## **51单片机实现三位十进制数加减乘除运算**

### **一.题目**

51单片机IO接口作业

请将附件给出的Proteus图用51单片机完成一个计算器功能。

1、显示采用动态分时8位共阳数码管输出。

2、采用4\*4矩阵键盘输入，键盘上已经标识对应键。

3、完成三位十进制数的加、减、乘、除运算。

4、开机显示最右边数码管显示‘0’，输入按键值后依次左移。"+—\*/"数码管显示分别“A B C D”。

按要求上交实验报告（加、减、乘、除的对应的Proteus仿真截图一定要有，下载到开发板图也必须有）。

### **实验思路**

程序分为以下几步：

1. 数码管显示模块，数码管上显示数字

2. 矩阵键盘模块，用户按下按键返回相应的按键值

3. 主函数，实现加减乘除运算，偷了点懒，其实加减乘除运算也可以独立出来做成一个模块

### **三.程序代码**

#include<reg52.h>

//共阳极数码管

unsigned char code segCodeTable[16] = {0xc0,0xf9,0xa4,0xb0,

0x99,0x92,0x82,0xf8,

0x80,0x90,0x88,0x83,

0xc6,0xa1,0x86,0x8e};

//位码表

unsigned char code posCodeTable[8] = {

0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};

unsigned char code key\_value[16]={0xD7,0xEB,0xDB,0xBB,0xED,0xDD,0xBD,0xEE,0xDE,0xBE,0x77,0x7B,0x7D,0x7E,

0xB7,0xE7};//键盘标识对应的键盘值

void dis\_seg7(unsigned long num,unsigned char dec\_hex);

unsigned char keyscan();

void delayms (void);

unsigned char pre\_P1;

void main()

{

unsigned char key0=0,key1=0,key2=0;

unsigned char value = 0,j = 0;

unsigned char state = 0;

unsigned char dec\_hex = 10;

unsigned int NUMA = 0,NUMB = 0;

unsigned long dis\_num = 0;

while(1)

{

value = keyscan();

//if((pre\_P1 == 0xf0)&&(P1 != 0xf0))

//实现数码管显示多位数字

if((pre\_P1 == 0xf0)&&(P1 != 0xf0))

{

if(value >= 0 && value <= 9)

{

dec\_hex = 10;

key2 = key1;

key1 = key0;

key0 = value;

dis\_num = 100\*key2 + 10\* key1 + key0;

}

//按下操作符+-\*/前,保存数码管显示数字dis\_num到NUMA

// key0 = 0;key1 = 0;

//key2 = 0;dis\_num = value;

//上述四条语句的目的:按下操作符显示对应的16进制数；

//实现数码管的清零,即实现120+150,数码管一次显示120;

//A(表示加号);150

else if(value>=10&&value<=13)

{

NUMA = dis\_num;

key0 = 0;

key1 = 0;

key2 = 0;

dis\_num = value;

dec\_hex = 16;

if(value == 10 )

{

state ='+';

}

else if(value == 11)

{

state = '-';

}

else if (value == 12)

{

state = '\*';

}

else if (value == 13)

{

state = '/';

}

}

//按下等号时,保存数码管显示数字dis\_num到NUMB,实现计算

else if(value == 14)

{

dec\_hex = 10;

NUMB = dis\_num;

if(state == '+')

{

dis\_num = NUMA + NUMB;

}

else if(state == '-')

{

dis\_num = NUMA - NUMB;

}

else if(state == '\*')

{

dis\_num = NUMA \* NUMB;

}

else if(state == '/')

{

dis\_num = NUMA / NUMB;

}

}

//数码管清零

else if(value == 15)

{

dec\_hex = 10;

key0 = 0;

key1 = 0;

key2 = 0;

NUMA = 0;

NUMB = 0;

dis\_num = 0;

}

}

pre\_P1 = P1;

dis\_seg7(dis\_num,dec\_hex);

for(j = 0;j < 200;j++);

}

}

//七段数码管显示模块

void dis\_seg7(unsigned long num,unsigned char dec\_hex)

{

char i = 0,count = 8;

unsigned int j =0;

char dis\_data = 0;

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

{

dis\_data = num % dec\_hex;

num = num / dec\_hex;

P2 = segCodeTable[dis\_data];

P3 = posCodeTable[i];

for(j = 0;j < 200;j++);

P2 = 0xff; //清除余晖

if(num == 0)

{

break;

}

}

}

//延时模块

void delayms (void)

{

unsigned char i;

for (i=200;i>0;i--);

}

//按键扫描模块

unsigned char keyscan()

{

unsigned char rowscan,colscan,row\_col,i;

static unsigned char value;

P1 = 0xf0; //P1输出 0xf0

if((P1&0xf0) != 0xf0) //P1读入数据，判断是否有按键按下

{

delayms();//延时消抖

if((P1&0xf0) != 0xf0)

{

//本代码采用逐行扫描的方式来确定按下的按键

rowscan = 0xfe;

while((rowscan&0x10)!=0)

{

P1 = rowscan;

colscan = P1 & 0xf0;

if(colscan != 0xf0)

{

//确定按下按键的行号与列号，查表，返回按键值

row\_col = colscan | (rowscan&0x0f);

for(i = 0;i<16;i++)

{

if(row\_col == key\_value[i])

{

value = i;

}

}

break;

}

else

{

rowscan = (rowscan << 1) | 0x1;

}

}

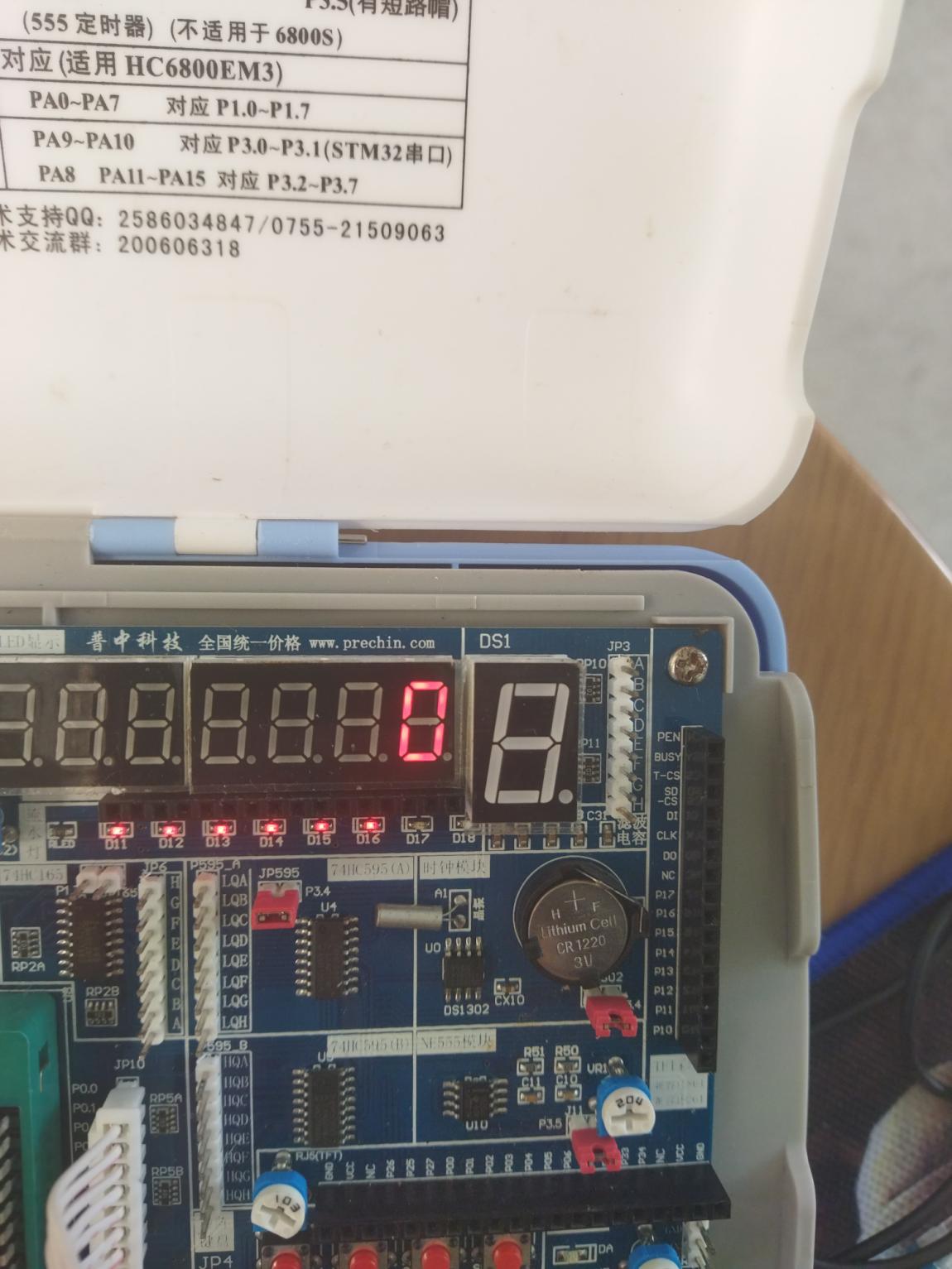
}

}

return value;

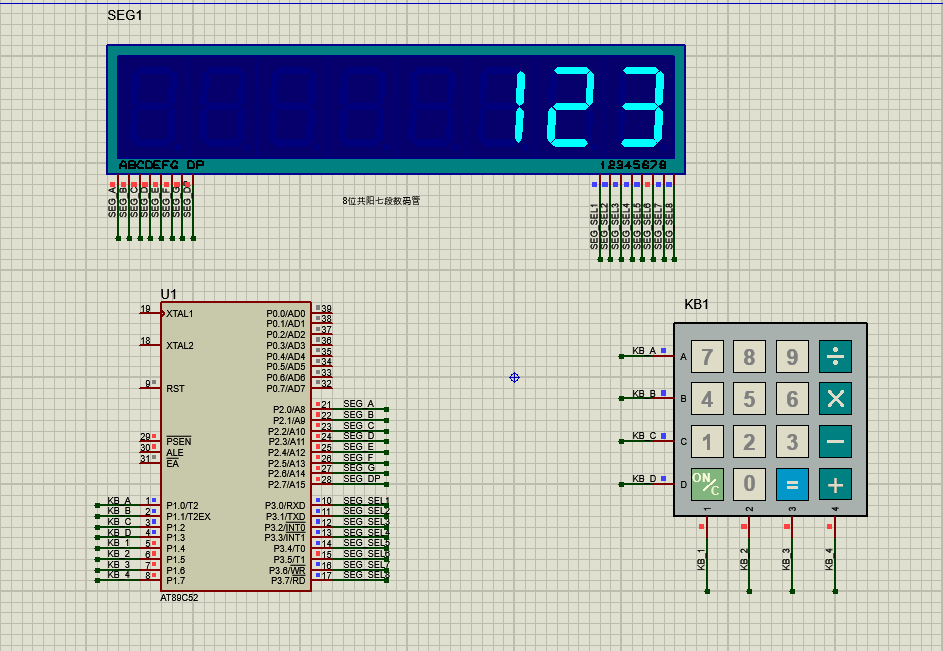
}

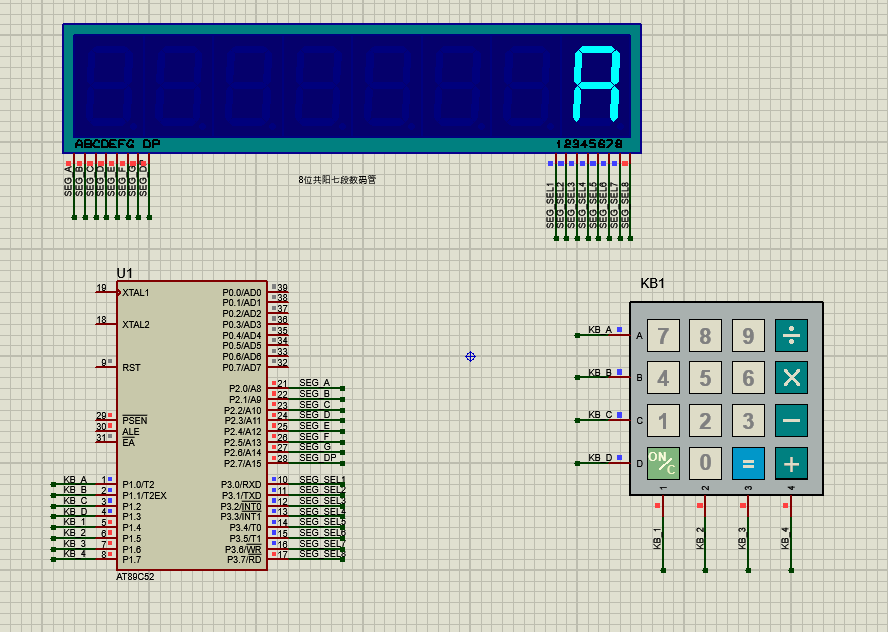
### **运行结果**

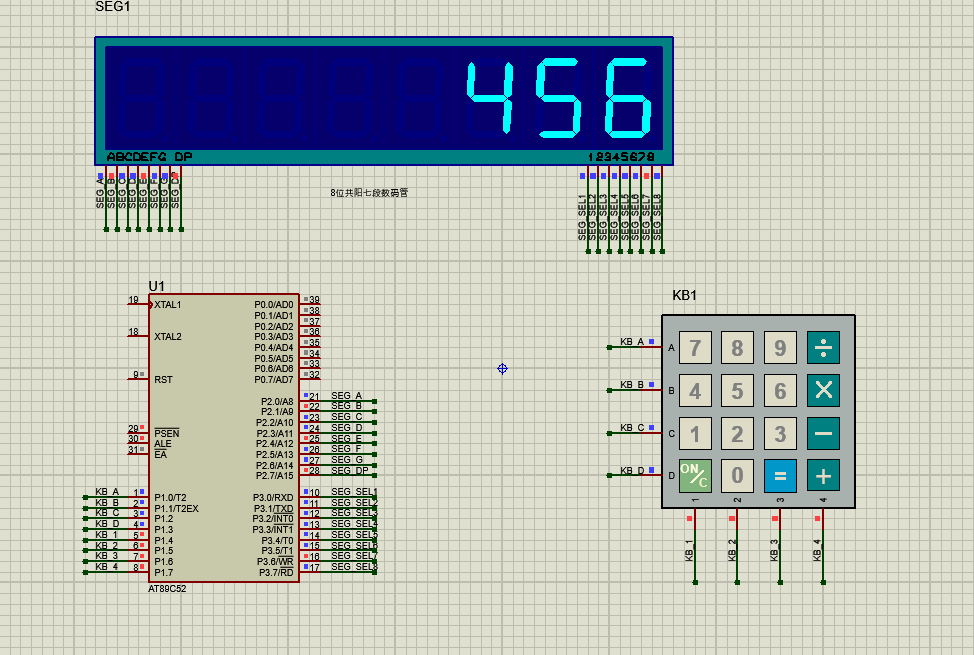


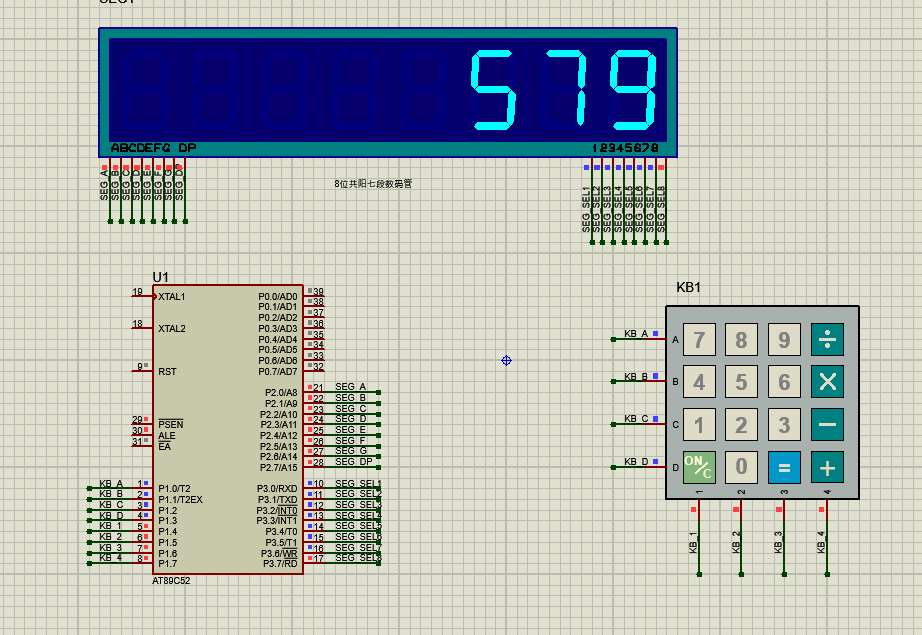
#### **1.加法**

123 + 456 = 579



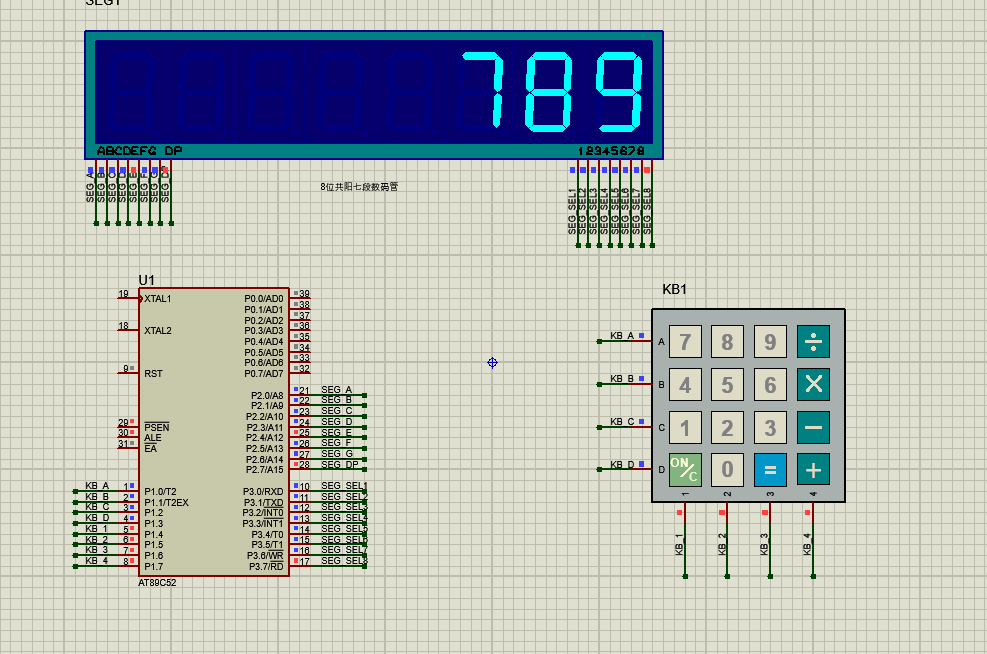


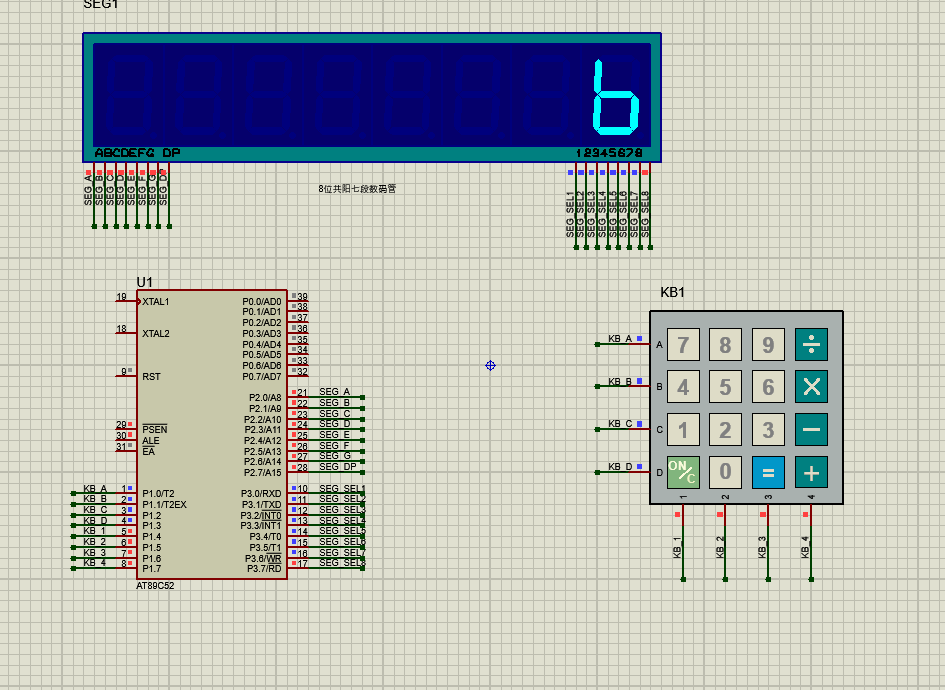


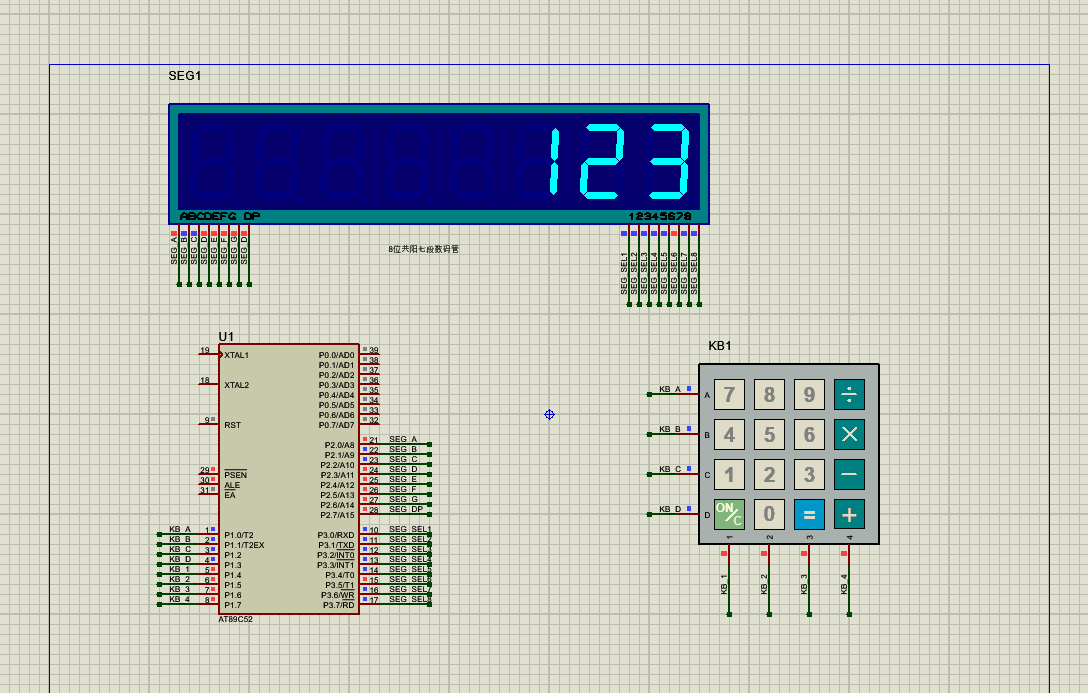


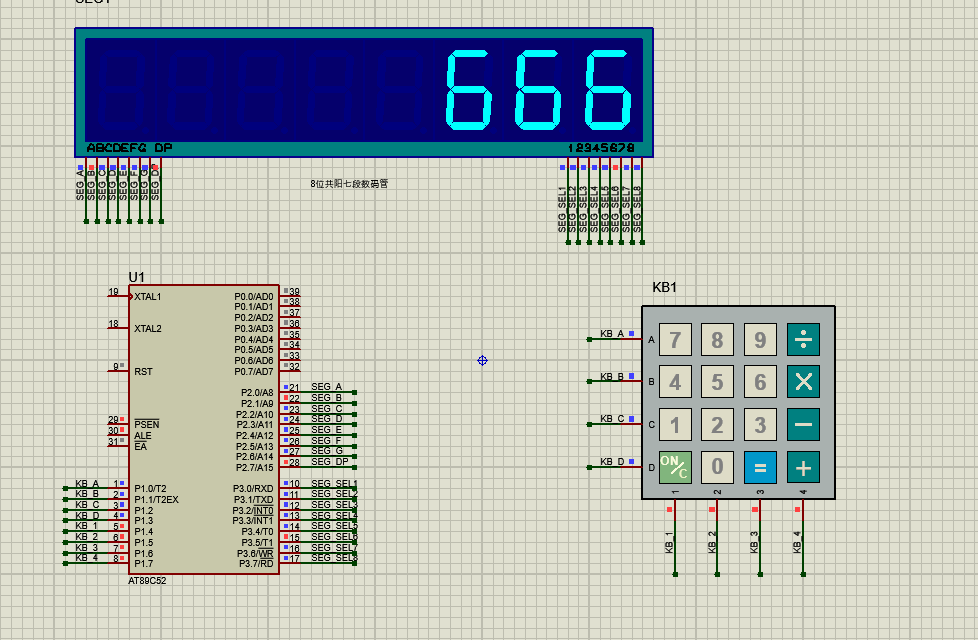
#### **2.减法**

789 - 123 = 666



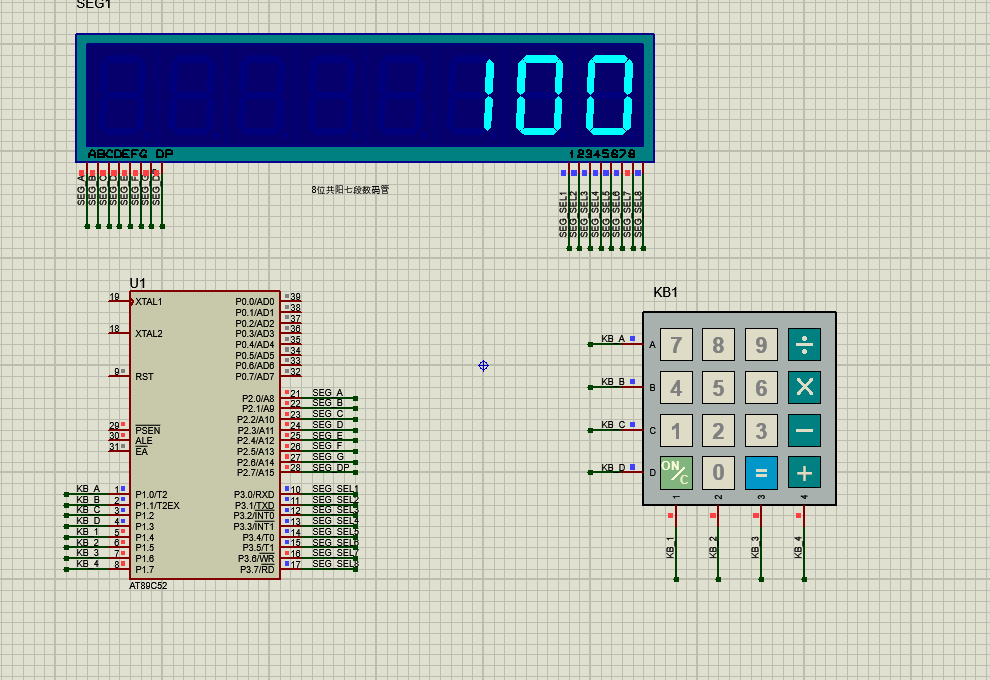


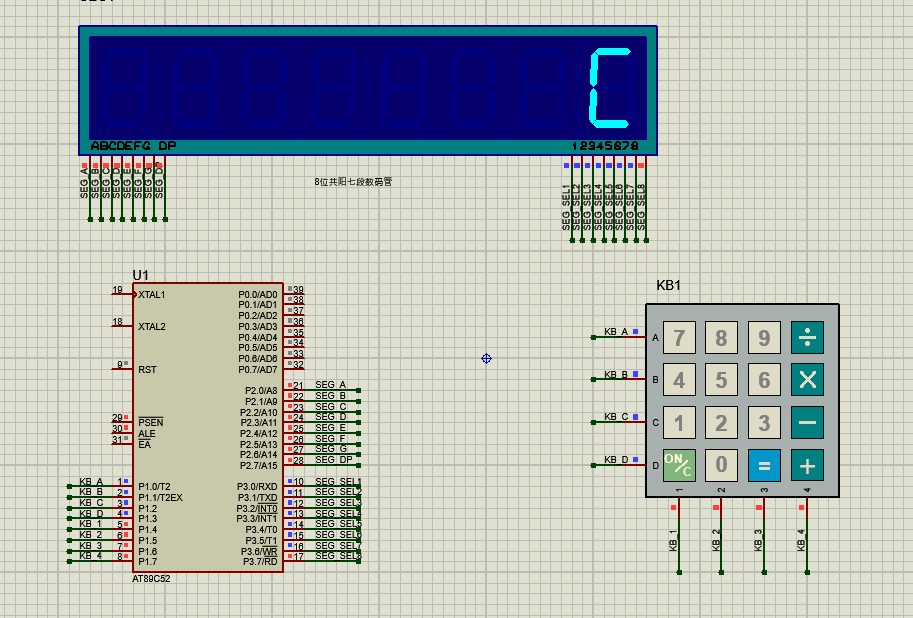


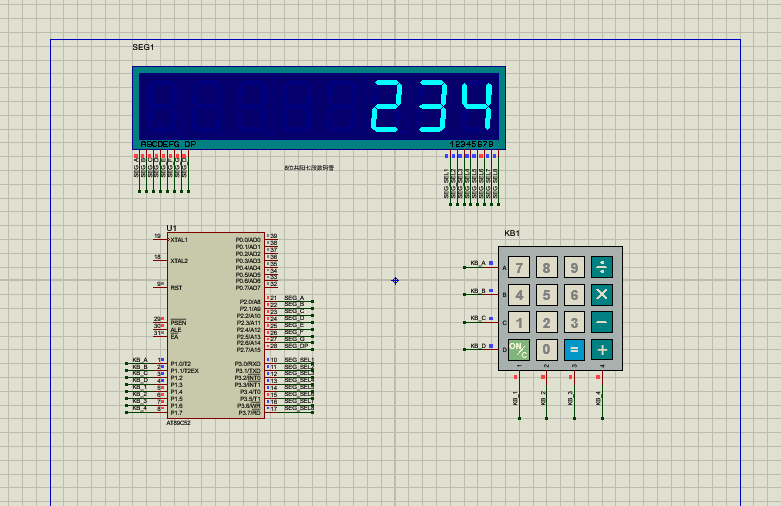


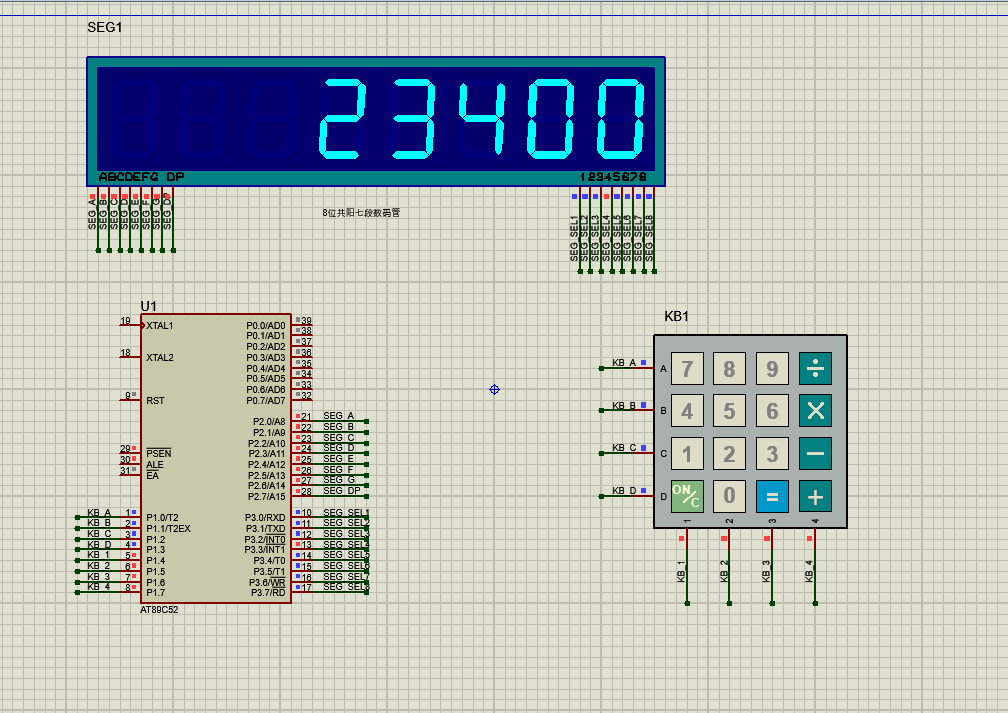
#### **3.乘法**

100 \* 234 = 23400



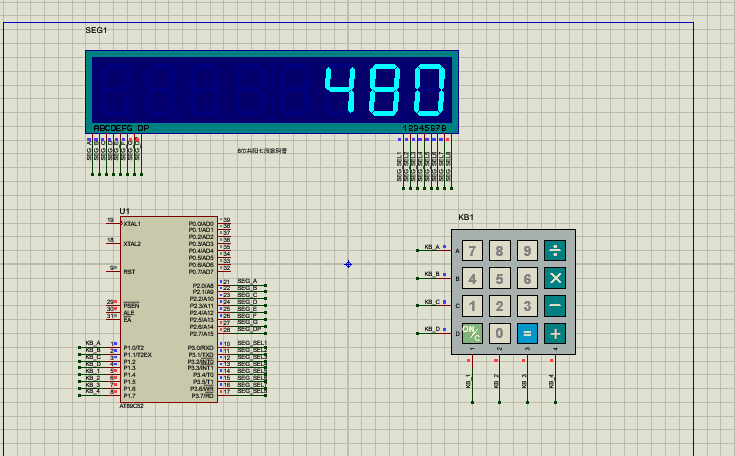


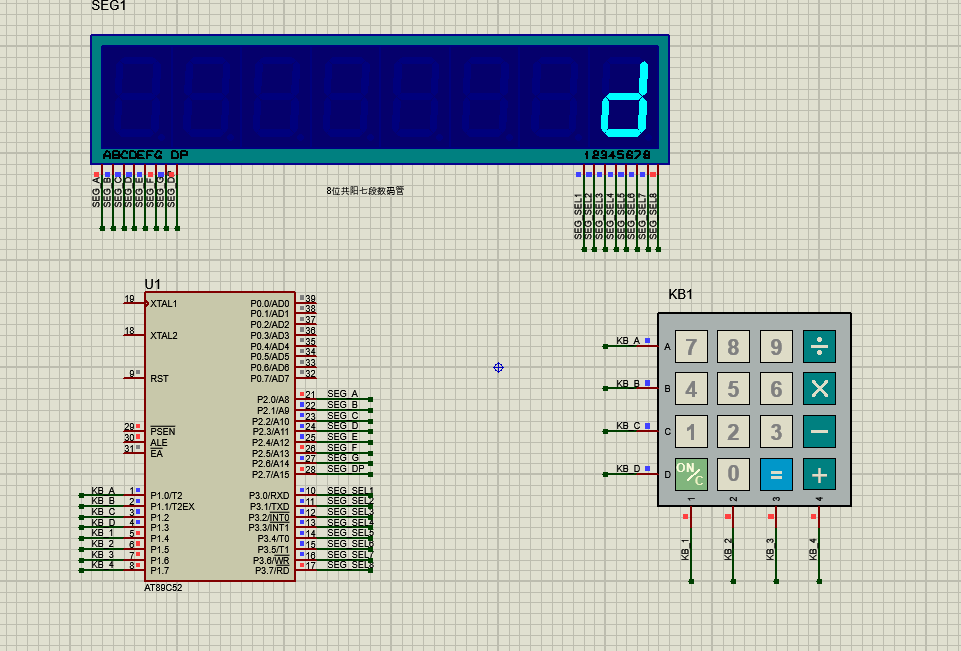


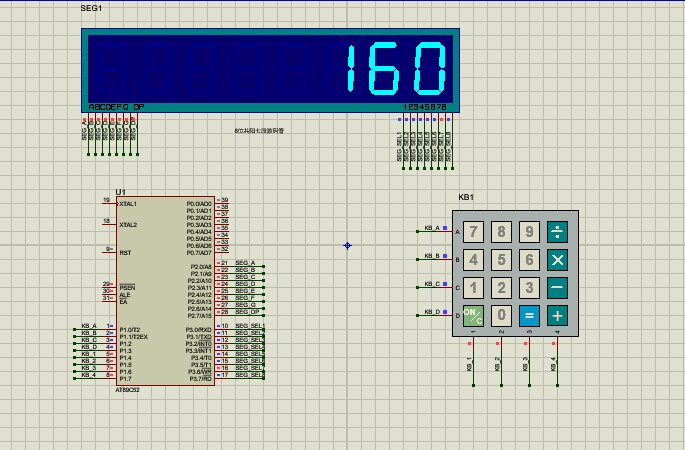


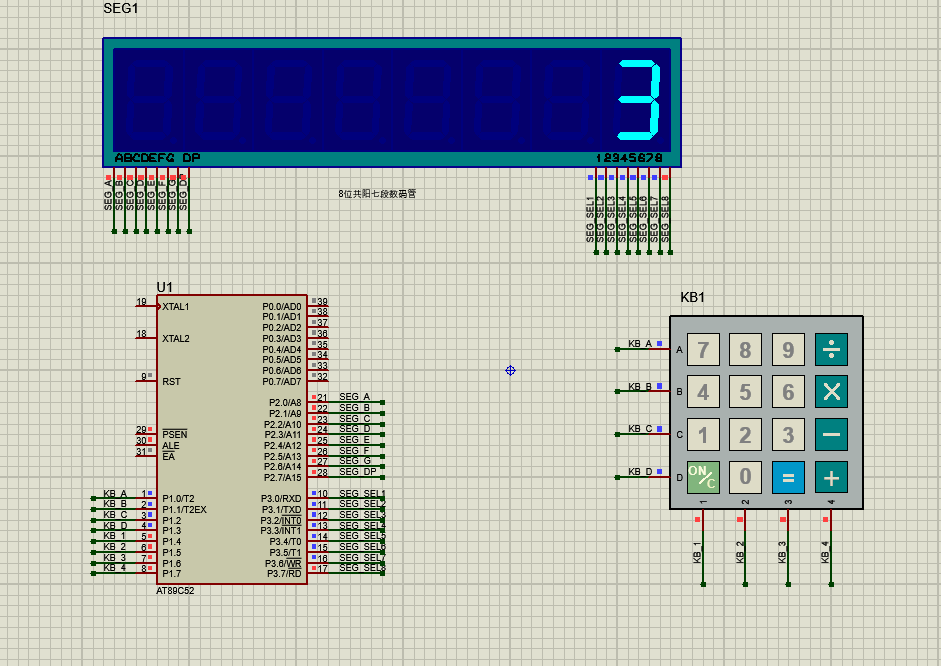
#### **4.除法**

480/160 =3









### **开发板运行结果**

开发板下程序代码与仿真代码有所不同，开发板为共阴极数码管，开发板个别引脚有问题，接线方式为P0接段码，P1接位码，P2接矩阵键盘，反着接

代码如下：

#include<reg52.h>

//共阴极数码管

unsigned char code segCodeTable[16] = {

0x3f,0x06,0x5b,0x4f,

0x66,0x6d,0x7d,0x07,

0x7f,0x6f,0x77,0x7c,

0x39,0x5e,0x79,0x71};

//位码表

unsigned char code posCodeTable[8] = {

0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe};//位选控制

unsigned char code key\_value[16]={0xD7,0xEB,0xDB,0xBB,

0xED,0xDD,0xBD,0xEE,

0xDE,0xBE,0x77,0x7B,

0x7D,0x7E,0xB7,0xE7};//键盘标识对应的键盘值

void dis\_seg7(unsigned long num,unsigned char dec\_hex);

unsigned char keyscan();

void delayms (void);

unsigned char pre\_P2;

void main()

{

unsigned char key0=0,key1=0,key2=0,dec\_hex=10,state=0;

unsigned char value = 0;

unsigned int j =0;

unsigned long dis\_num = 0,NUMA = 0,NUMB =0;

while(1)

{

value = keyscan();

if((pre\_P2 == 0xf0)&(P2 != 0xf0))

{

if(value>=0&&value<=9)

{

dec\_hex = 10;

key2 = key1;

key1 = key0;

key0 = value;

dis\_num = 100\*key2 + 10\* key1 + key0;

}

else if(value >=10&&value<=13)

{

NUMA = dis\_num;

dis\_num = value;

dec\_hex = 16;

key0 = 0;

key1 = 0;

key2 = 0;

if(value == 10)

{

state = '+';

}

else if(value == 11)

{

state = '-';

}

else if (value == 12)

{

state = '\*';

}

else if (value == 13)

{

state = '/';

}

}

else if(value == 14)

{

NUMB = dis\_num;

dec\_hex = 10;

if(state == '+')

{

dis\_num = NUMA + NUMB;

if(state == '+')

{

dis\_num = NUMA + NUMB;

}

else if(state == '-')

{

dis\_num = NUMA - NUMB;

}

else if(state == '\*')

{

dis\_num = NUMA \* NUMB;

}

else if(state == '/')

{

dis\_num = NUMA / NUMB;

}

}

}

//数码管清零

else if(value == 15)

{

dec\_hex = 10;

key0 = 0;

key1 = 0;

key2 = 0;

NUMA = 0;

NUMB = 0;

dis\_num = 0;

}

}

pre\_P2 = P2;

dis\_seg7(dis\_num,dec\_hex);

for(j = 0;j < 200;j++);

}

}

//七段数码管显示模块

void dis\_seg7(unsigned long num,unsigned char dec\_hex)

{

char i = 0,count = 8;

unsigned int j =0;

char dis\_data = 0;

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

{

dis\_data = num % dec\_hex;

num = num / dec\_hex;

P0 = segCodeTable[dis\_data];

P1 = posCodeTable[i];

for(j = 0;j < 200;j++);

P0 = 0x00; //清除余晖

if(num == 0)

{

break;

}

}

}

//延时模块

void delayms (void)

{

unsigned char i;

for (i=200;i>0;i--);

}

//按键扫描模块

unsigned char keyscan()

{

unsigned char rowscan,colscan,row\_col,i;

static unsigned char value = 0;

P2 = 0xf0; //P1输出 0xf0

if((P2&0xf0) != 0xf0) //P1读入数据，判断是否有按键按下

{

delayms();//延时消抖

if((P2&0xf0) != 0xf0)

{

//本代码采用逐行扫描的方式来确定按下的按键

rowscan = 0xfe;

while((rowscan&0x10)!=0)

{

P2 = rowscan;

colscan = P2 & 0xf0;

if(colscan != 0xf0)

{

//确定按下按键的行号与列号，查表，返回按键值

row\_col = colscan | (rowscan&0x0f);

for(i = 0;i<16;i++)

{

if(row\_col == key\_value[i])

{

value = i;

}

}

break;

}

else

{

rowscan = (rowscan << 1) | 0x1;

}

}

}

}

return value;

}

1.加法

123 + 456 =579

