

**本科生实验报告**

**实验课程 单片微机原理及应用**

**学院名称 核技术与自动化工程学院**

**专业名称 测控技术与仪器**

**学生姓名 追梦少年南南**

**学生学号 202006010336**

**指导教师 周伟**

**实验地点 6C701**

**实验成绩**

**二〇二二 年 十一 月 二〇二三 年 一 月**

**填写说明**

1. 适用于本科生所有的实验报告（印制实验报告册除外）；
2. 专业填写为专业全称，有专业方向的用小括号标明；
3. 格式要求：
4. 用A4纸双面打印（封面双面打印）或在A4大小纸上用蓝黑色水笔书写。
5. 打印排版：正文用宋体小四号，1.5倍行距，页边距采取默认形式（上下2.54cm，左右2.54cm，页眉1.5cm，页脚1.75cm）。字符间距为默认值（缩放100%，间距：标准）；页码用小五号字底端居中。
6. 具体要求：

**题目**（二号黑体居中）；

**摘要**（“摘要”二字用小二号黑体居中，隔行书写摘要的文字部分，小4号宋体）；

**关键词**（隔行顶格书写“关键词”三字，提炼3-5个关键词，用分号隔开，小4号黑体)；

正文部分采用三级标题；

**第1章** ××(小二号黑体居中，段前0.5行)

**1.1** ×××××小三号黑体×××××（段前、段后0.5行）

**1.1.1**小四号黑体（段前、段后0.5行）

**参考文献**（黑体小二号居中，段前0.5行），参考文献用五号宋体，参照《参考文献著录规则（GB/T 7714－2005）》。

**实验一 数码管与矩阵按键**

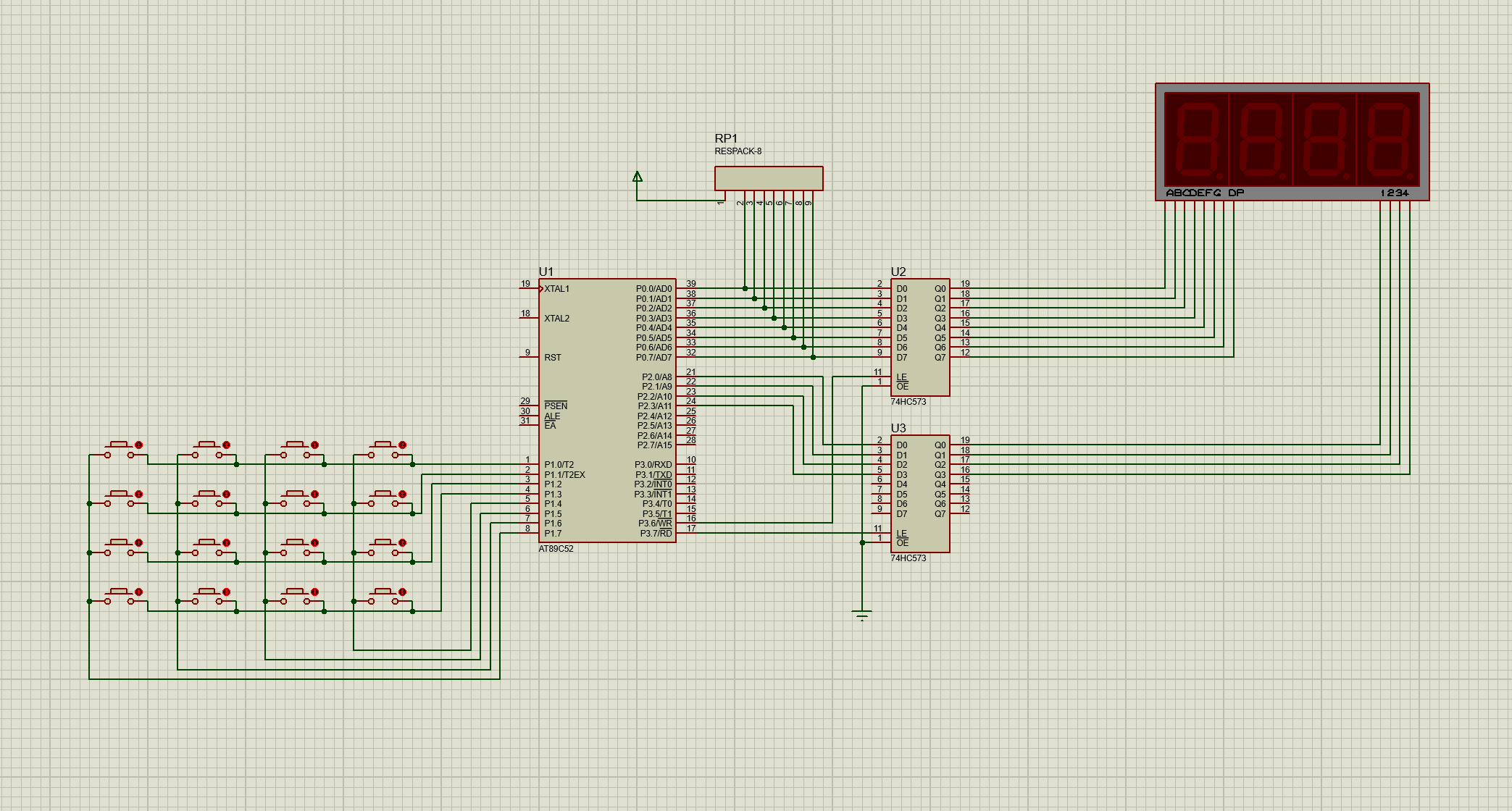
**一．实验要求**

使用矩阵按键输入数据，将结果输出到数码管上，并打开开关使数据流动。

**二．实验目的**

熟练掌握数码管、定时器和矩阵键盘的工作原理及代码写法。

**三．硬件结构电路图**



**四．实验程序**

**(1)SMG.c:**

1. #include <STC89C5xRC.H>

2. #include "SMG.h"

3. unsigned char SMG[]={ //

4. // 0 1 2 3 4 5 6 7 8 9 A B C D E F

5. 0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F,0x77,0x7C,0x39,0x5E,0x79,0x71,

6. //black - H J K L N o P U t G Q r M y

7. 0x00,0x40,0x76,0x1E,0x70,0x38,0x37,0x5C,0x73,0x3E,0x78,0x3d,0x67,0x50,0x37,0x6e,

8. 0xBF,0x86,0xDB,0xCF,0xE6,0xED,0xFD,0x87,0xFF,0xEF,0x46}; //0. 1. 2. 3. 4. 5. 6. 7. 8. 9. -1

9. unsigned char T\_COM[4]={0x01,0x02,0x04,0x08};

10. unsigned char mo1[4]={0,0,0,0};

11. unsigned char mo2[4]={16,16,16,16};

12. unsigned char \*SMG\_cun=mo1;

13. unsigned char p=0;

14. void SMG\_dis()

15. {

16. P2=0x00;

17. P3|=0x80;

18. P3&=0x7f;

19.

20. P0=~SMG[SMG\_cun[p]];

21. P3|=0x40;

22. P3&=0xbf;

23.

24. P2=T\_COM[p];

25. P3|=0x80;

26. P3&=0x7f;

27. p++;

28. if(p>=4)

29. {

30. p=0;

31. }

32. }

**(2)SMG.h:**

1. #ifndef \_\_SMG\_H\_\_

2. #define \_\_SMG\_H\_\_

3. extern unsigned char \*SMG\_cun;

4. extern unsigned char mo1[];

5. extern unsigned char mo2[];

6. void SMG\_dis();

7. #endif

**(3)button.c:**

1. #include <STC89C5xRC.H>

2. #include "button.h"

3. unsigned char buff[4][4]=

4. {

5. 4,5,6,7,

6. 8,9,10,11,

7. 12,13,14,15,

8. 16,17,18,19,

9. };

10. unsigned char C;

11. unsigned char R;

12. unsigned char key\_s;

13. unsigned char key\_v;

14. void button\_pad()

15. {

16. P1=0x0f; //0000 1111; 0000 1110 ;1111 0010 &0000 1111

17. if(~P1&0x0f)

18. {

19. if(key\_s<255)

20. {key\_s++;}

21.

22. if(P13==0)

23. C=0;

24. if(P12==0)

25. C=1;

26. if(P11==0)

27. C=2;

28. if(P10==0)

29. C=3;

30.

31. P1=0xf0;//1111 0000

32. if(P14==0)

33. R=0;

34. if(P15==0)

35. R=1;

36. if(P16==0)

37. R=2;

38. if(P17==0)

39. R=3;

40.

41. key\_v=buff[3-C][3-R];

42. }

43. else

44. {

45. key\_s=0;

46. key\_v=0;

47. }

48. }

49.

**(4)button.h:**

1. #ifndef \_\_BUTTON\_H\_\_

2. #define \_\_BUTTON\_H\_\_

3. void button\_pad();

4. extern unsigned char key\_s;

5. extern unsigned char key\_v;

6. #endif

**(5)main.c:**

1. #include <STC89C5xRC.H>

2. #include "SMG.h"

3. #include "button.h"

4. unsigned int t10ms=0;

5. unsigned char i=0;

6. unsigned char statue=0;

7. bit control=0;

8. void Timer0Init(void) //10毫秒@12.000MHz

9. {

10. AUXR |= 0x80; //定时器时钟1T模式

11. TMOD &= 0xF0; //设置定时器模式

12. TL0 = 0x20; //设置定时初值

13. TH0 = 0xD1; //设置定时初值

14. TF0 = 0; //清除TF0标志

15. TR0 = 1; //定时器0开始计时

16. ET0=1;

17. EA=1;

18. }

19. void timer0() interrupt 1

20. {

21. SMG\_dis();

22. button\_pad();

23. t10ms++;

24. }

25. unsigned char status=0;

26. void main()

27. {

28. Timer0Init();

29. while(1)

30. {

31. if(control==1)

32. {

33. if(t10ms>100)

34. {

35. t10ms=0;

36. mo2[3]++;

37. }

38. }

39. if(key\_v==7&&key\_s==2)//0

40. {

41. if(statue==0)

42. {

43. statue=1;

44. mo2[i]=0;

45. i++;

46. }

47. }

48. if(key\_v==11&&key\_s==2)//1

49. {

50. if(statue==0)

51. {

52. statue=1;

53. mo2[i]=1;

54. i++;

55. }

56. }

57. if(key\_v==15&&key\_s==2)//2

58. {

59. if(statue==0)

60. {

61. statue=1;

62. mo2[i]=2;

63. i++;

64. }

65. }

66. if(key\_v==19&&key\_s==2)//3

67. {

68. if(statue==0)

69. {

70. statue=1;

71. mo2[i]=3;

72. i++;

73. }

74. }

75. if(key\_v==6&&key\_s==2)//4

76. {

77. if(statue==0)

78. {

79. statue=1;

80. mo2[i]=4;

81. i++;

82. }

83. }

84. if(key\_v==10&&key\_s==2)//5

85. {

86. if(statue==0)

87. {

88. statue=1;

89. mo2[i]=5;

90. i++;

91. }

92. }

93. if(key\_v==14&&key\_s==2)//6

94. {

95. if(statue==0)

96. {

97. statue=1;

98. mo2[i]=6;

99. i++;

100. }

101. }

102. if(key\_v==18&&key\_s==2)//7

103. {

104. if(statue==0)

105. {

106. statue=1;

107. mo2[i]=7;

108. i++;

109. }

110. }

111. if(key\_v==5&&key\_s==2)//8

112. {

113. if(statue==0)

114. {

115. statue=1;

116. mo2[i]=8;

117. i++;

118. }

119. }

120. if(key\_v==9&&key\_s==2)//9

121. {

122. if(statue==0)

123. {

124. statue=1;

125. mo2[i]=9;

126. i++;

127. }

128. }

129. if(key\_v==13&&key\_s==2)//input

130. {

131. SMG\_cun=mo2;

132. }

133. if(key\_v==17&&key\_s==2)//start

134. {

135. control=1;

136. }

137. if(key\_v==4&&key\_s==2)//stop

138. {

139. control=0;

140. }

141. if(P1==0x0f)

142. {

143. statue=0;

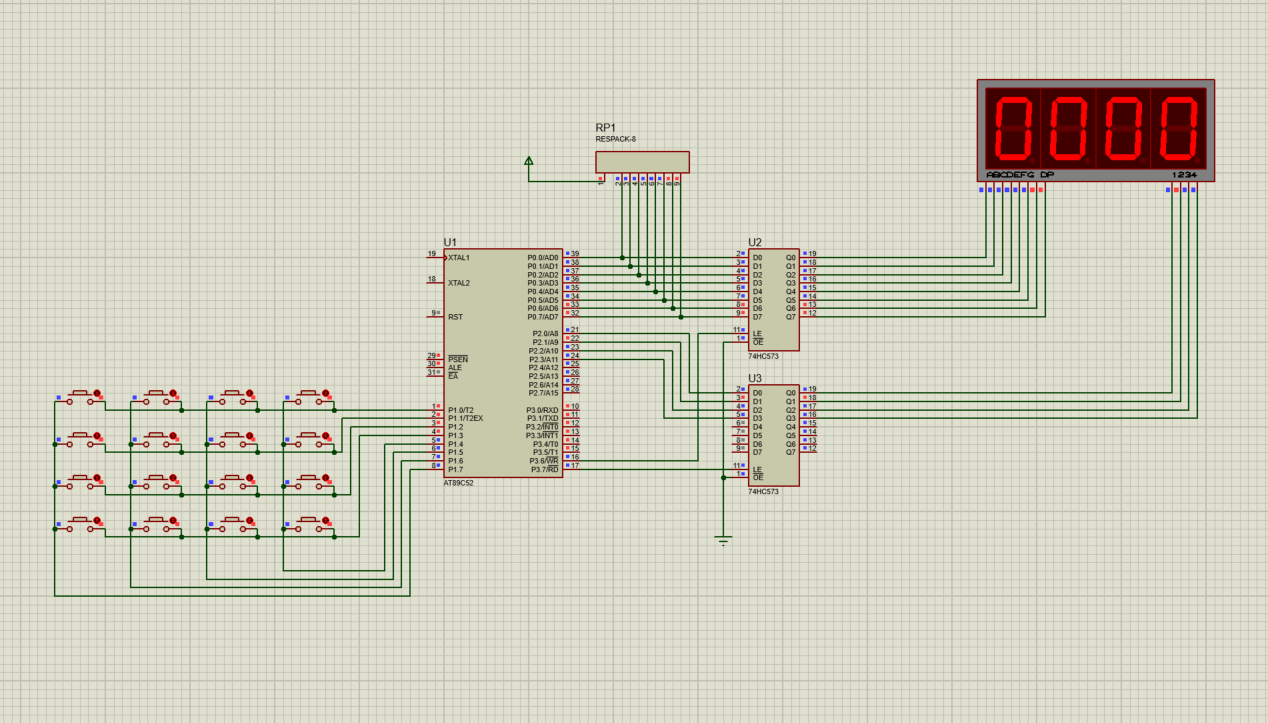
144. }

145. }

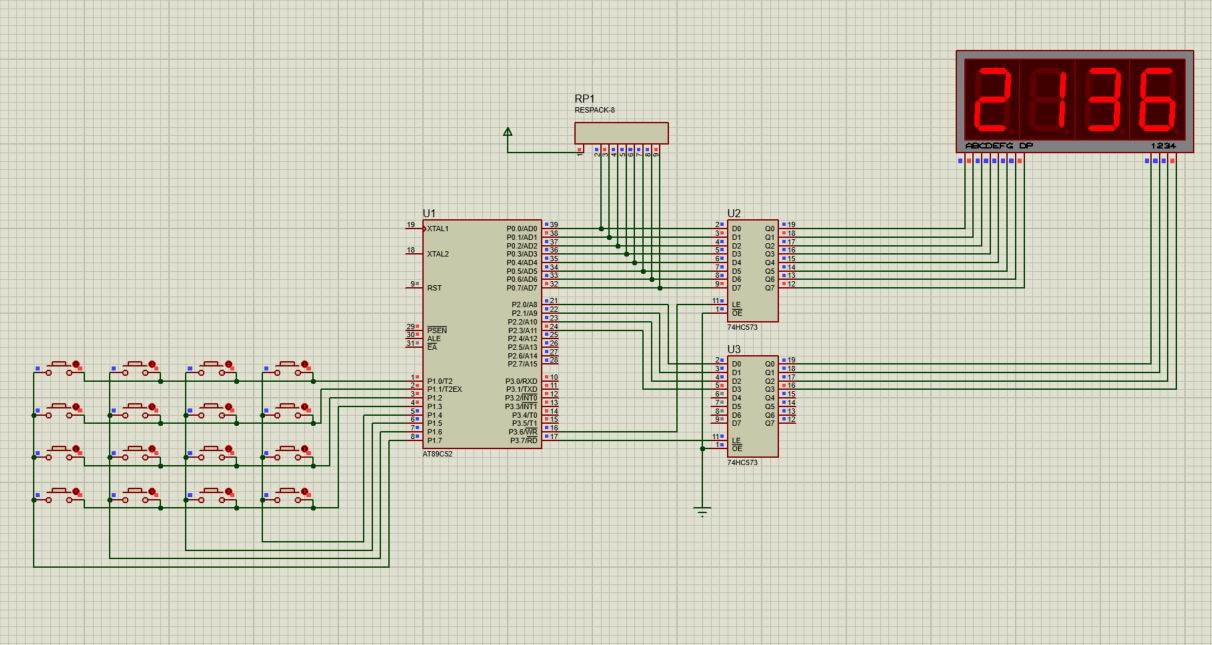
146. }

**五．程序调试及实验结果**

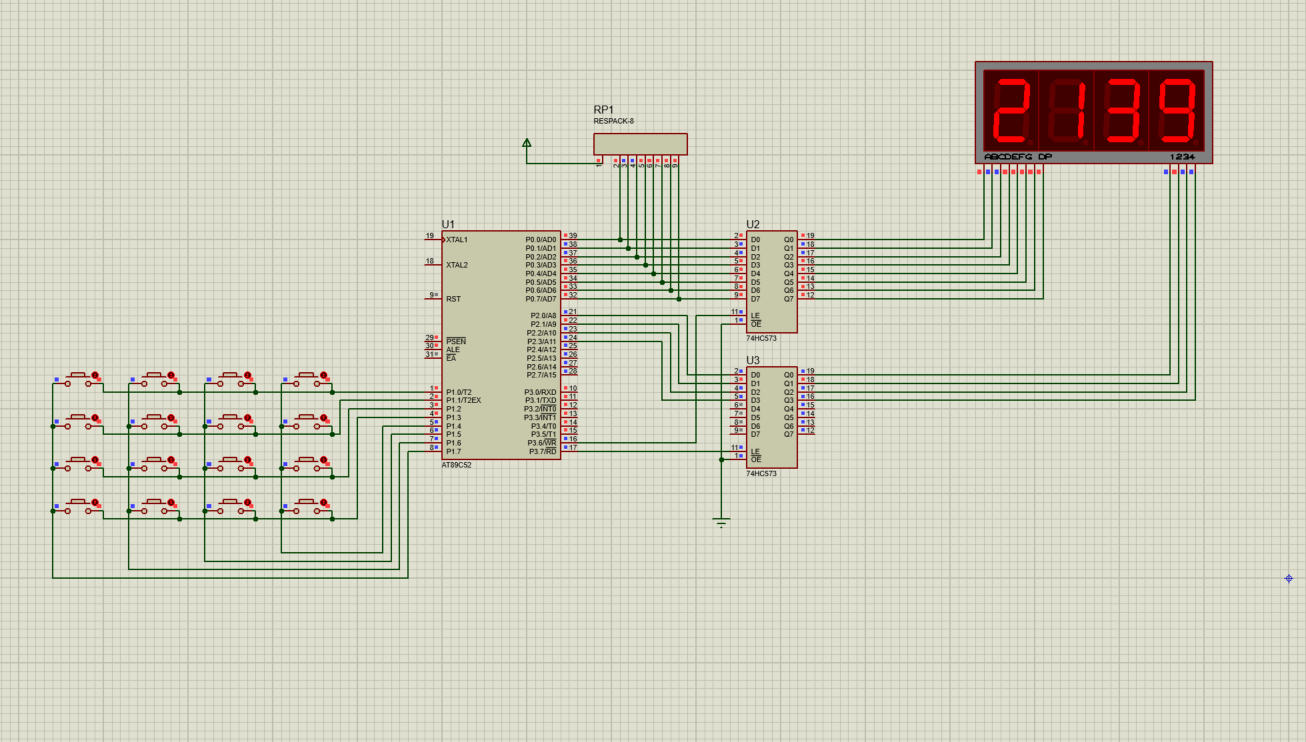
**初始化阶段，显示4个0，按下按键后进入输入阶段。**



**进入输入阶段后，输入四个数字。**



**按下计时按钮，开始计时**



**实验二 TLC549实现ADC采集**

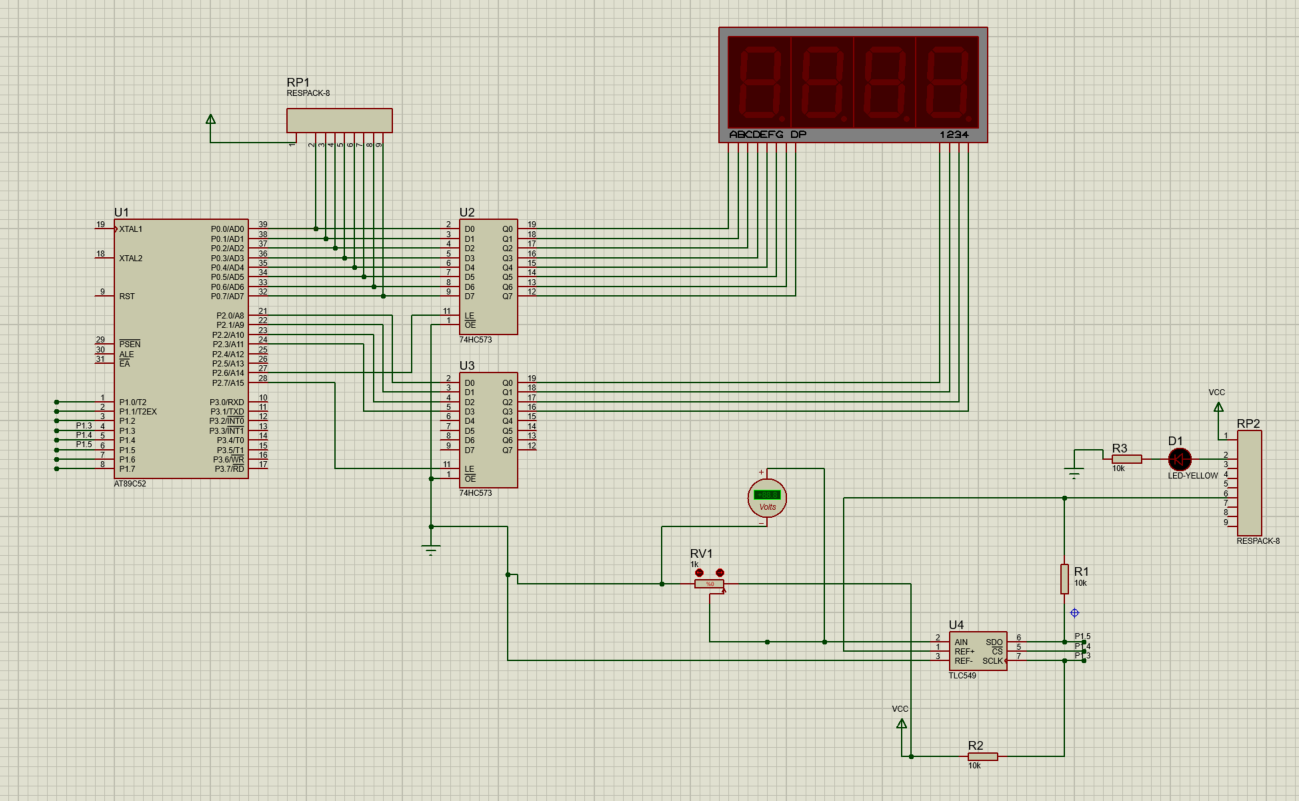
**一．实验要求**

使用TLC549芯片将ADC数据进行采集，将结果显示在数码管上。

**二．实验目的**

掌握51单片机、TLC549芯片原理及程序编写。

**三．硬件结构电路图**



**四．实验程序**

**(1)SMG.c:**

1. #include <STC89C5xRC.H>

2. #include "SMG.h"

3. unsigned char SMG[]={ //

4. // 0 1 2 3 4 5 6 7 8 9 A B C D E F

5. 0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F,0x77,0x7C,0x39,0x5E,0x79,0x71,

6. //black - H J K L N o P U t G Q r M y

7. 0x00,0x40,0x76,0x1E,0x70,0x38,0x37,0x5C,0x73,0x3E,0x78,0x3d,0x67,0x50,0x37,0x6e,

8. 0xBF,0x86,0xDB,0xCF,0xE6,0xED,0xFD,0x87,0xFF,0xEF,0x46}; //0. 1. 2. 3. 4. 5. 6. 7. 8. 9. -1

9. unsigned char T\_COM[4]={0x01,0x02,0x04,0x08};

10. unsigned char mo1[4]={16,0,0,0};

11. unsigned char mo2[4]={16,0,0,0};

12. unsigned char mo3[4]={16,0,0,0};

13. unsigned char \*SMG\_cun=mo2;

14. unsigned char p=0;

15. void SMG\_dis()

16. {

17. P2=0x00;

18. P2|=0x80;

19. P2&=0x7f;

20.

21. P0=~SMG[SMG\_cun[p]];

22. P2|=0x40;

23. P2&=0xbf;

24.

25. P2=T\_COM[p];

26. P2|=0x80;

27. P2&=0x7f;

28. p++;

29. if(p>=4)

30. {

31. p=0;

32. }

33. }

34.

**(2)SMG.h:**

1. #ifndef \_\_SMG\_H\_\_

2. #define \_\_SMG\_H\_\_

3. extern unsigned char \*SMG\_cun;

4. extern unsigned char mo1[];

5. extern unsigned char mo2[];

6. extern unsigned char mo3[];

7. void SMG\_dis();

8. #endif

9.

**(3)main.c:**

1. #include <STC89C5xRC.H>

2. #include <SMG.h>

3. #include <intrins.h>

4. sbit DO=P1^5;

5. sbit CS=P1^4;

6. sbit SCLK=P1^3;

7. void Timer0Init(void) //10ms

8. {

9. AUXR |= 0x80;

10. TMOD &= 0xF0;

11. TL0 = 0x20;

12. TH0 = 0xD1;

13. TF0 = 0;

14. TR0 = 1;

15. ET0=1;

16. EA=1;

17. }

18. void delay(unsigned int t)

19. {

20. while(t--);

21. }

22. void timer0() interrupt 1

23. {

24. SMG\_dis();

25. }

26. unsigned int counting(unsigned char a)

27. {

28. float j;

29. j=a\*1.9607843137254;

30. return (unsigned int)j;

31. }

32. unsigned int TLC549\_AD() //TLC549处理

33. {

34. unsigned char i;

35. unsigned int data\_ad=0;

36. CS=1; //初始化，启动

37. SCLK=0;

38. DO=1;

39. CS=0;

40. \_nop\_();

41. for(i=0;i<8;i++) //读取采集数据，读取的是上一次采集数据

42. {

43. SCLK=1;

44. if(DO)data\_ad|=0x01;

45. SCLK=0;

46. data\_ad<<=1;

47. }

48. CS=1;

49. // data\_ad=data\_ad\*(500/256)+0.5; //0.5V进行四舍五入补偿

50. return data\_ad;

51. }

52. unsigned char ad549\_read()

53. {

54. unsigned int i;

55. unsigned char k=0;

56. DO=1;

57. for(i=0;i<8;i++)

58. {

59. k=k<<1;

60. if(DO)

61. k++;

62. SCLK=1;

63. SCLK=0;

64. delay(10);

65.

66. }

67. return k;

68. }

69. unsigned char ad549\_export()

70. {

71. unsigned char date;

72. CS=0;

73. date=ad549\_read();

74. CS=1;

75. delay(20);

76. return date;

77. }

78. void main()

79. {

80. unsigned char i;

81. unsigned int count;

82. Timer0Init();

83. while(1)

84. {

85. count=counting(ad549\_export());

86. mo2[3]=count%10;

87. mo2[2]=count/10%10;

88. mo2[1]=count/100+32;

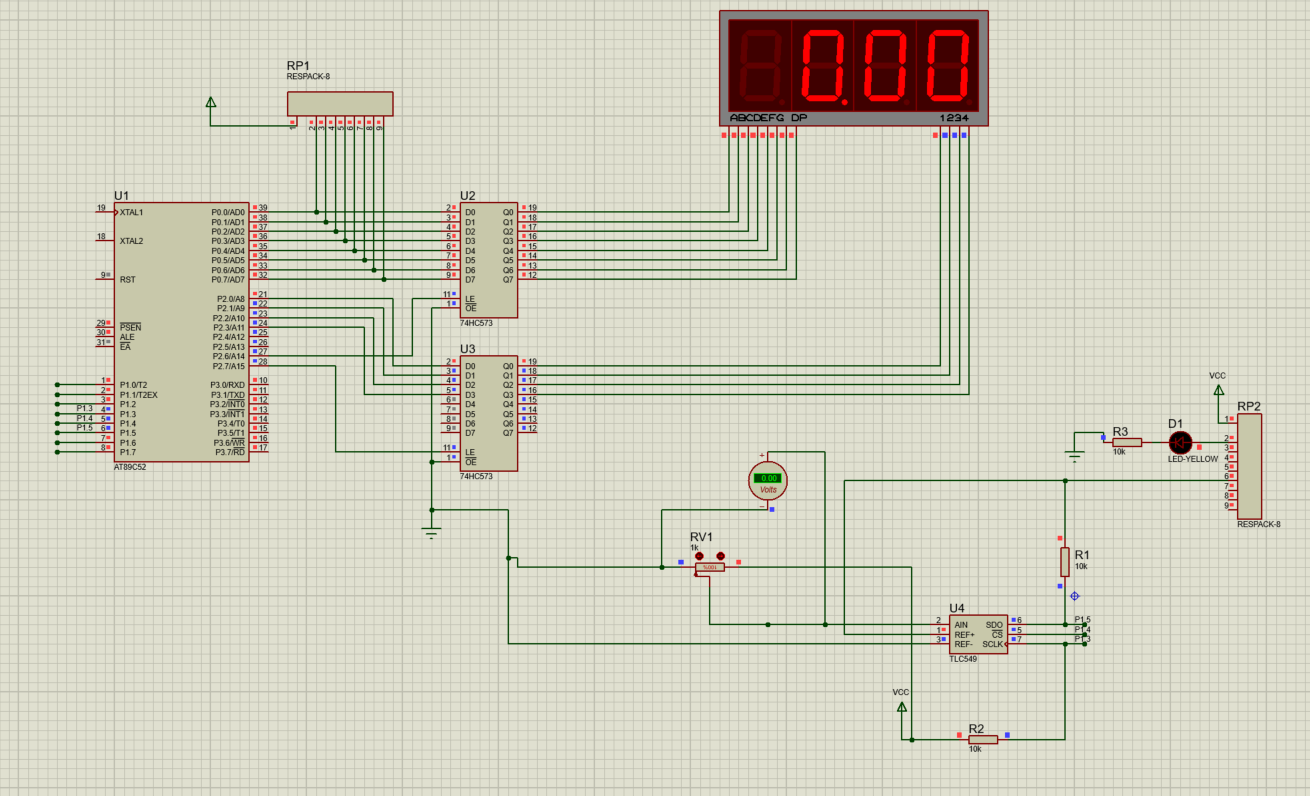
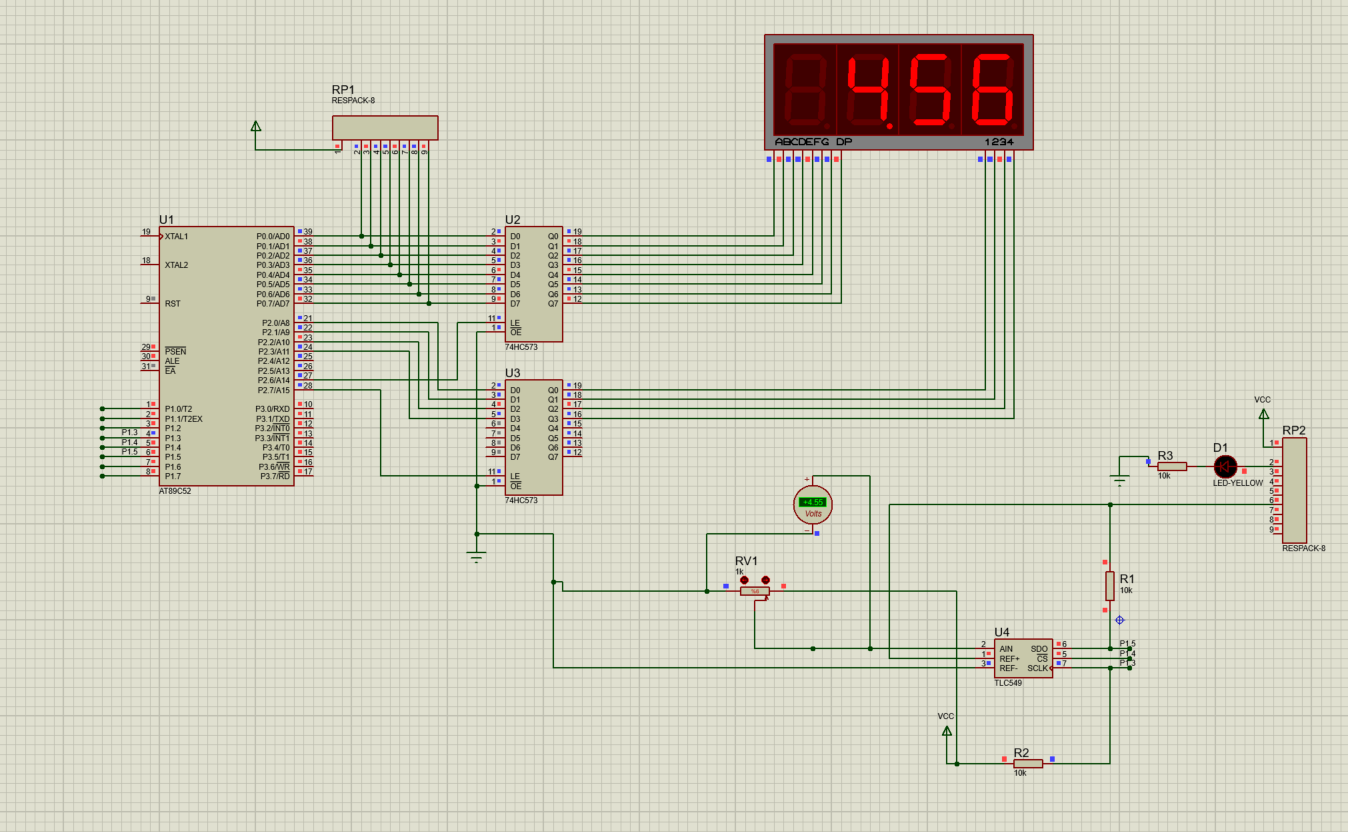
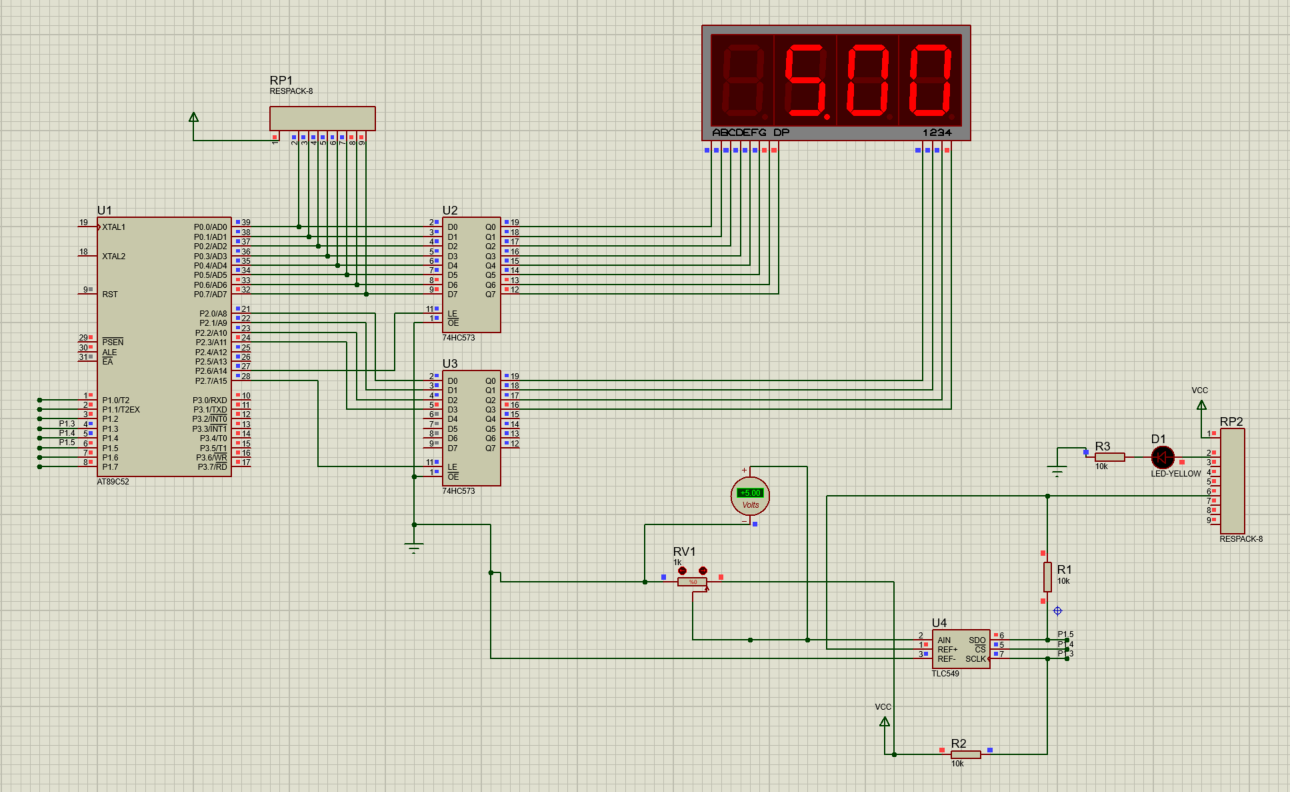
89. }

90. }

91.

**五．程序调试及实验结果**

**可以看出，当我们调整滑动变阻器到任意阻值时，数码管会显示对应的数值。**



**实验三 TLC5615实现DAC转换**

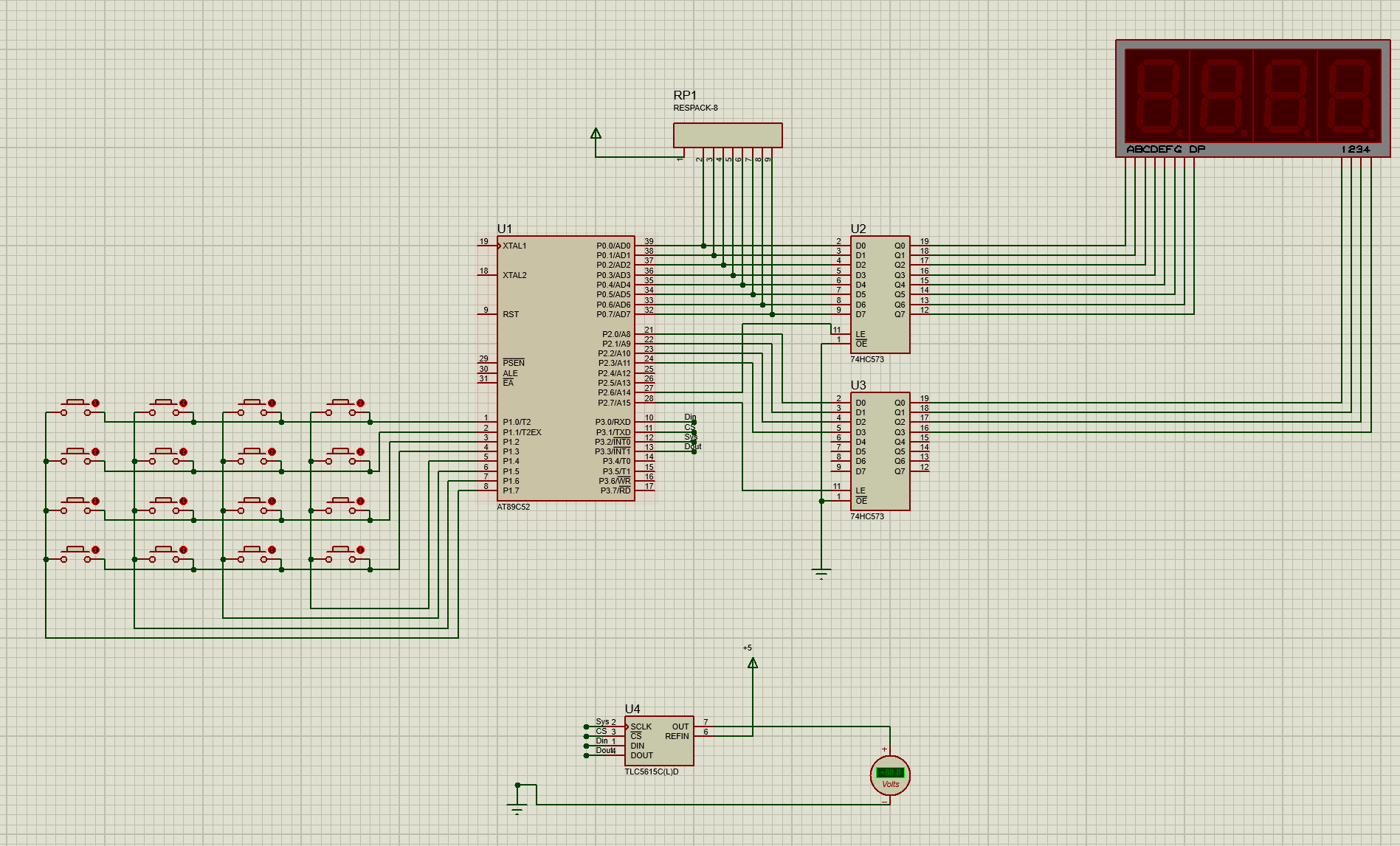
**一．实验要求**

使用矩阵键盘输入想要得到的电压值，使用TLC5615进行DAC转换，最后将结果显示到数码管上。

**二．实验目的**

掌握51单片机、TLC5615芯片原理及程序编写。

**三．硬件结构电路图**



**四．实验程序**

**（1）SMG.c:**

1. #include <STC89C5xRC.H>

2. #include "SMG.h"

3. unsigned char SMG[]={ //

4. // 0 1 2 3 4 5 6 7 8 9 A B C D E F

5. 0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F,0x77,0x7C,0x39,0x5E,0x79,0x71,

6. //black - H J K L N o P U t G Q r M y

7. 0x00,0x40,0x76,0x1E,0x70,0x38,0x37,0x5C,0x73,0x3E,0x78,0x3d,0x67,0x50,0x37,0x6e,

8. 0xBF,0x86,0xDB,0xCF,0xE6,0xED,0xFD,0x87,0xFF,0xEF,0x46}; //0. 1. 2. 3. 4. 5. 6. 7. 8. 9. -1

9. unsigned char T\_COM[4]={0x01,0x02,0x04,0x08};

10. unsigned char mo1[4]={16,0,0,0};

11. unsigned char mo2[4]={0,0,0,0};

12. unsigned char mo3[4]={16,0,0,0};

13. unsigned char \*SMG\_cun=mo1;

14. unsigned char p=0;

15. void SMG\_dis()

16. {

17. P2=0x00;

18. P2|=0x80;

19. P2&=0x7f;

20.

21. P0=~SMG[SMG\_cun[p]];

22. P2|=0x40;

23. P2&=0xbf;

24.

25. P2=T\_COM[p];

26. P2|=0x80;

27. P2&=0x7f;

28. p++;

29. if(p>=4)

30. {

31. p=0;

32. }

33. }

34.

**(2)SMG.h:**

1. #ifndef \_\_SMG\_H\_\_

2. #define \_\_SMG\_H\_\_

3. extern unsigned char \*SMG\_cun;

4. extern unsigned char mo1[];

5. extern unsigned char mo2[];

6. extern unsigned char mo3[];

7. void SMG\_dis();

8. #endif

9.

**(3)button.c:**

1. #include <STC89C5xRC.H>

2. #include <button.h>

3. unsigned char buff[4][4]=

4. {

5. 4,5,6,7,

6. 8,9,10,11,

7. 12,13,14,15,

8. 16,17,18,19,

9. };

10. unsigned char C;

11. unsigned char R;

12. unsigned char key\_s;

13. unsigned char key\_v;

14. void button()

15. {

16. P1=0x0f; //0000 1111; 0000 1110 ;1111 0010 &0000 1111

17. if(~P1&0x0f)

18. {

19. if(key\_s<255)

20. {key\_s++;}

21.

22. if(P13==0)

23. C=0;

24. if(P12==0)

25. C=1;

26. if(P11==0)

27. C=2;

28. if(P10==0)

29. C=3;

30.

31. P1=0xf0;//1111 0000

32. if(P14==0)

33. R=0;

34. if(P15==0)

35. R=1;

36. if(P16==0)

37. R=2;

38. if(P17==0)

39. R=3;

40.

41. key\_v=buff[3-C][3-R];

42. }

43. else

44. {

45. key\_s=0;

46. key\_v=0;

47. }

48. }

**(4)button.h:**

1. #ifndef \_\_BUTTON\_H\_\_

2. #define \_\_BUTTON\_H\_\_

3. extern unsigned char key\_v;//状态

4. extern unsigned char key\_s;//键值

5. void button();

6. #endif

**(5)main.c:**

1. #include <STC89C5xRC.H>

2. #include <SMG.h>

3. #include <button.h>

4. sbit din=P3^0;

5. sbit sclk=P3^2;

6. sbit cs=P3^1;

7. sbit dout=P3^3;

8. unsigned int t10ms=0;

9. #define Vref 5

10. float v0=0;

11. float count\_DAC(unsigned char u);

12. unsigned char i=0;

13. unsigned char statue=0;

14. bit control=0;

15. void Timer0Init(void) //10ms

16. {

17. AUXR |= 0x80;

18. TMOD &= 0xF0;

19. TL0 = 0x20;

20. TH0 = 0xD1;

21. TF0 = 0;

22. TR0 = 1;

23. ET0=1;

24. EA=1;

25. }

26. float count()

27. {

28. float sum=0;

29. float xianshi=0;

30. sum=mo2[1]\*100+mo2[2]\*10+mo2[3];

31. xianshi=(sum/4.096)\*1024;

32. return xianshi;

33. }

34. //float counting()//

35. //{

36. // float sum=0;

37. // float xianshi=0;

38. // sum=mo2[1]\*100+mo2[2]\*10+mo2[3];

39. // xianshi=(sum/500)\*256;

40. // return xianshi;

41. //}

42. void Write5615(unsigned int wdata)

43. {

44. unsigned char i;

45. cs=0;

46. wdata<<=2;

47. for(i=0;i<12;i++)

48. {

49. sclk=0;

50. din=(wdata&0x0800)?1:0;

51. sclk=1;

52. wdata<<=1;

53. }

54. cs=1;

55. }

56. void timer0() interrupt 1

57. {

58. SMG\_dis();

59. t10ms++;

60. button();

61. }

62. unsigned int volt=0;

63. void main()

64. {

65. Timer0Init();

66. while(1)

67. {

68.

69.

70. if(key\_v==7&&key\_s==2)//0

71. {

72. if(statue==0)

73. {

74. statue=1;

75. mo2[i]=0;

76. i++;

77. }

78. }

79. if(key\_v==11&&key\_s==2)//1

80. {

81. if(statue==0)

82. {

83. statue=1;

84. mo2[i]=1;

85. i++;

86. }

87. }

88. if(key\_v==15&&key\_s==2)//2

89. {

90. if(statue==0)

91. {

92. statue=1;

93. mo2[i]=2;

94. i++;

95. }

96. }

97. if(key\_v==19&&key\_s==2)//3

98. {

99. if(statue==0)

100. {

101. statue=1;

102. mo2[i]=3;

103. i++;

104. }

105. }

106. if(key\_v==6&&key\_s==2)//4

107. {

108. if(statue==0)

109. {

110. statue=1;

111. mo2[i]=4;

112. i++;

113. }

114. }

115. if(key\_v==10&&key\_s==2)//5

116. {

117. if(statue==0)

118. {

119. statue=1;

120. mo2[i]=5;

121. i++;

122. }

123. }

124. if(key\_v==14&&key\_s==2)//6

125. {

126. if(statue==0)

127. {

128. statue=1;

129. mo2[i]=6;

130. i++;

131. }

132. }

133. if(key\_v==18&&key\_s==2)//7

134. {

135. if(statue==0)

136. {

137. statue=1;

138. mo2[i]=7;

139. i++;

140. }

141. }

142. if(key\_v==5&&key\_s==2)//8

143. {

144. if(statue==0)

145. {

146. statue=1;

147. mo2[i]=8;

148. i++;

149. }

150. }

151. if(key\_v==9&&key\_s==2)//9

152. {

153. if(statue==0)

154. {

155. statue=1;

156. mo2[i]=9;

157. i++;

158. }

159. }

160. if(key\_v==13&&key\_s==2)//input

161. {

162. SMG\_cun=mo2;

163. }

164. if(key\_v==17&&key\_s==2)//start

165. {

166. SMG\_cun=mo3;

167. mo3[1]=mo2[1]+32;

168. mo3[2]=mo2[2];

169. mo3[3]=mo2[3];

170. }

171. if(key\_v==4&&key\_s==2)//stop

172. {

173. control=0;

174. }

175. if(P1==0x0f)

176. {

177. statue=0;

178. }

179. volt=mo2[1]\*100+mo2[2]\*10+mo2[3];

180. Write5615(volt);

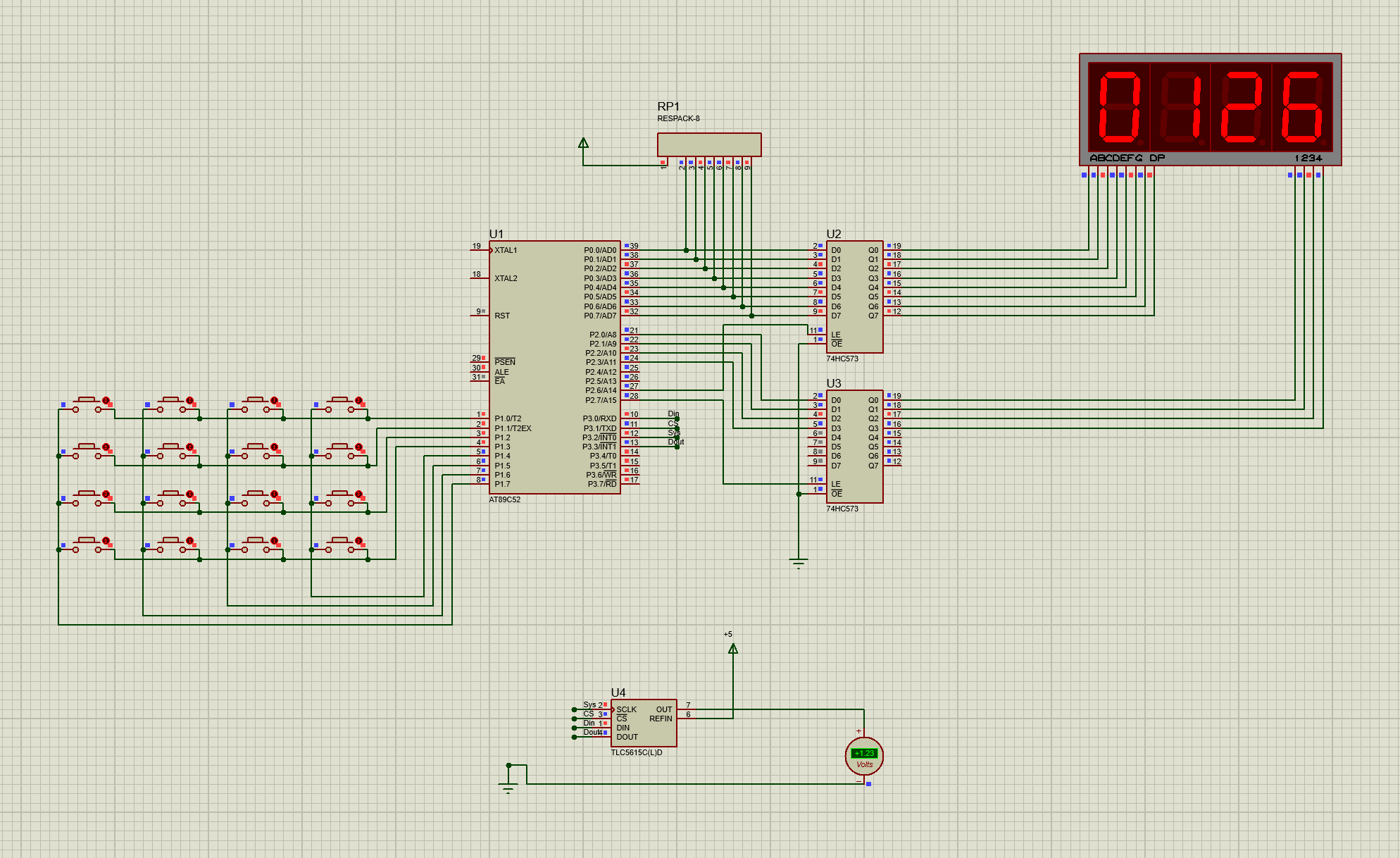
181.

182. }

183. }

**五．程序调试及实验结果**

**输入：假设我要得到1.26V电压**



**输出：可以得到1.23V电压（有些许误差）**

