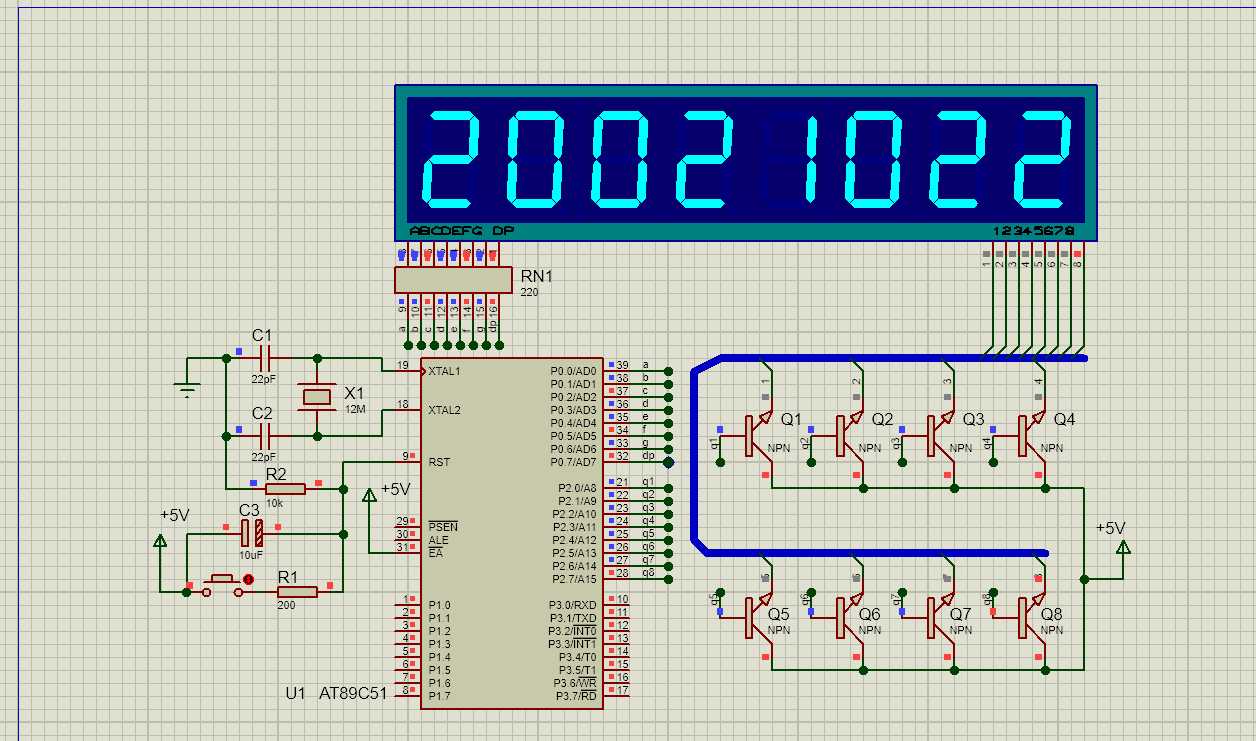
**[四． 实验步骤](#_Toc32164_WPSOffice_Level2)**

1. **Proteus仿真图：**



**（2）软件设计**

1. #include <reg51.h>
2. #include <intrins.h>
3. #define uchar unsigned char
4. #define unit unsigned int
5. uchar code dis\_code[]={0xa4,0xc0,0xc0,0xa4,0xc0,0xa4,0xf9,0x10};
6. //uchar code dis\_code[]={0xa4,0xc0,0xc0,0xa4,0xf9,0xc0,0xa4,0xa4};
7. **void** delay()
8. {
9. uchar i,j;
10. **for**(i=0;i<10;i++)
11. **for**(j=0;j<10;j++);
12. }
13. **void** main()
14. {
15. uchar i,j=0x80;
16. **while**(1)
17. {
18. **for**(i=0;i<8;i++)
19. {
20. j=\_crol\_(j,1);
21. P0=dis\_code[i];
22. P2=j;
23. delay();
24. P0=0XFF;
25. }
26. }
27. }
28. **效果图**



**[五． 实验结论及分析](#_Toc29664_WPSOffice_Level2)**

1.结论

通过本次实验，成功实现了使用单片机控制8位数码管动态显示“生日”字样。利用Keil编译软件和Proteus仿真软件进行代码设计和仿真，实现了逐个显示字符并从左至右进行动态扫描。代码中使用了延时函数以及对数码管显示的控制，顺利地完成了预期的功能。

2.分析

编程实现： 代码中的dis\_code数组储存了要显示的字符对应的数码管段选控制码，通过循环依次显示这些字符，配合位移操作（\_crol\_函数）实现了从左至右的动态显示效果。

动态扫描原理： 使用了位移操作 j=\_crol\_(j,1) 实现了数码管的动态扫描显示。这种方法通过改变j的值来选择点亮哪一位数码管，从而形成逐个扫描的效果。

延时控制： delay() 函数用于产生延时，控制数码管的刷新速度。通过调整延时时间可以改变数码管刷新的速度，实现不同的显示效果。

硬件控制： 代码中通过 P0 和 P2 控制数码管的段选和位选，从而实现了字符的显示和动态扫描。

3.总结

这个实验展示了如何使用单片机控制数码管动态显示字符，并通过软件设计和仿真实现了预期的效果。理解并掌握了位移操作、延时控制和端口控制等基本原理，对单片机应用有了更深入的认识和理解。

计算机科学与技术系实验报告

课程名称：单片机原理与应用

班级：计科 21-4 姓名：赵泽辉 学号：4202150101443 成绩：

实验项目名称：键盘接口设计

**[一． 实验目的](#_Toc22732_WPSOffice_Level2)**

学会使用4\*4矩阵键盘，并通过数码管显示矩阵键盘的键号

**[二． 实验设备](#_Toc7708_WPSOffice_Level2)**

1.pc机器一台

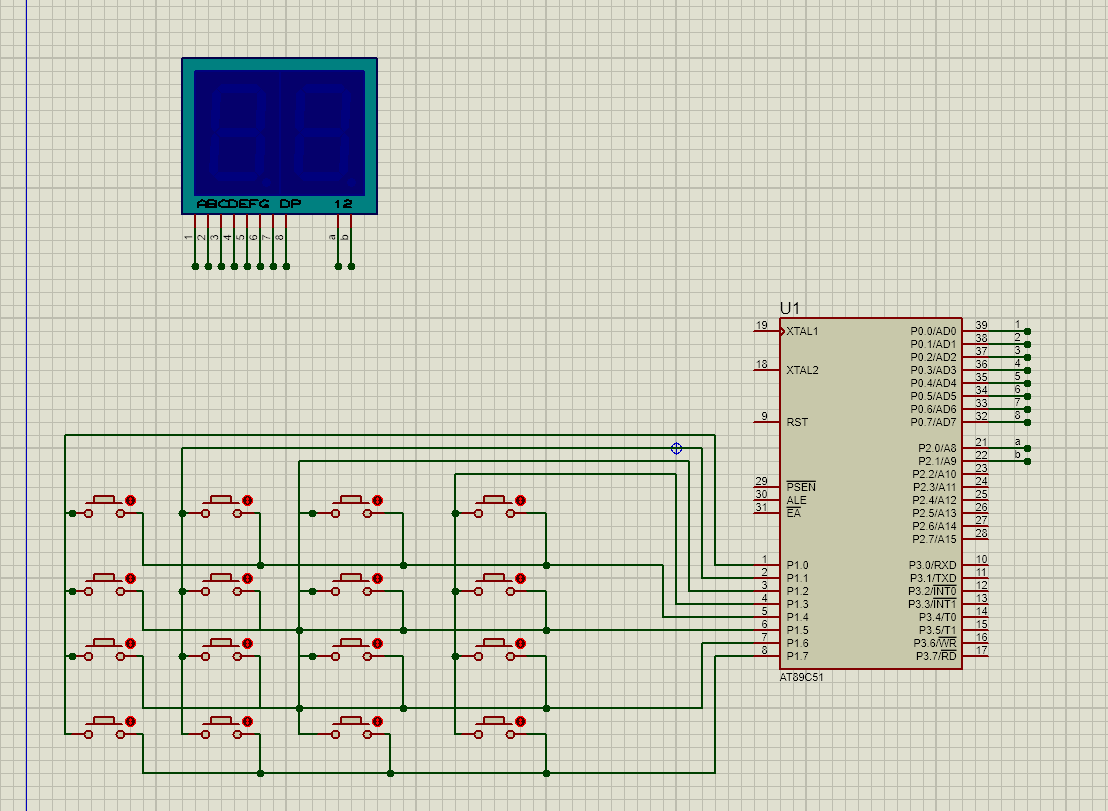
2.keil编译软件与Proteus仿真软件

**[三． 实验内容](#_Toc11783_WPSOffice_Level2)**

4/Proteus 虚拟仿真。单片机的P1口的P1.0~P1.7 连接4×4矩阵键盘，并通过PO口控制2LED数码管显示4×4矩阵键盘16个按键的键号，键号分别为“0,1,,9,A,B,…,E,F”。当键盘中的某个按键按下时，2位数码管上显示对应的十进制键号。例如，1号键按下时，2位数码管显示“01”；E号键按下时，2位数码管显示“14”等

**[四． 实验步骤](#_Toc32164_WPSOffice_Level2)**

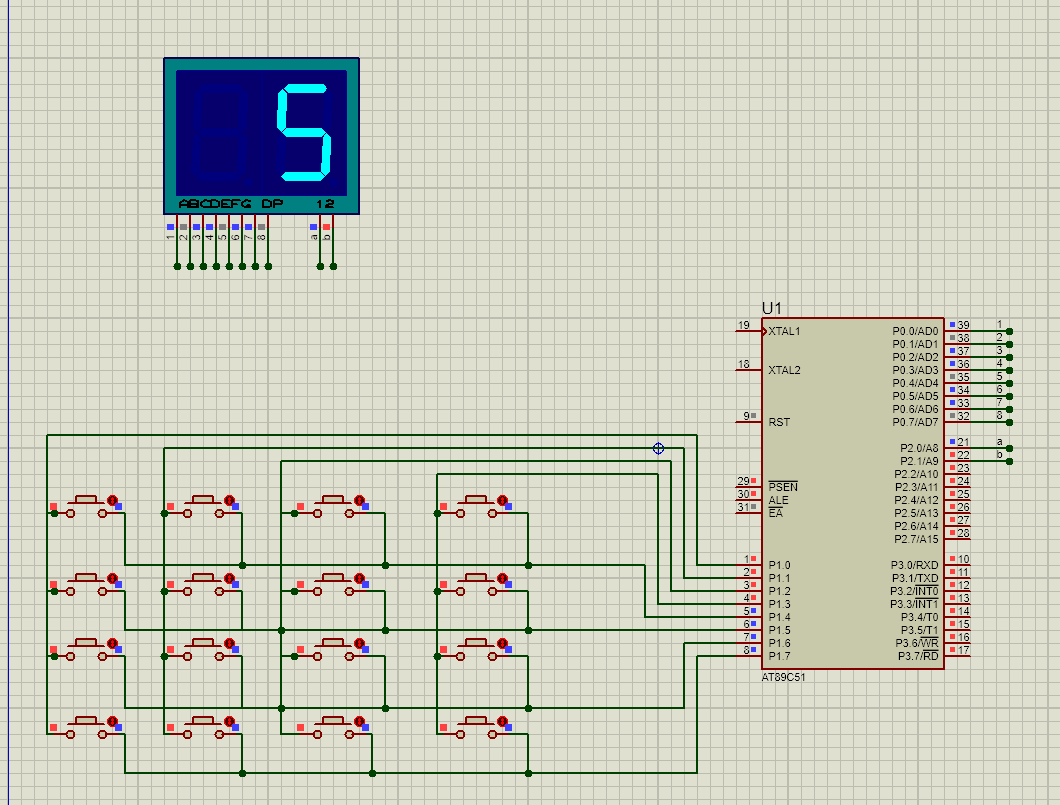
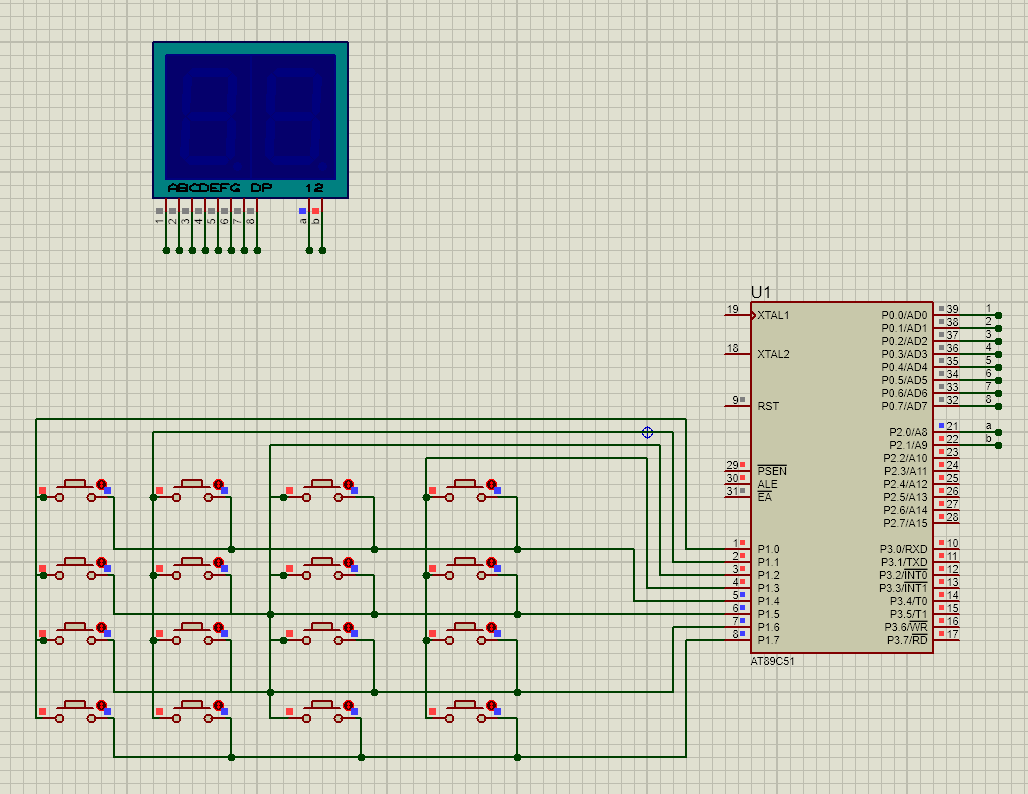
**（1）Proteus仿真图：**

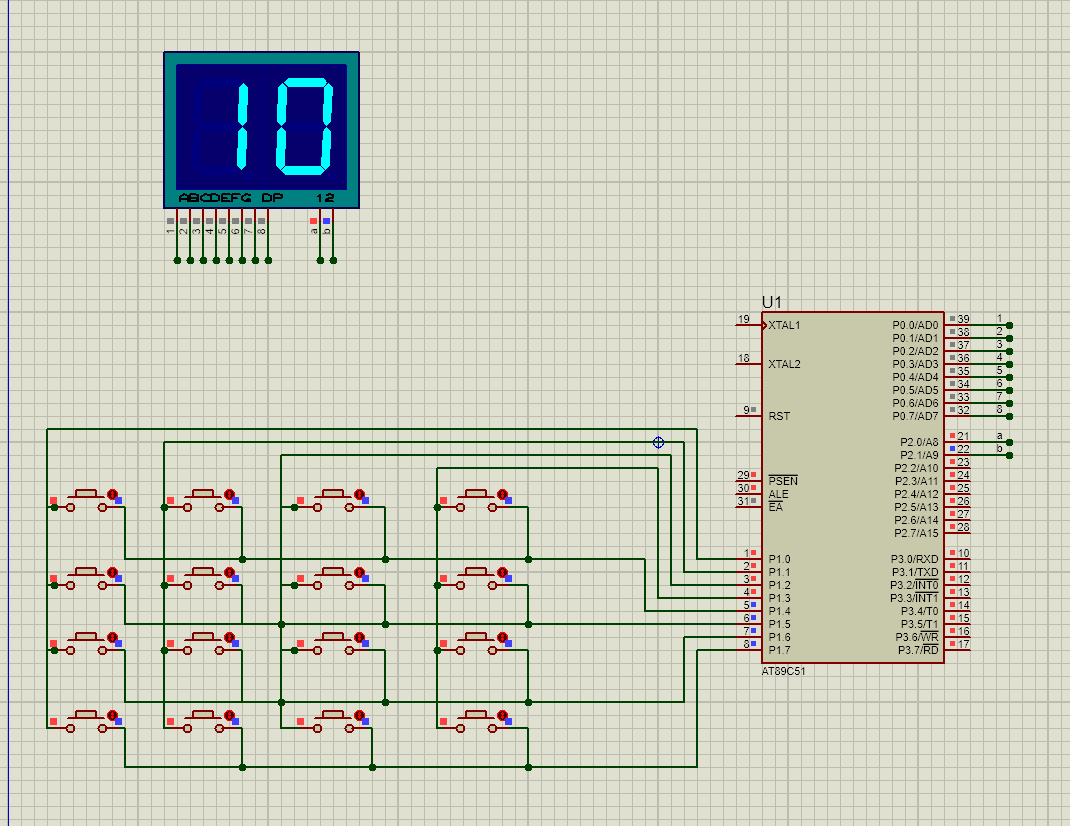


（2）**软件设计**

1. #include <reg51.h>
2. #define uchar unsigned char
3. #define uint unsigned int
4. uchar keyval=0;      //定义一个键值 判断按钮
5. uchar a[11]={0xc0,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};       //0-9十个共阳数组
6. uint k,i=0;   //定义k用于动态刷新两个数码管
7. **void** delay10()    //十毫秒延时
8. {
9. uchar i,j;
10. **for**(i=0;i<100;i++)
11. **for**(j=0;j<100;j++)
12. ;
13. }
15. **void** down()    //判断按下按键的键值
16. {
17. P1=0x0f;
19. **if**(P1!=0x0f)
20. {
21. delay10();
22. i=i+1;
23. **switch**(P1)     //判断列
24. {
25. **case** (0x0e) : keyval=0; **break**;
26. **case** (0x0d) : keyval=1; **break**;
27. **case** (0x0b) : keyval=2; **break**;
28. **case** (0x07) : keyval=3; **break**;
29. }
30. P1=0xf0;        //根据行判断按键
31. **switch**(P1)
32. {
33. **case** (0xe0) : keyval=keyval;**break**;
34. **case** (0xd0) : keyval=keyval+4;**break**;
35. **case** (0xb0) : keyval=keyval+8;**break**;
36. **case** (0x70) : keyval=keyval+12;**break**;
37. }
38. }
39. }
40. **void** main()
41. {
42. **while**(1)
43. {
44. down();  //获取按键键值
45. **if**(keyval<10&&keyval>=0&&i!=0)     //一位数字 只运行一个数码管
46. {
47. P2=0xfe;
48. P0=a[keyval];
49. }
50. **else**                        //两位数字
51. {
52. **for**(k=0;k<2;k++)   //动态刷新两个数码管
53. {
54. **if**(k==0)
55. {
56. P2=0xfd;
57. P0=a[keyval/10];
58. P0=0xff;
59. }
60. **else**
61. {
62. P2=0xfe;
63. P0=a[keyval%10];
64. P0=0xff;
65. }
66. }
67. }
68. }
69. }

（3）效果图





**[五． 实验结论及分析](#_Toc29664_WPSOffice_Level2)**

这次的实验是关于使用4×4矩阵键盘，并通过数码管显示按键的十进制键号。在实验中，通过Proteus虚拟仿真，将单片机的P1口的P1.0~P1.7连接到4×4矩阵键盘，同时通过PO口控制2位LED数码管显示16个按键的键号，这些键号包括数字 0 到 9 和字母 A 到 F。

在软件设计中，采用了基于单片机的程序设计，利用延时、按键扫描和数码管显示等方法来实现按键检测和显示相应按键的十进制键号。程序设计中使用了按键检测函数 down()，根据按键的行列来确定按键的键值，并通过数码管来显示相应的按键编号。程序的主要逻辑是不断扫描按键状态，如果检测到按键按下，则根据按键的行列位置来确定其对应的键值，并在数码管上显示出来。

通过这次实验，我们成功地掌握了使用矩阵键盘的方法，并且实现了通过数码管显示按键的十进制键号的功能。这个实验对于理解单片机输入输出控制、按键检测以及数码管显示等方面有着重要的意义。

总体来说，这次实验有效地实现了设定的目标，成功地运用了单片机编程和硬件连接来完成按键检测和数码管显示，为后续更复杂功能的实现提供了基础。

计算机科学与技术系实验报告

课程名称：单片机原理与应用

班级：计科 21-4 姓名：赵泽辉 学号：4202150101443 成绩：

实验项目名称：中断系统的应用

**[一． 实验目的](#_Toc22732_WPSOffice_Level2)**

了解并掌握中断系统的使用控制与原理

**[二． 实验设备](#_Toc7708_WPSOffice_Level2)**

1.pc机器一台

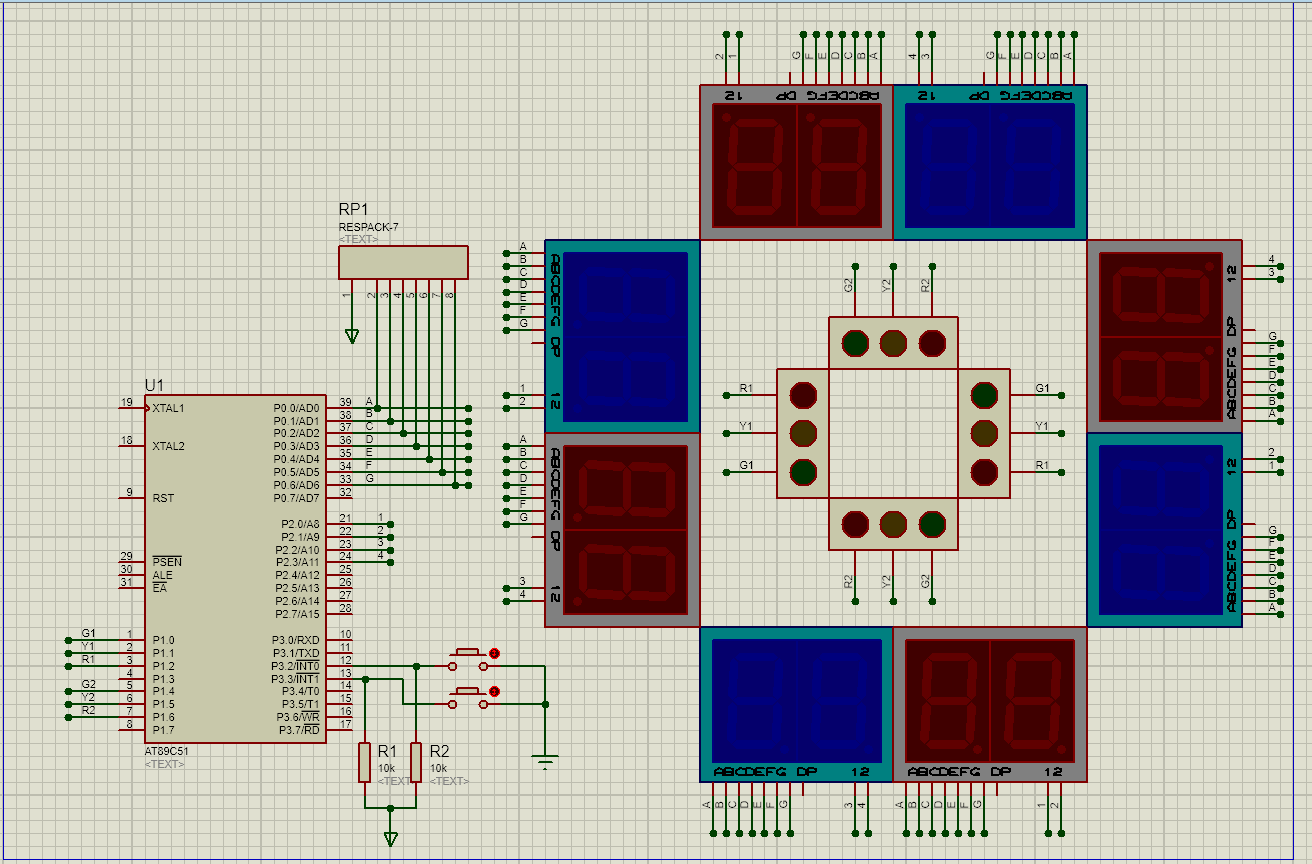
2.keil编译软件与Proteus仿真软件

**[三． 实验内容](#_Toc11783_WPSOffice_Level2)**

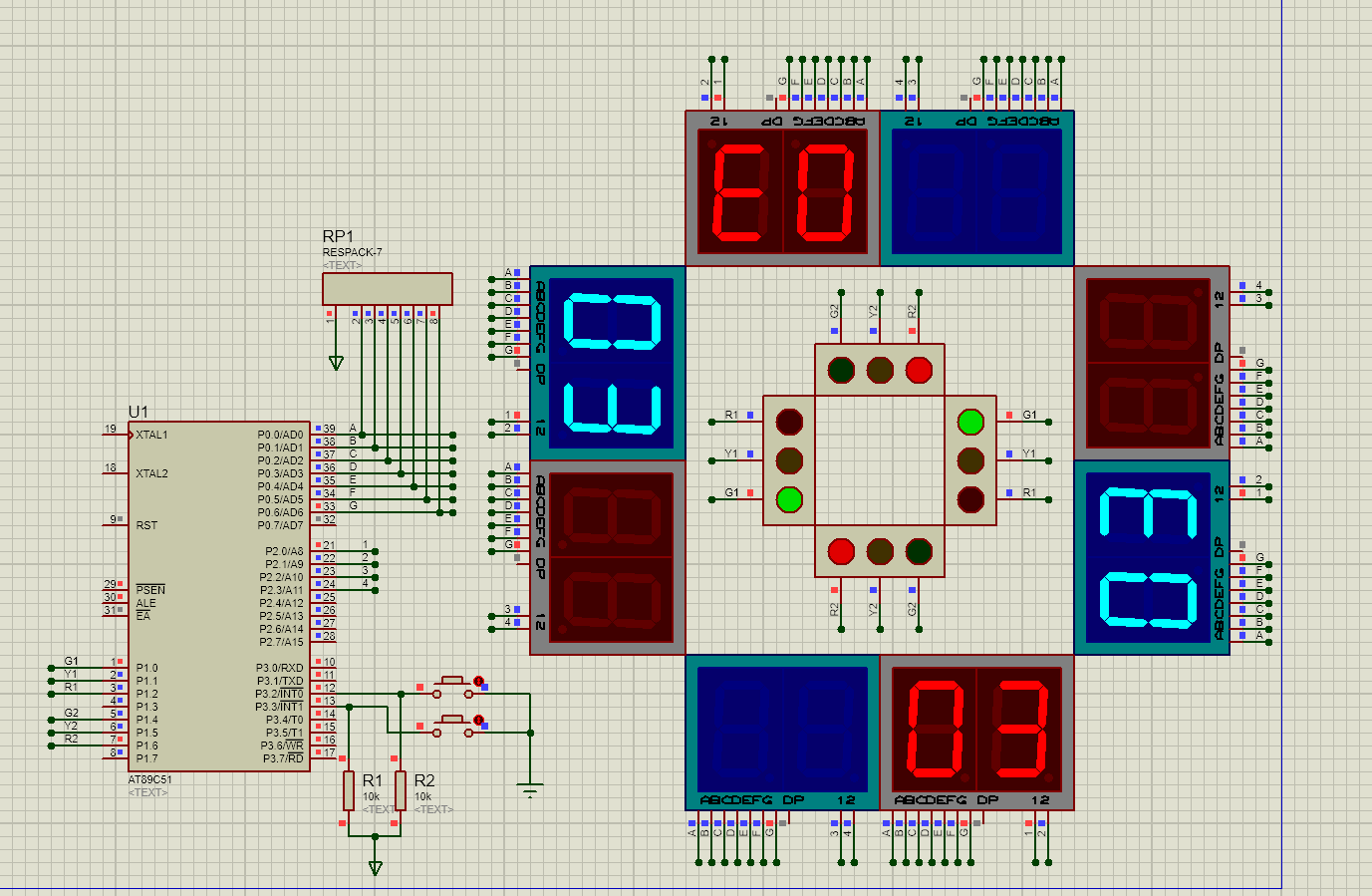
使用C51单片机制作红绿灯控制系统，并加入中断系统控制

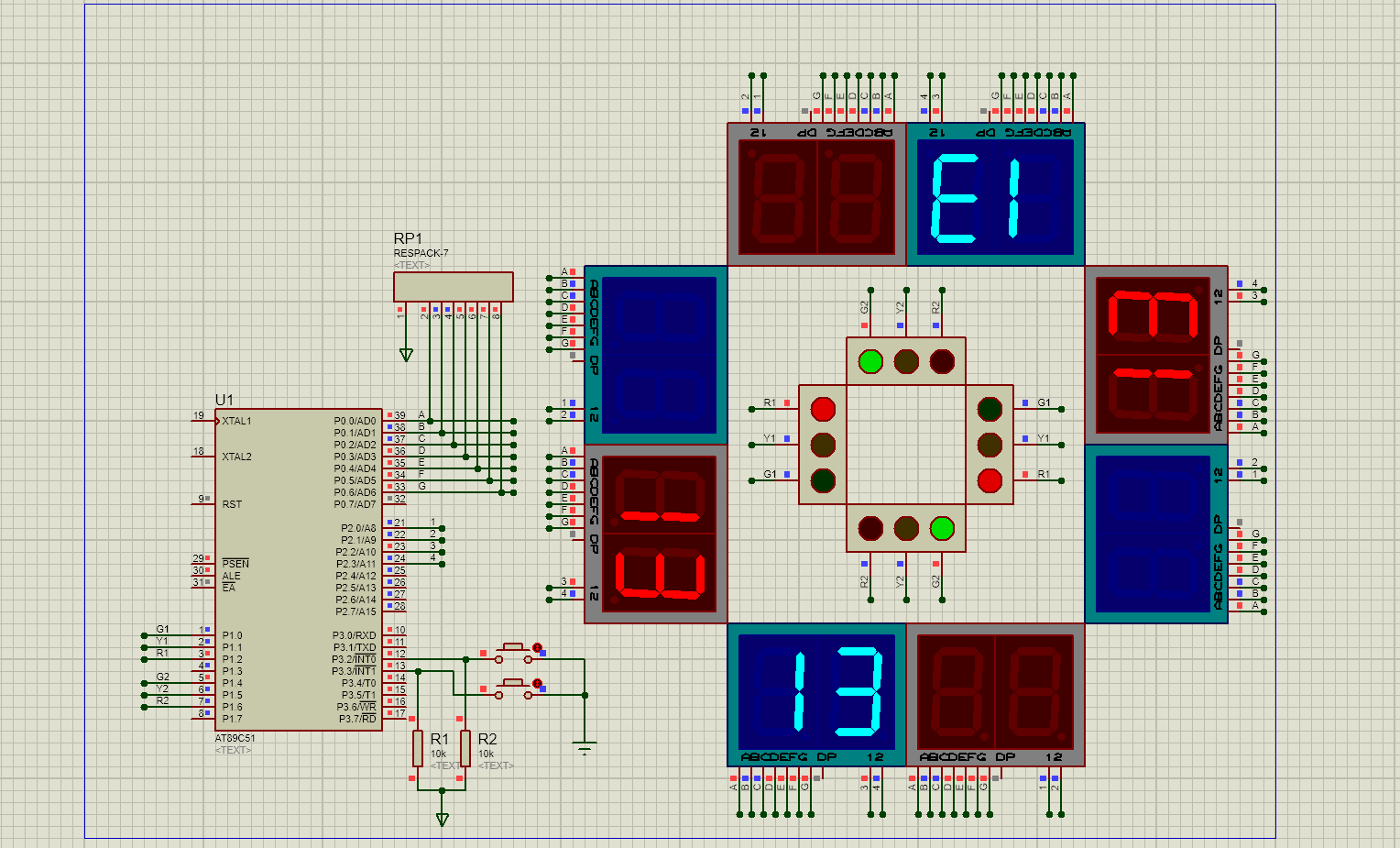
**[四． 实验步骤](#_Toc32164_WPSOffice_Level2)**

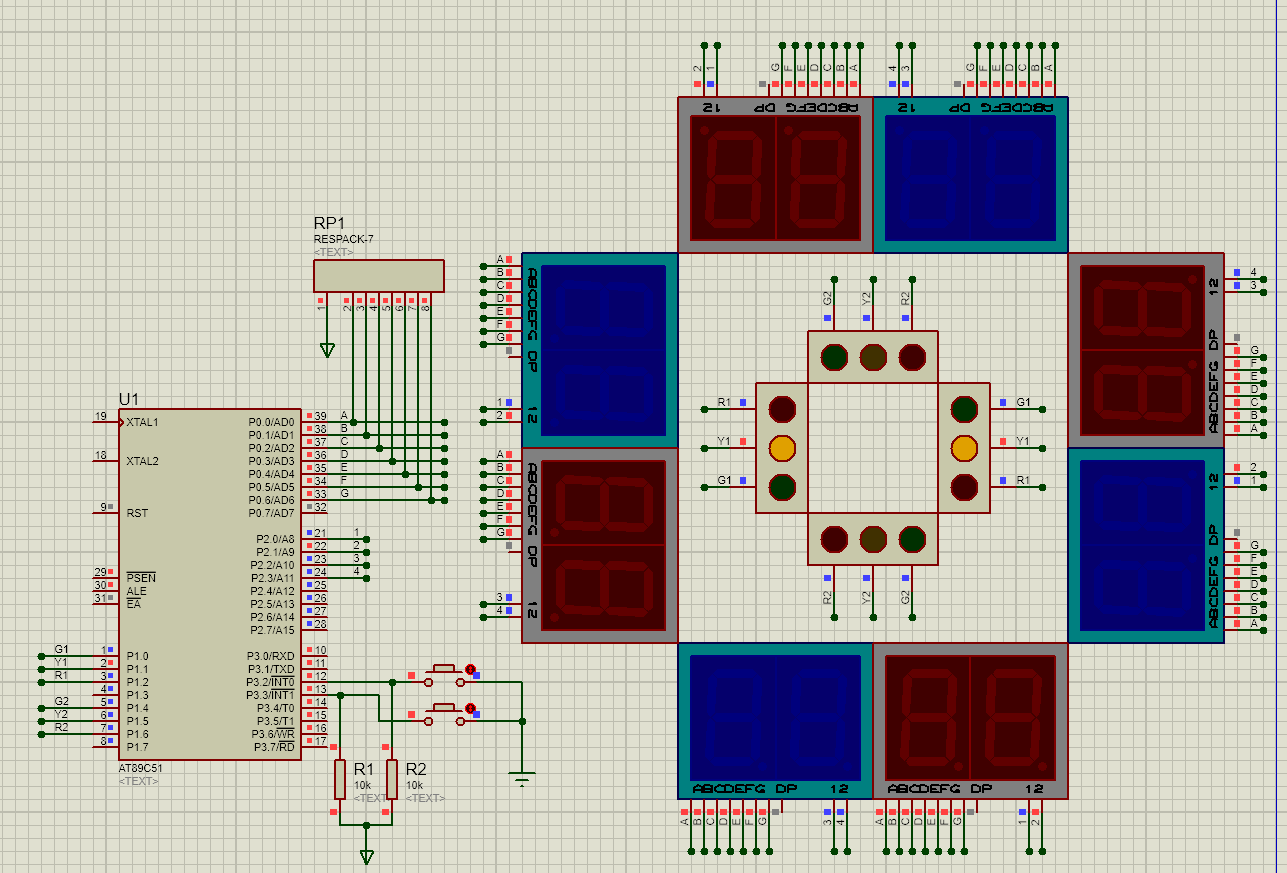
1. **Proteus仿真图**



1. **软件设计**
2. #include<reg51.h>
3. #define uchar unsigned char
4. uchar sz[]={0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0x80,0x90};
5. sbit G1=P1^0;
6. sbit Y1=P1^1;
7. sbit R1=P1^2;
8. sbit G2=P1^4;
9. sbit Y2=P1^5;
10. sbit R2=P1^6;
11. **void** delay(){
12. uchar i,j;
13. **for**(i=0;i<255;i++)
14. **for**(j=0;j<255;j++)  ;
15. }
16. **void** zd(){             //发出中断及中断被允许
17. IT0=1;
18. IT1=1;
19. EA=1;
20. EX0=1;
21. EX1=1;
22. }
23. **void** LED\_1(){          //东西绿灯（蓝数码管），南北红灯{红数码管）
24. uchar m,n;
25. P0=0xff;
26. P1=0x41;
27. **for**(m=0;m<3;m++){
28. **for**(n=0;n<10;n++){
29. P2=0x01;
30. P0=sz[m];
31. delay();
32. P2=0x02;
33. P0=sz[n];
34. delay();
36. **if**(m==2&&n>=7){
37. P1=0x00;
38. delay();
39. P1=0x22;
40. delay();
41. P1=0x00;
42. }
43. }
44. delay();
45. }
46. P0=0xff;
47. }
48. **void** LED\_2(){         //东西红灯（红数码管），南北绿灯{蓝数码管）
49. uchar m,n,j;
50. P0=0xff;
51. P1=0x14;
52. **for**(m=0;m<3;m++){
53. **for**(n=0;n<10;n++){
54. P2=0x04;
55. P0=sz[m];
56. delay();
57. P2=0x08;
58. P0=sz[n];
59. delay();
61. **if**(m==2&&n>=7){
62. P1=0x00;
63. delay();
64. P1=0x22;
65. delay();
66. P1=0x00;
67. }
68. }
69. delay();
70. }
71. P0=0xff;
72. }
73. **void** main(){
74. zd();
75. **while**(1){
76. LED\_1();
77. LED\_2();
78. }
79. }
80. **void** int0() interrupt 0{
81. uchar i;
82. P0=0xff;
83. P1=0x00;
84. **for**(i=0;i<60;i++){
85. Y1=1;
86. delay();
87. Y1=0;
88. delay();
89. Y1=1;
90. }
91. }
92. **void** int1() interrupt 2{
93. uchar i;
94. P0=0xff;
95. P1=0x00;
96. **for**(i=0;i<60;i++){
97. Y2=1;
98. delay();
99. Y2=0;
100. delay();
101. Y2=1;
102. }
103. }
104. **效果图**







**[五． 实验结论及分析](#_Toc29664_WPSOffice_Level2)**

在这个实验中，我们成功地实现了一个红绿灯控制系统，并将中断系统引入到控制中。通过使用C51单片机、Keil编译软件和Proteus仿真软件，我们构建了一个可以控制红绿灯交替变换的系统。

在软件设计部分，我们利用了C语言编写了控制程序。通过对IO口的控制，我们实现了红绿灯的交替变换，并且在程序中加入了中断服务函数，使得当特定事件发生时，可以触发中断并执行相应的操作。其中，我们使用了外部中断0（INT0）和外部中断1（INT1），分别控制了东西方向和南北方向的红绿灯切换。

在效果图中，可以清楚地看到红绿灯的交替变换，符合预期的系统设计要求。红绿灯在一定时间内按照设定的规律变换，同时通过数码管显示当前的状态信息。

通过这个实验，我们深入了解了中断系统的使用控制与原理。中断系统的引入使得系统可以在特定事件发生时做出及时响应，提高了系统的实时性和灵活性。同时，通过对单片机的编程设计，加深了对单片机原理与应用的理解和掌握。

综上所述，本次实验成功地实现了中断系统的应用，并且通过红绿灯控制系统展示了中断系统在实际应用中的重要性和有效性。

计算机科学与技术系实验报告

课程名称：单片机原理与应用

班级：计科 21-4 姓名：赵泽辉 学号：4202150101443 成绩：

实验项目名称：定时/计数器的应用

**[一． 实验目的](#_Toc22732_WPSOffice_Level2)**

定时/计数器的应用

**[二． 实验设备](#_Toc7708_WPSOffice_Level2)**

1.pc机器一台

2.keil编译软件与Proteus仿真软件

**[三． 实验内容](#_Toc11783_WPSOffice_Level2)**

Proteus 虚拟仿真。制作一个采用LCD1602显示的电子钟，在LCD上显示当前的时间。显示格式为“时时：分分：秒秒”。设有4个功能键k1~k4，功能如下：

（1）kl——进入时间修改。

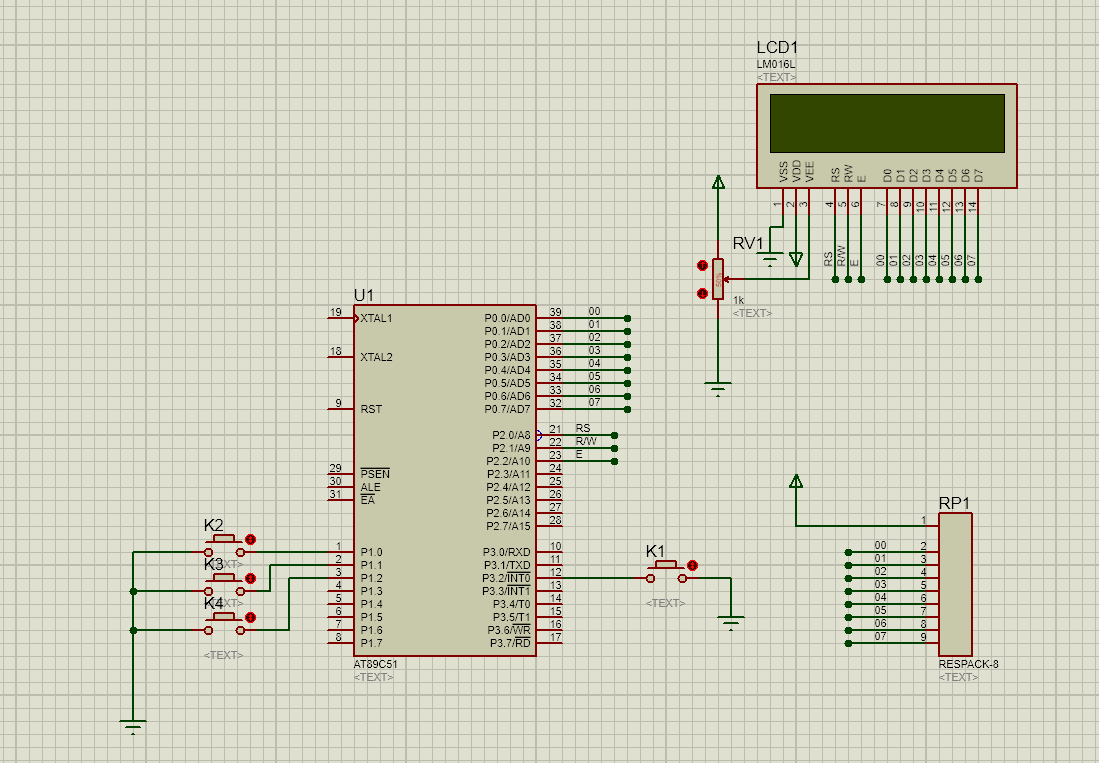
（2）K2——修改时数，按一下k2，当前时数增1。

（3）k3——修改分数，按一下k3，当前分数增1。

（4)k4—一确认修改完成，电子钟按修改后的时间运行显示。

**[四． 实验步骤](#_Toc32164_WPSOffice_Level2)**

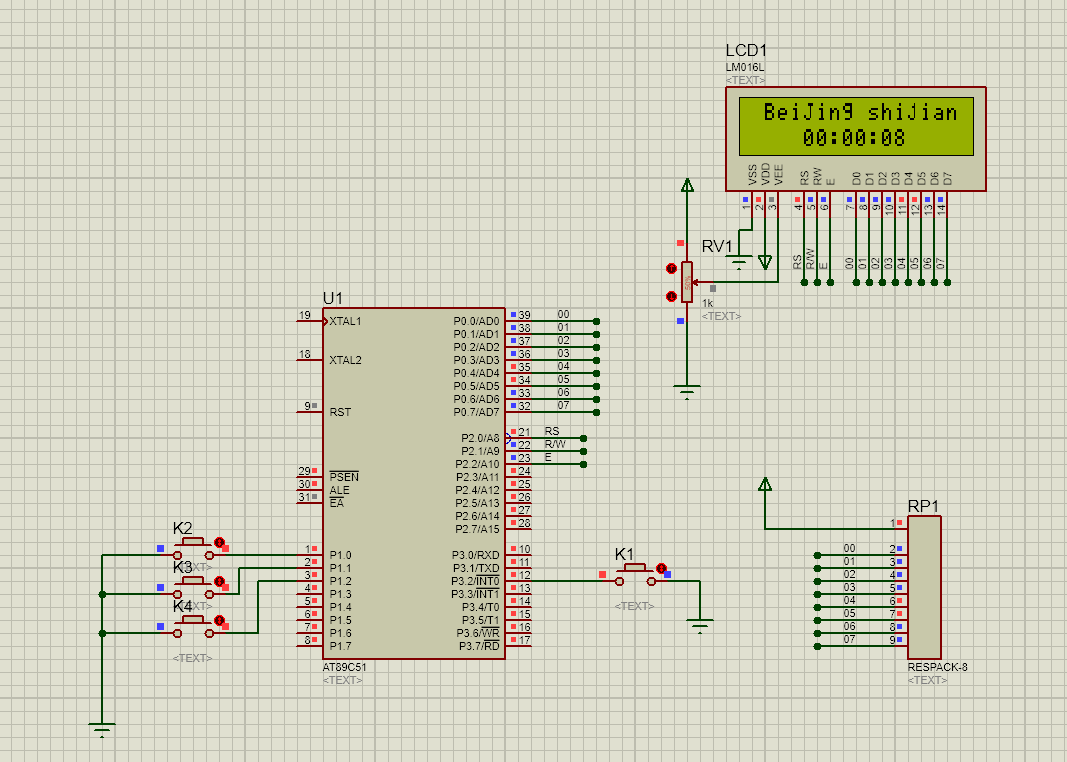
**（1）Proteus仿真图：**



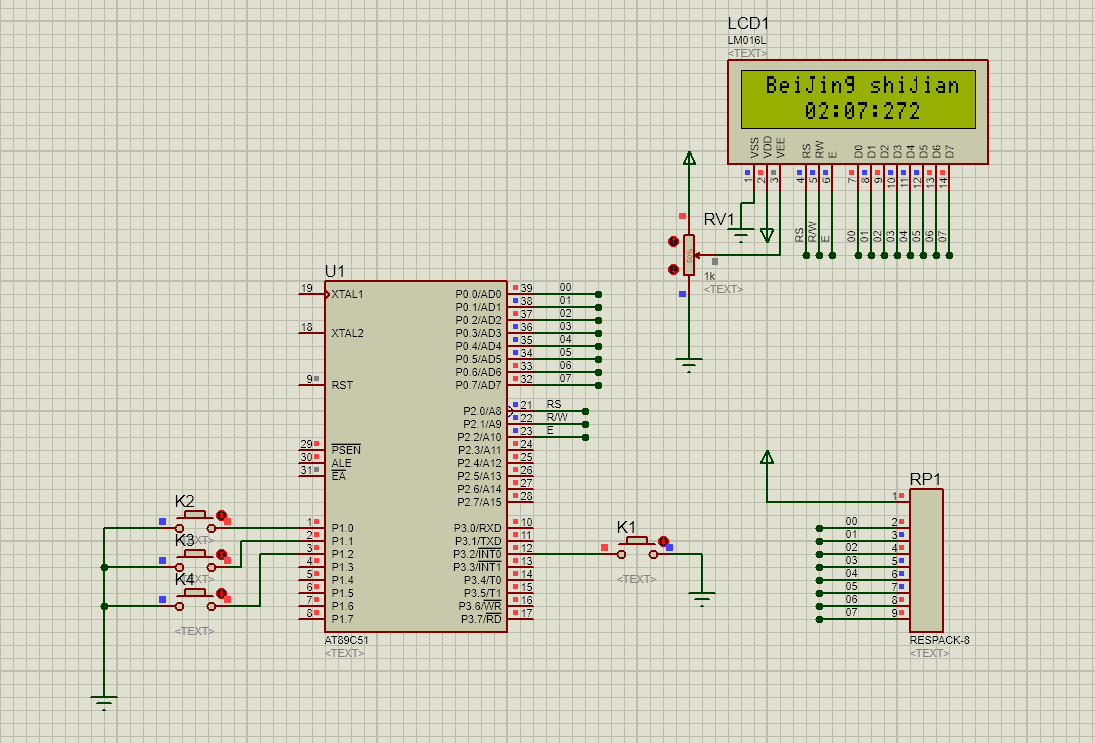
（2）软件设计

1. #include<reg51.h>
2. #include <intrins.h>
3. #define uchar unsigned char
4. #define uint unsigned int
5. #define out P0
6. sbit RS=P2^0;
7. sbit RW=P2^1;
8. sbit E=P2^2;
9. **void** delay(uint j)
10. {
11. uchar i=250;
12. **for**(;j>0;j--)
13. {
14. **while**(--i);
15. i=249;
16. **while**(--i);
17. i=250;
18. }
19. }
20. **void** check\_busy(**void**)
21. {
22. uchar dt;
23. **do**
24. {
25. dt=0xff;
26. E=0;
27. RS=0;
28. RW=1;
29. E=1;
30. dt=out;
31. }
32. **while**(dt&0x80);
33. E=0;
34. }
35. **void** write\_command(uchar com)
36. {
37. check\_busy();
38. E=0;
39. RS=0;
40. RW=0;
41. out=com;
42. E=1;
43. \_nop\_();
44. E=0;
45. delay(1);
46. }
47. **void** write\_data(uchar dat)
48. {
49. check\_busy();
50. E=0;
51. RS=1;
52. RW=0;
53. out=dat;
54. E=1;
55. \_nop\_();
56. E=0;
57. delay(1);
58. }
59. **void** lcd\_initial(**void**)
60. {
61. write\_command(0x38);
62. \_nop\_();
63. write\_command(0x0c);
64. \_nop\_();
65. write\_command(0x04);
66. \_nop\_();
67. write\_command(0x01);
68. delay(1);
69. }
70. **void** string(uchar ad,uchar \*s)
71. {
73. **while**(\*s!='\0')
74. {
75. write\_command(ad);
76. write\_data(\*s++);
77. ad++;
78. }
79. }
80. uint cishu;
81. uchar miao;
82. uchar fen;
83. uchar shi;
84. sbit K1=P3^2;
85. sbit K2=P1^0;
86. sbit K3=P1^1;
87. sbit K4=P1^2;
88. uchar miao=0,fen=0,shi=0;
89. uchar code tab[]={0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37,0x38,0x39};
90. **void** delay10ms()   //软件消抖延时
91. {
92. uchar i,j;
93. **for**(i=0;i<200;i++)
94. **for**(j=0;j<200;j++);
95. }
96. **void** clock\_init()//时钟初始化
97. {
98. string(0x81,"BeiJing shijian");
99. string(0xC4,"00:00:00");
100. }
101. **void** write\_sfm(uchar add,uchar date)
102. {
103. uchar ge,shi;
104. shi=tab[date/10];
105. ge=tab[date%10];
106. write\_command(add);
107. write\_data(shi);
108. write\_command(add+0x01);
109. write\_data(ge);
110. }
111. **void** clock\_write(uint s,uint m,uint h)//设置时、分、秒
112. {
113. write\_sfm(0xc4,h);
114. write\_sfm(0xc7,m);
115. write\_sfm(0xca,s);
116. }
117. **void** main()
118. {
119. lcd\_initial();
120. clock\_init();
121. TMOD=0X01;
122. EA=1;
123. PX0=1;
124. PT0=0;
125. ET0=1;
126. EX0=1;
127. IT0=1;
128. TH0=(65536-50000)/256;
129. TL0=(65536-50000)%256;
130. TR0=1;
131. cishu=0;
132. miao=0;
133. fen=0;
134. shi=0;
135. **while**(1)
136. {
137. clock\_write(miao,fen,shi);
138. }
139. }
140. **void** T0\_int(**void**) interrupt 1
141. {
142. cishu++;
143. **if**(cishu==20)
144. {
145. cishu=0;
146. miao++;
147. }
148. **if**(miao==60)
149. {
150. miao=0;
151. fen++;
152. }
153. **if**(fen==60)
154. {
155. fen=0;
156. shi++;
157. }
158. **if**(shi==24)
159. {
160. shi=0;
161. }
162. TH0=(65536-50000)/256;
163. TL0=(65536-50000)%256;
164. }
165. **void** xuishi(**void**) interrupt 0
166. {
167. **do**
168. {
169. EA=0;
170. **if**(K2==0)
171. {
172. delay10ms();
173. **if**(K2==0)
174. {
175. shi++;
176. clock\_write(miao,fen,shi);
177. }
178. }
179. **if**(K3==0)
180. {
181. delay10ms();
182. **if**(K3==0)
183. {
184. fen++;
185. clock\_write(miao,fen,shi);
186. }
187. }
188. }
189. **while**(K4!=0);
190. EA=1;
191. }

**（3）效果图**



通过中断修改时间



**[五． 实验结论及分析](#_Toc29664_WPSOffice_Level2)**

在这个实验中，成功地设计并实现了一个基于单片机的电子钟系统。该系统利用了LCD1602显示屏以及4个功能按键，能够显示当前的时间，并且通过按键进行时间的修改和确认操作。在这个实验中，几个关键的点需要进行结论和分析：

功能性分析： 实验目的是实现一个电子钟系统，最终成功实现了在LCD上显示时、分、秒，并且通过按键进行时间的修改和确认。这一功能性方面得到了很好的实现。

代码设计分析： 所给出的代码实现了LCD1602的初始化、显示字符串、显示时间等功能。其中利用定时器中断实现了时间的自动更新和计时。需要分析代码的逻辑性、结构性，以及代码的效率和可维护性等方面。

实验结果验证： 实验中的效果图显示了系统正常运行并能够通过按键修改时间，通过中断也能实现时间的自动更新。需要对实验结果的准确性和稳定性进行验证和分析，确保系统能够在长时间运行中保持准确的时间显示和稳定的运行状态。

实验应用与拓展： 电子钟系统作为一个典型的单片机应用，可以进一步思考其在实际生活中的应用场景，比如定时器、计时器的使用，或者是如何通过添加功能来改进这个系统，提高其实用性和便捷性等。

计算机科学与技术系实验报告

课程名称：单片机原理与应用

班级：计科 21-4 姓名：赵泽辉 学号：4202150101443 成绩：

实验项目名称：串行口的应用

**[一． 实验目的](#_Toc22732_WPSOffice_Level2)**

实现甲、乙机之间采用方式1双向串行通信

**[二． 实验设备](#_Toc7708_WPSOffice_Level2)**

1.pc机器一台

2.keil编译软件与Proteus仿真软件

**[三． 实验内容](#_Toc11783_WPSOffice_Level2)**

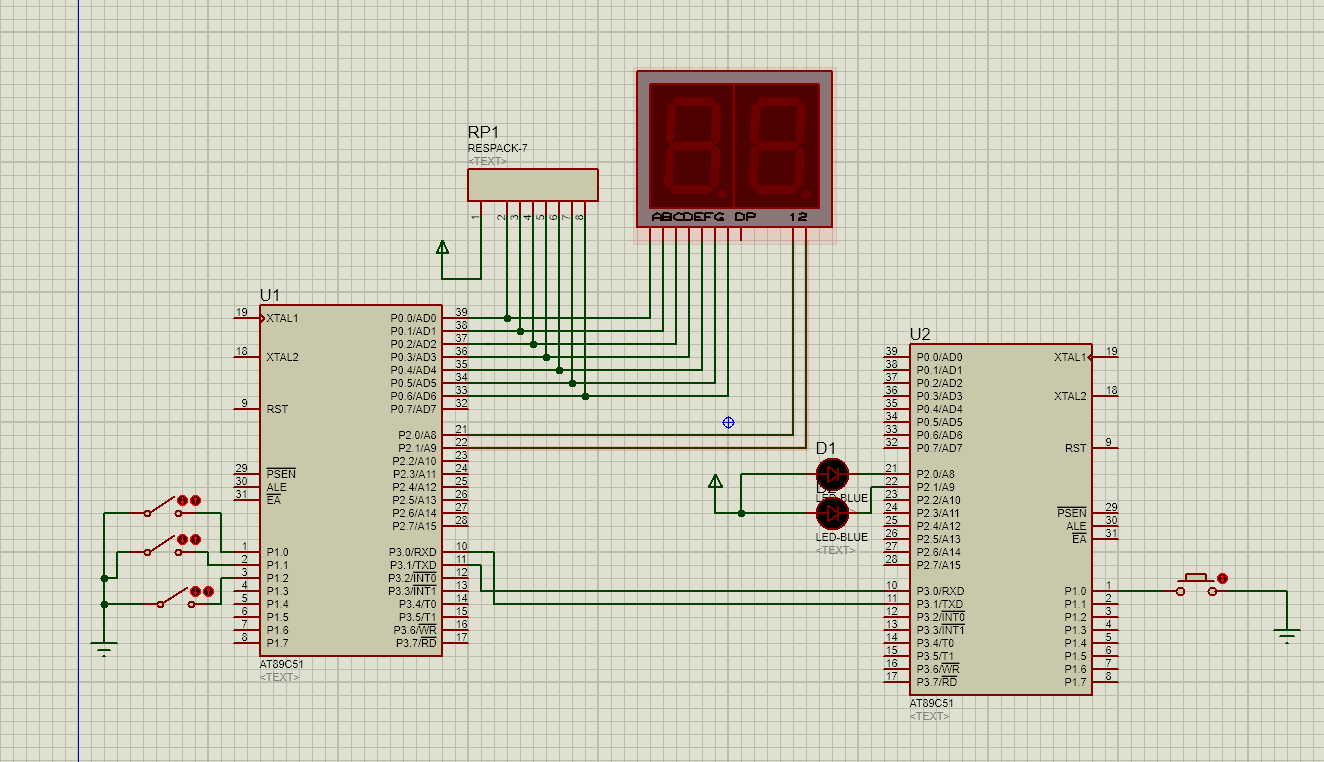
roteus 虚拟仿真。甲、乙机之间采用方式1双向串行通信，具体要求如下：

（1）甲机的k1按键可通过串行口控制乙机的LED1点亮、LED2灭，甲机的k2按键控制乙机的LED1灭、LED2点亮，甲机的k3按键控制乙机的LED1和LED2全亮。

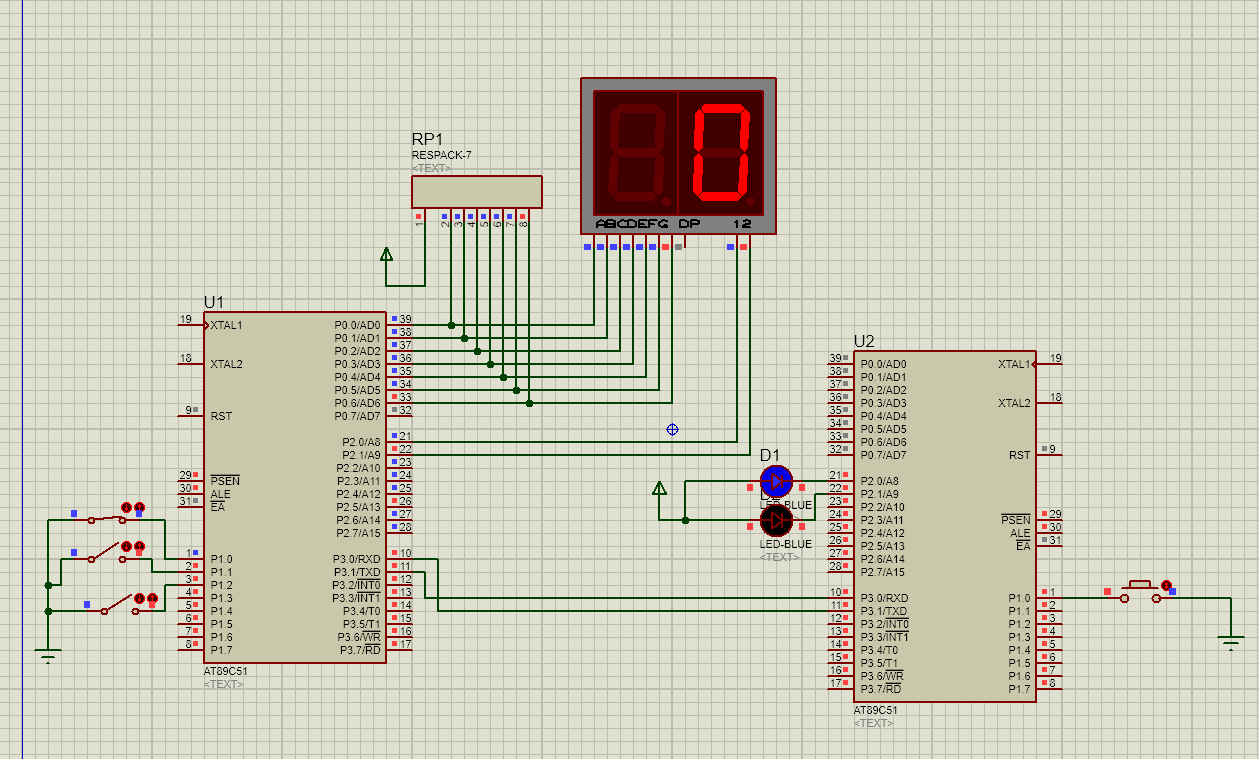
（2)乙机的k4按键可控制串行口向甲机发送 k4按键按下的次数

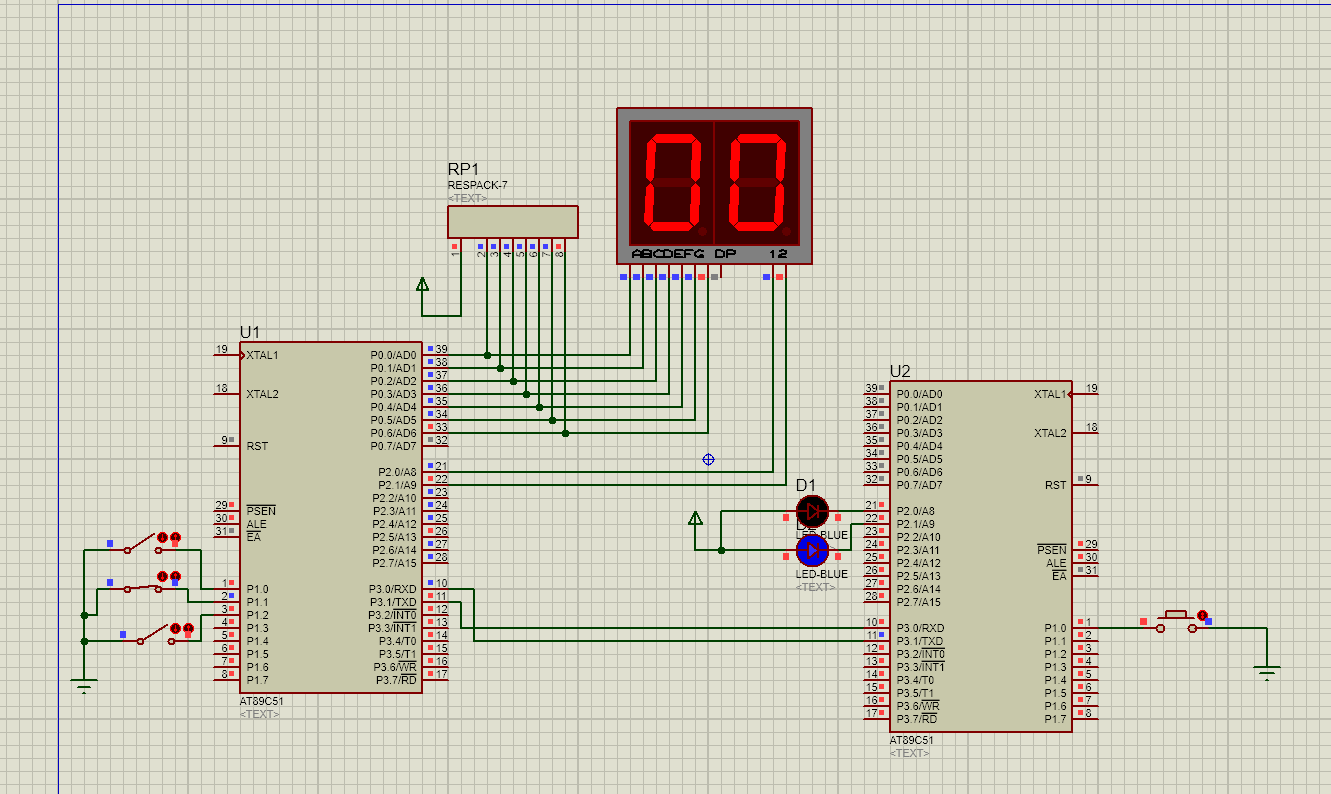
**[四． 实验步骤](#_Toc32164_WPSOffice_Level2)**

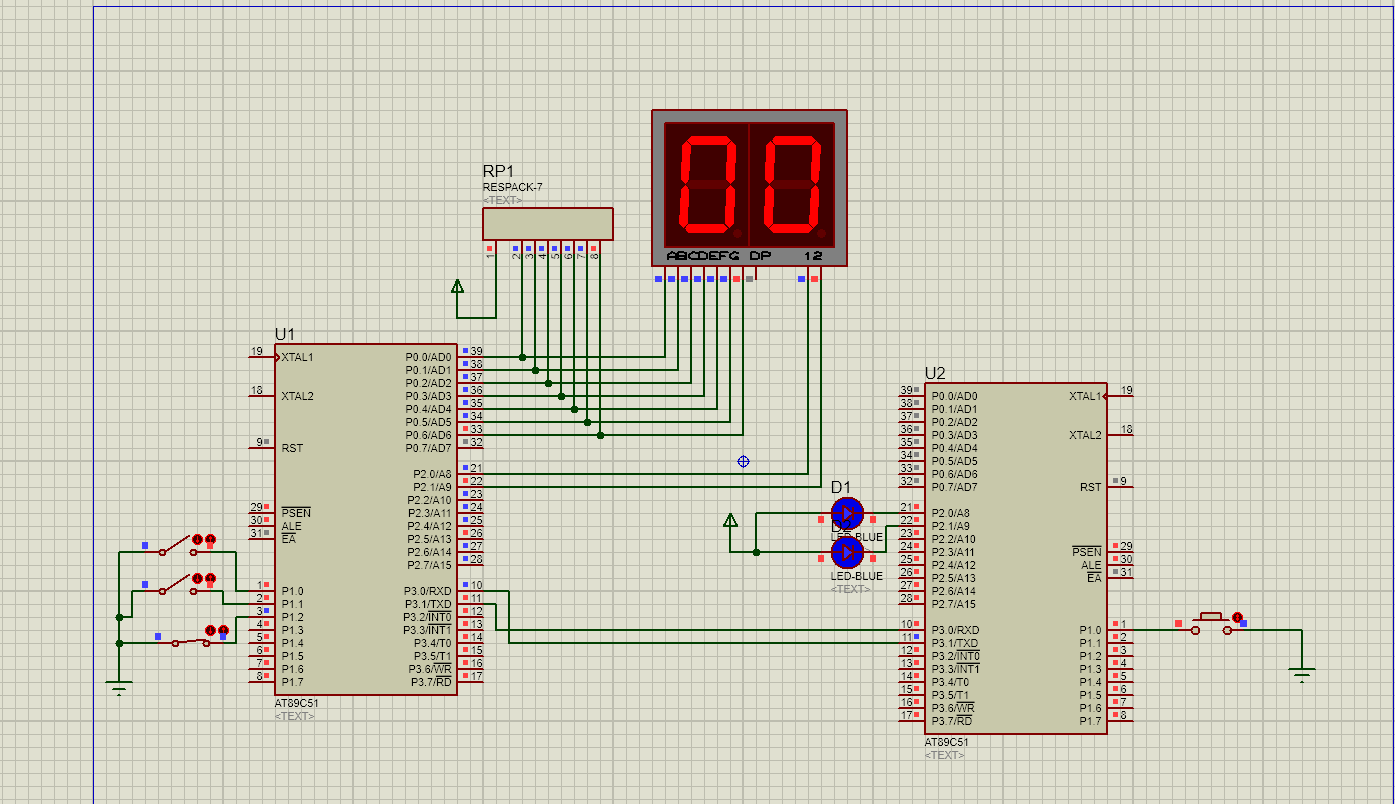
1. **硬件设计**

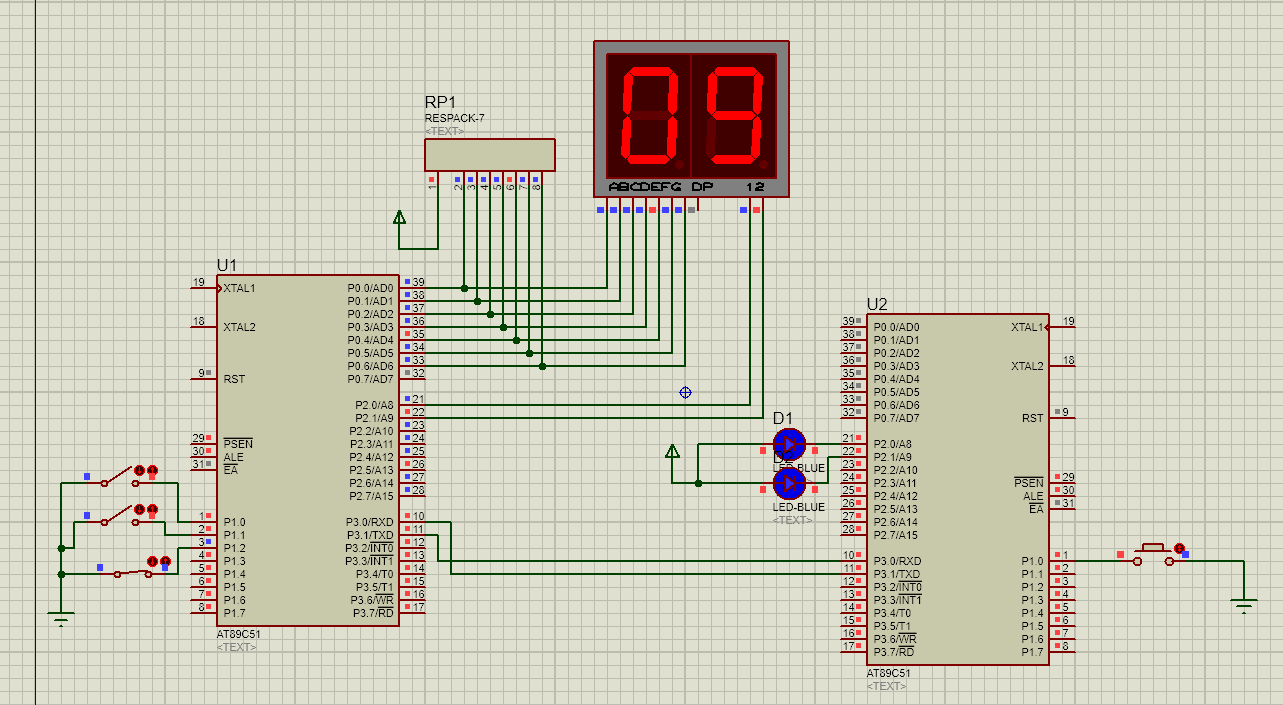


1. **软件设计**
2. //甲机
3. #include<reg51.h>
4. unsigned **char** j=10;
5. **char** sign=1;
6. sbit P10=P1^0;
7. sbit P11=P1^1;
8. sbit P12=P1^2;
9. **void** delay(unsigned **char** k){
10. unsigned **char** i,j,h;
11. **for**(h=0;h<k;h++){
12. **for**(i=0;i<25;i++){
13. **for**(j=0;j<20;j++);
14. }
15. }
16. }
17. **void** twoDigitDisplay(unsigned **char** num,unsigned **char** time,unsigned **char** portNumber){
18. unsigned **char** box[] = {0xc0,0xf9,0xa4,0xb0,0x99,0x92,0xf82,0xf8,0x80,0x90};
19. unsigned **char** k;
20. **if**(portNumber==0){
21. **if**(num<0||num>99||time<0){
22. **for**(k=0;k<40;k++){
23. P2=0x00;
24. }
25. }**else**{
26. **for**(k=0;k<time;k++){
27. unsigned **char** numR = num%10;
28. unsigned **char** numL = num/10;
29. P2=0x11;
30. P0=box[numL];
31. delay(12);
32. P2=0x22;
33. P0=box[numR];
34. delay(12);
35. }
36. }
37. }
38. }
39. **void** main(){
40. unsigned **char** sum=0;
41. TR0=1;
42. EA=1;
43. ET0=1;
44. TMOD=0x21;
45. TH0=0xee;
46. TL0=0x00;
48. **while**(1){
49. **if**(sign==1){
50. //执行数据发送
51. TR1=1;
52. TMOD=0x20;
53. TL1=0xfd;
54. TH1=0xfd;
55. SCON=0x40;
56. PCON=0x00;
57. }
58. **while**(sign==1){
59. //发送数据SBUF
60. **if**(P10==0){SBUF=0xfe;**while**(TI==0);TI=0;**continue**;}
61. **if**(P11==0){SBUF=0xfd;**while**(TI==0);TI=0;**continue**;}
62. **if**(P12==0){SBUF=0xfc;**while**(TI==0);TI=0;**continue**;}
63. SBUF=0xff;
64. }
65. **if**(sign==-1){
66. //执行数据接收
67. TR1=1;
68. TL1=0xfd;
69. TH1=0xfd;
70. SCON=0x50;
71. PCON=0x00;
72. }
73. **while**(sign==-1){
74. //接收数据SBUF
75. **if**(SBUF==0xf0){
76. sum++;
77. twoDigitDisplay(sum,5,0);
78. }
79. **if**(SBUF==0x0f){
80. twoDigitDisplay(sum,5,0);
81. }
82. }
83. }
84. }
85. **void** int0() interrupt 1 {
86. j--;
87. **if**(j==0){
88. TF0=0;
89. TH0=0xee;
90. TL0=0x00;
91. sign=sign\*(-1);
92. j=10;
93. }
94. }
95. //乙机
96. #include<reg51.h>
97. unsigned **char** j=10;
98. **char** sign=1;
99. sbit P10=P1^0;
100. **void** main(){
101. unsigned **char** increment=0;
102. TR0=1;
103. EA=1;
104. ET0=1;
105. TMOD=0x21;
106. TH0=0xee;
107. TL0=0x00;
108. **if**(sign==1){
109. //执行数据接收
110. TR1=1;
111. TL1=0xfd;
112. TH1=0xfd;
113. TMOD=0x20;
114. SCON=0x50;
115. PCON=0x00;
116. }
117. **while**(sign==1){
118. //接收数据SBUF
119. P2=SBUF;
120. }
121. **if**(sign==-1){
122. //执行数据发送
123. TR1=1;
124. TL1=0xfd;
125. TH1=0xfd;
126. SCON=0x40;
127. PCON=0x00;
128. }
129. **while**(sign==-1){
130. //发送数据SBUF
131. **if**(P10==0){
132. SBUF=0xf0;
133. **while**(TI==0);TI=0;
134. }**else**{
135. SBUF=0x0f;
136. **while**(TI==0);TI=0;
137. }
138. }
139. }
140. **void** int0() interrupt 1 {
141. j--;
142. **if**(j==0){
143. TF0=0;
144. TH0=0xee;
145. TL0=0x00;
146. sign=sign\*(-1);
147. j=10;
148. }
149. }
150. **效果图**









**[五． 实验结论及分析](#_Toc29664_WPSOffice_Level2)**

1.结论

通信功能验证： 实验成功实现了甲、乙机之间的双向串行通信，按照要求能够通过串行口控制LED的点亮和熄灭，以及发送按键按下次数的功能。

硬件与软件设计： 硬件和软件设计均按照预期进行，并能够在仿真环境中正常运行。

稳定性与准确性： 在实验过程中，系统表现出稳定性和准确性，能够正确地传输和接收信息。

2.分析

设计优化： 或许有改进空间，比如优化代码以提高效率或改善算法以增加系统的稳定性和响应速度。

学习收获： 通过这个实验，可能学到了如何在单片机中实现串行通信，理解了通信协议和数据传输的基本原理。

问题与挑战： 可能在实验过程中遇到了一些挑战或问题，比如调试困难、理解特定部分的算法等。这些挑战为更深入地理解系统的工作原理提供了机会。

3.总结

总的来说，通过这个实验，对单片机串行通信的原理与应用有了更深入的了解。实验成功验证了系统设计的可行性，但也可能有改进的空间和进一步优化的可能性。

计算机科学与技术系实验报告

课程名称：单片机原理与应用

班级：计科 21-4 姓名：赵泽辉 学号：4202150101443 成绩：

实验项目名称：串行扩展应用

**[一． 实验目的](#_Toc22732_WPSOffice_Level2)**

使用和熟悉利用DS18B20和LED数码管实现单总线温度测量系统

**[二． 实验设备](#_Toc7708_WPSOffice_Level2)**

1.pc机器一台

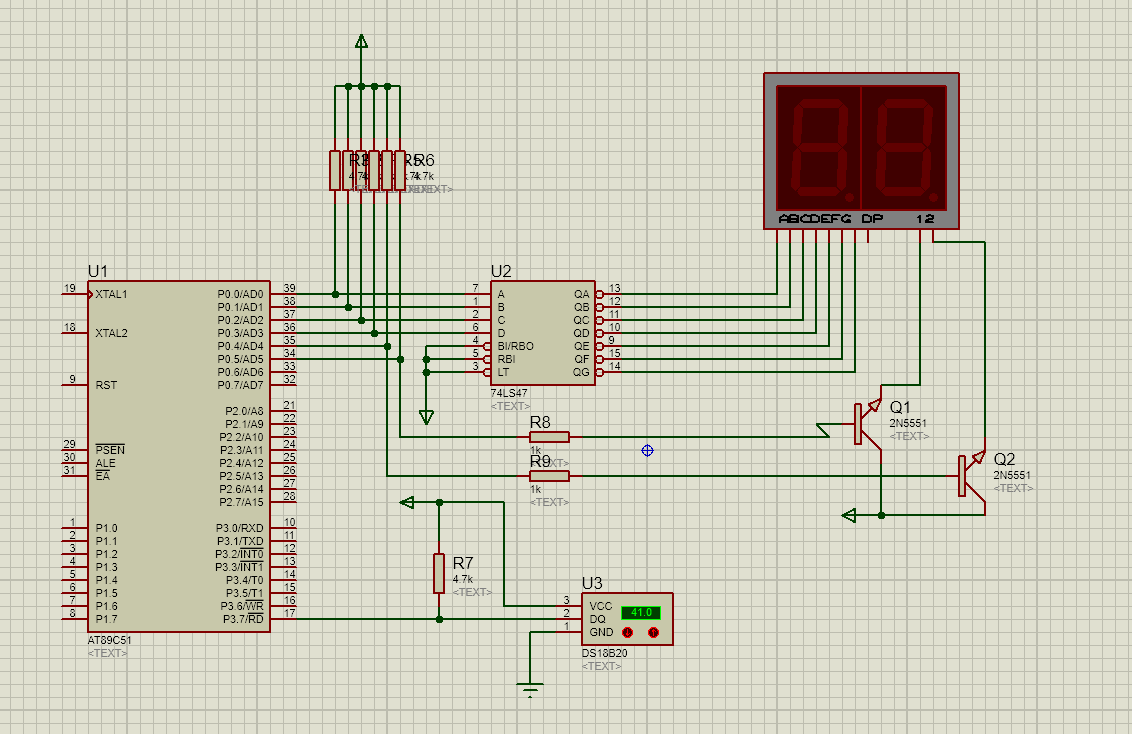
2.keil编译软件与Proteus仿真软件

**[三． 实验内容](#_Toc11783_WPSOffice_Level2)**

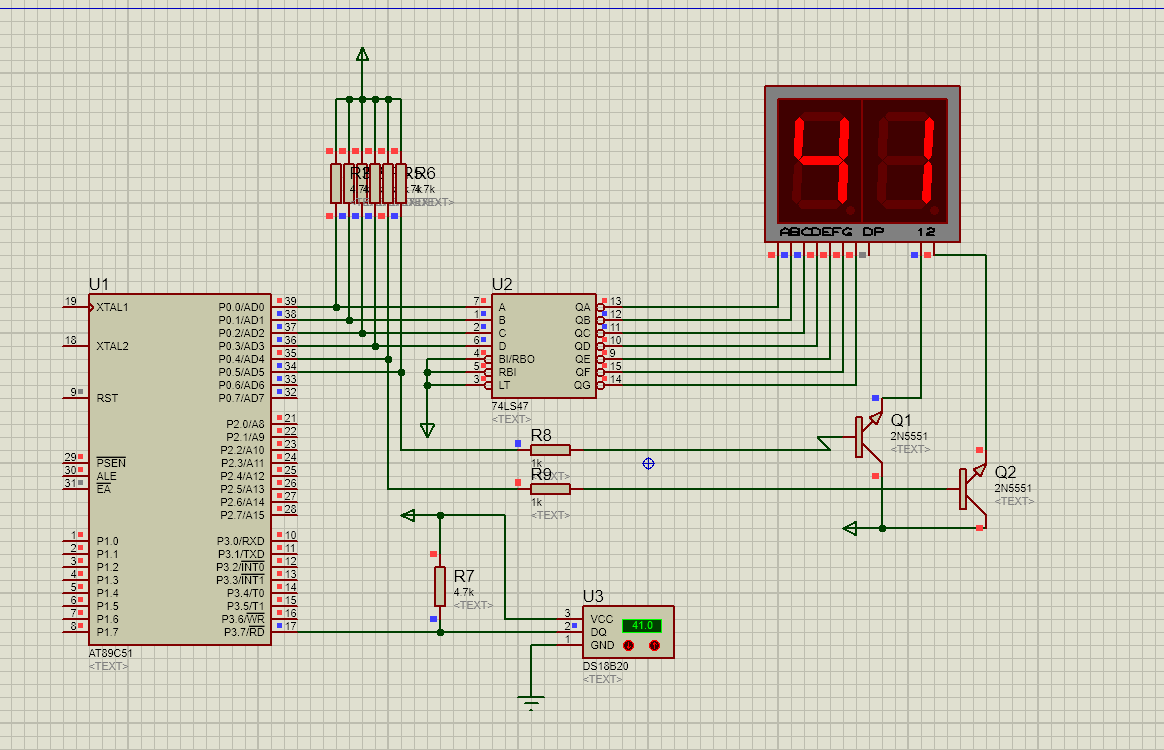
利用DS18B20和LED数码管实现单总线温度测量系统，其仿真原理图如图10-3所示。DS18B20的测量范围是-55℃~128℃。本例由于只接有两个LED数码管，所以显示的数值只能在00~99之间。读者通过本例应掌握DS18B20的特性及单片机VO口实现单总线协议的方法。

**[四． 实验步骤](#_Toc32164_WPSOffice_Level2)**

1. **硬件设计**



1. **软甲设计**
2. #include "reg51.h"
3. #include "intrins.h"
4. #define uchar unsigned char
5. #define uint unsigned int
6. #define out P0
7. sbit smgl=out^4;
8. sbit smg2=out^5;
9. sbit DQ=P3^7;
10. **void** delay5(uchar);
11. **void** init\_ds18b20(**void**);
12. uchar readbyte(**void**);
13. **void** writebyte(uchar);
14. uchar retemp(**void**);
16. **void** main(**void**)
17. {
18. uchar i,temp;
19. delay5(1000);
20. **while**(1)
21. {
22. temp=retemp();
23. **for**(i=0;i<10;i++) //连续扫描LED 数码管10次
24. {
25. out=(temp/10)&0x0f;
26. smgl=0;
27. smg2=1;
28. delay5(1000);
29. out=(temp%10)&0x0f;
30. smgl=1;
31. smg2=0;
32. delay5(1000);
33. }
34. }
35. }
37. **void** delay5(uchar n) //延时 Sus 函数
38. {
39. **do**
40. {
41. \_nop\_();
42. \_nop\_();
43. \_nop\_();
44. n--;
45. }
46. **while**(n);
47. }
49. **void** init\_ds18b20(**void**) //DS18B20 初始化函数
50. {
51. uchar x=0;
52. DQ=0;
53. delay5(120);
54. DQ=1;
55. delay5(16);
56. delay5(80);
57. }
59. uchar readbyte(**void**) //函数功能：读取1字节数据
60. {
61. uchar i=0;
62. uchar date=0;
63. **for**(i=8;i>0;i--)
64. {
65. DQ=0;
66. delay5(1);
67. DQ=1; //15μs内释放总线
68. date>>=1;
69. **if**(DQ)
70. date|=0x80;
71. delay5(11);
72. }
73. **return**(date);
74. }
76. **void** writebyte(uchar dat) //写1字节函数
77. {
78. uchar i=0;
79. **for**(i=8;i>0;i--)
80. {
81. DQ=0;
82. DQ=dat&0x01;
83. delay5(12);
84. DQ=1;
85. dat>>=1;
86. delay5(5);
87. }
88. }
90. uchar retemp(**void**) //读取温度函数
91. {
92. uchar a,b,tt;
93. uint t;
94. init\_ds18b20();
95. writebyte(0xcc);
96. writebyte(0x44);
97. init\_ds18b20();
98. writebyte(0xcc);
99. writebyte(0xbe);
100. a=readbyte();
101. b=readbyte();
102. t=b;
103. t<<=8;
104. t=t|a;
105. tt=t\*0.0625;
106. **return**(tt);
107. }
108. **效果图**



**[五． 实验结论及分析](#_Toc29664_WPSOffice_Level2)**

在这个实验中，我们成功地利用DS18B20和LED数码管搭建了一个单总线温度测量系统。通过这个系统，我们能够在仿真环境下模拟温度传感器的测量，并将结果显示在LED数码管上。

从实验结果来看，我们能够通过单片机VO口实现单总线协议，从而有效地读取DS18B20传感器的数据。在本例中，由于只有两个LED数码管，所以显示的温度数值被限制在00~99之间。然而，DS18B20传感器的测量范围是-55℃到128℃，这意味着我们可以在更广泛的温度范围内进行测量。

在软件设计部分，我们使用了嵌入式C语言编写了程序来实现温度的读取和LED数码管的显示。通过对DS18B20的初始化、写入指令以及读取温度等步骤，我们成功地建立了与传感器的通讯，并将获取的温度数据在数码管上进行了有效的显示。

这个实验让我们深入了解了DS18B20传感器的特性，并学会了如何利用单片机来实现单总线协议，从而建立温度测量系统。通过这样的实践，我们对单片机在实际应用中的功能有了更深入的认识，并提升了我们在嵌入式系统设计方面的能力。