# 实验3.7 LCD1602显示实验

姓名：孟麟芝 学号：201800121050 实验时间：2020.6.24

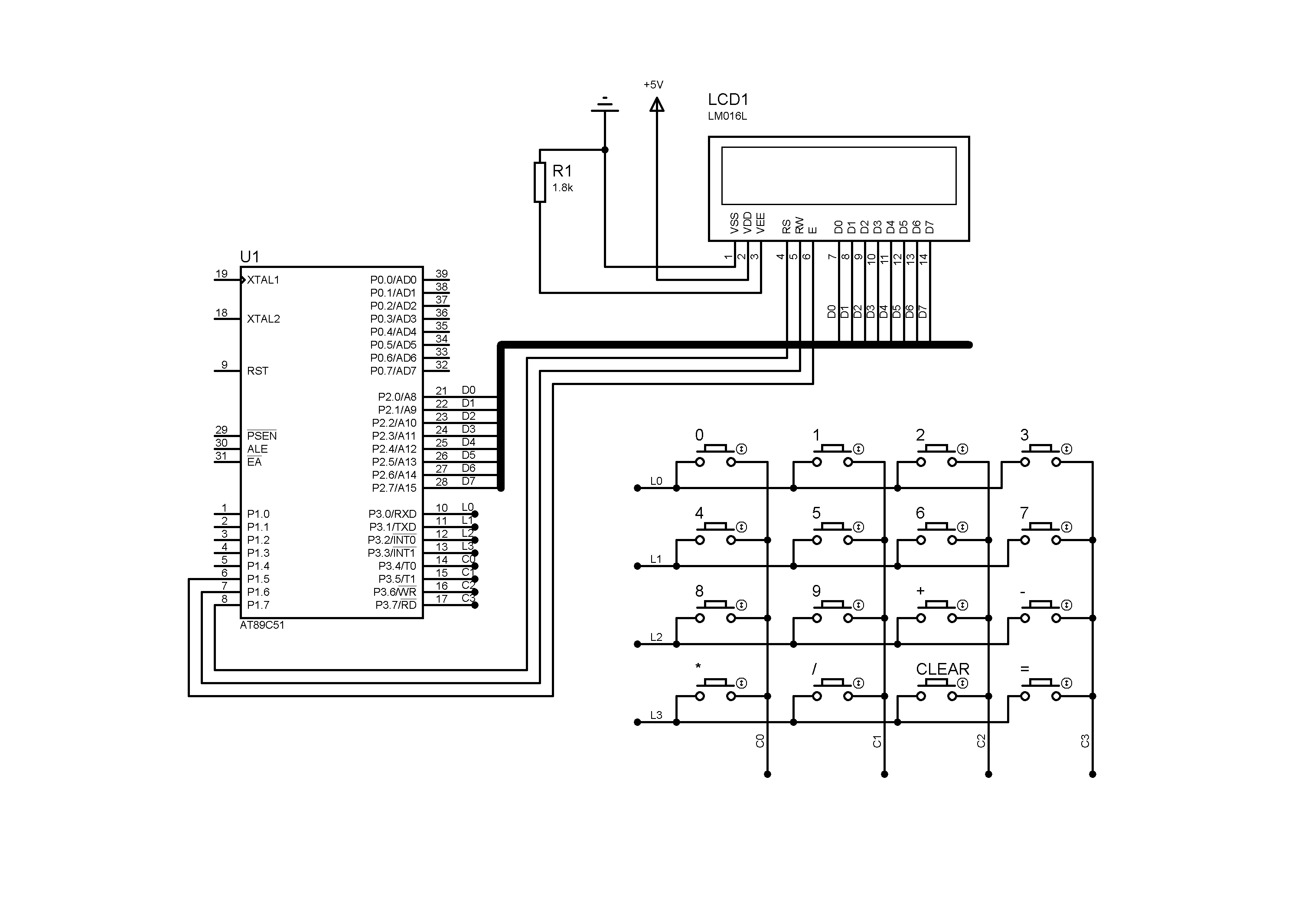
## 实验要求

（1）编写源程序并进行注释，叙述实验原理，画出流程图

（2）记录实验过程

（3）记录Proteus仿真结果

## 实验电路及功能



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **选件编号** | **元件名称** | **参数** | **所在元件库类名** | **子类名** | **生产厂家** |
| U1 | AT89C51 |  | Microprocessor ICs | 8051 Family | ATMEL |
| 微处理器 | 8051 家族 |
| LCD1 | LM016L |  | Optoelectronic | Alpha Numeric LCDs |  |
| 光电器件 | 字符数字 LCD |
| R1 | RES | 1.8k | Resistors | Generic |  |
| 电阻 | 一般的 |
| KEY0-KEY15 | BUTTON |  | Switches & Relays | Switches |  |
| 开关与继电器 | 开关 |

实验功能为利用LCD1602和16个按键实现简单的十进制数加减乘除四则混合运算。其中按键KEY0-KEY9分别代表了数字0-9，按键10-13分别代表了运算符“+,-,\*,/”，按键“15”代表“=”，按键14代表清除命令，以便进行下一次的输入和计算。不管什么时候按下清除按键计算过程停止，两个输入变量都将清0，屏幕将清屏。

LCD1602的第一行用于显示所输入的两个计算数以及计算符，第二行用于显示计算结果。结果允许为负数，但输入的两个输入数都必须是双字节正整数范围内的数，即：0-32767之间。除数必须保证不为0，否则将报错。除数时必须能同时显示商与余数。

## 实验原理

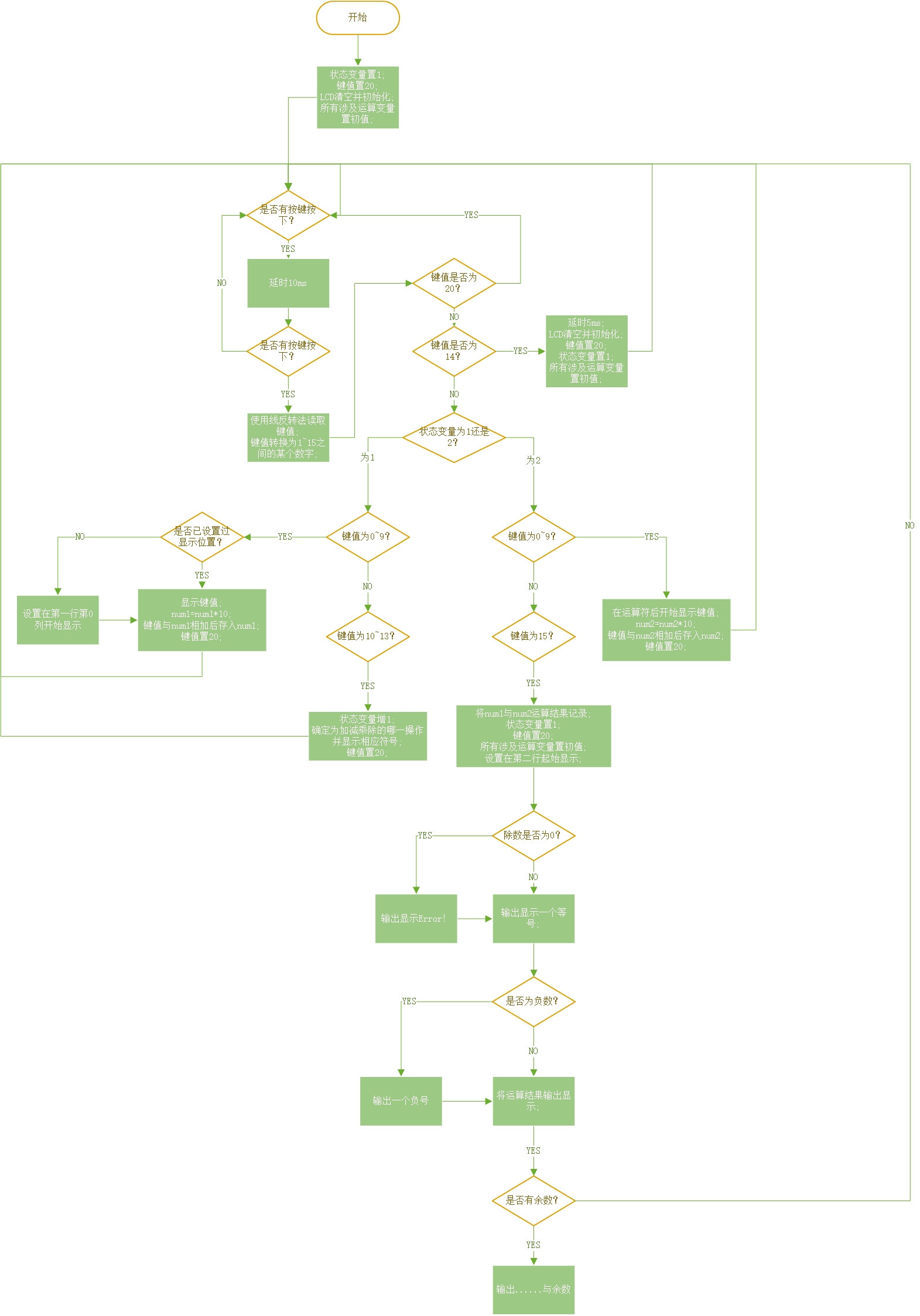
按键的扫描与识别可以参考“3.4.1 示例实验”中的方法。LCD1602的显示控制可以参考“3.7.2 LCD1602示例实验”。

编成时要有一个状态变量，该变量用于记录当前是输入的哪个变量。输入第一个变量，遇到输入运算符时结束第一个变量的输入。输入第二个变量，遇到“=”号时结束第二个变量的输入，并且开始计算结果。

计算结果由于是16进制的，要将其转换成十进制，并将该十进制的数转换成字符串后逐位显示出来。减法时要注意结果是否为负，除法时要注意除数是否为0，结果是否带有余数。

另外，按键要注意去抖动处理。

下面是程序的流程图



## 实验过程

（1）根据上述实验内容，参考1.2.2，在Proteus环境下建立图3.22所示原理图。

（2）根据实验原理画出流程图，并编写源程序。

（3）运行Keil5C51开发环境，按照1.1.3节介绍的方法建立工程，CPU为AT89C51。

（4）按照1.2.2第（6）节介绍的方法将C语言源程序加入工程，并设置工程属性，将其晶振频率设置为12MHz，选择输出可执行文件，仿真方式为选择硬仿真，并选择其中的“PROTEUS VSM MONITOR51 DRIVER”仿真器。

（5）构造（Build）工程。如果输入有误进行修改，直至构造正确，生成可执行程序为止。

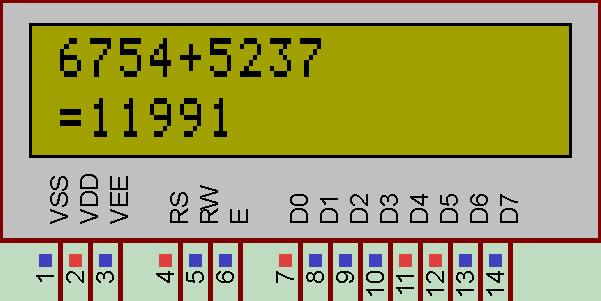
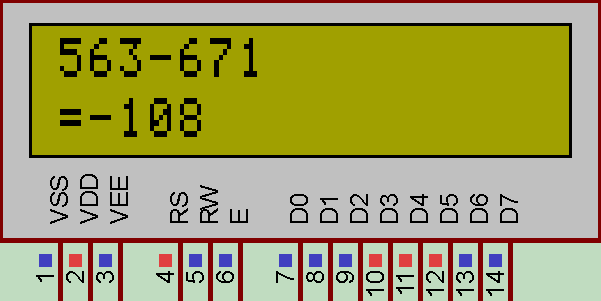
（6）为AT89C51设置该可执行程序。

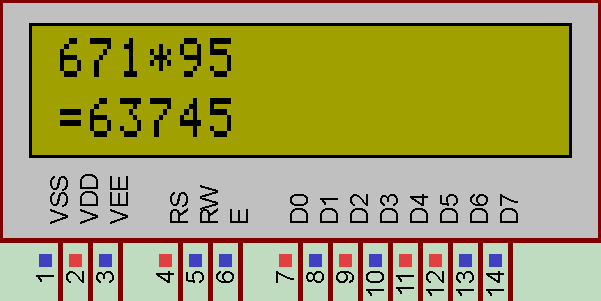
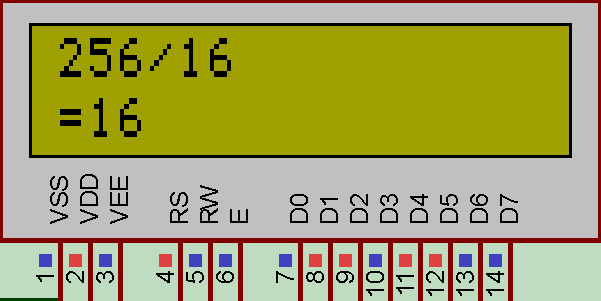
（7）运行程序，点击按键输入数据与运算符，计算，观察计算结果，并验证其是否正确。

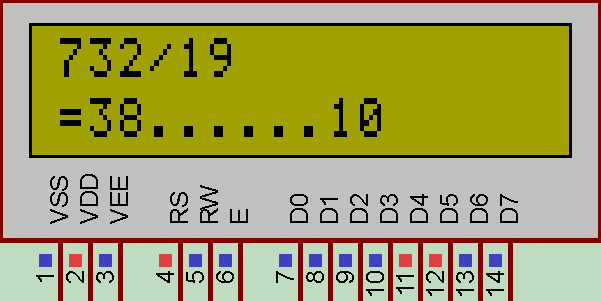
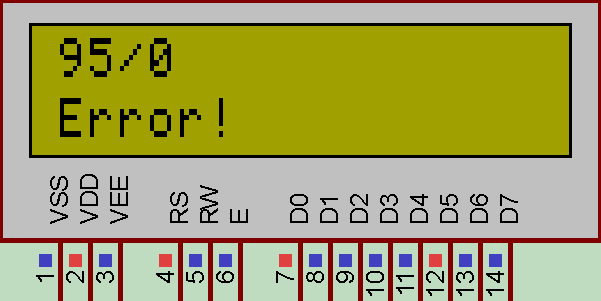
（8）输入过程中，按“清除按键”观察结果，重新输入数据计算并验证。

## 实验结果

进行连续的加减乘除操作，每做一次操作后点击CLEAR实现复位，得到如下结果

可见，该程序可以实现指定的功能

## 实验源程序

#include<reg51.h>

#include<intrins.h>

#define uchar unsigned char

#define uint unsigned int

uchar code KEY\_TABLE[] = { 0xee,0xde,0xbe,0x7e,0xed,0xdd,0xbd,0x7d,0xeb,0xdb,0xbb,0x7b,0xe7,0xd7,0xb7,0x77 };//按键键码表，分别为1~9,加减乘除，清除和等号

uchar data disresult[] = { 0xf,0xf,0xf,0xf,0xf,0xf };

uchar data disremain[] = { 0xf,0xf,0xf,0xf,0xf,0xf };

uchar code line1[] = "Error!";

uchar code line2[] = "......";

sbit LCD\_RS = P1^7;//定义控制引脚

sbit LCD\_RW = P1^6;

sbit LCD\_EN = P1^5;

sbit BF = P2^7;//用于忙检测子程序

uchar flag = 0;//防止忙检测死循环

uint temp, temp2, key, m, j, order, error, minus, num1, num2, result, opr, remain;

/\*\*\*\*\*\*\*\*\*\*\*\*延时子程序\*\*\*\*\*\*\*\*\*\*\*\*/

void delays(uint ms)//参数为多少即为多少毫秒

{

uint t;

while (ms--)for (t = 0; t < 120; t++);

}

/\*\*\*\*\*\*\*\*\*\*\*\*LCD 忙检查子程序\*\*\*\*\*\*\*\*\*\*\*\*/

bit lcd\_busy()

{

bit result;

LCD\_RS = 0;

LCD\_RW = 1;

LCD\_EN = 1;

delays(5);

result = BF; //返回数据最高位BF代表是否忙

LCD\_EN = 0;

flag++;

return result;

}

/\*\*\*\*\*\*\*\*\*\*\*\*写命令子程序\*\*\*\*\*\*\*\*\*\*\*\*/

void lcd\_wcmd(uchar cmd)

{

{while (lcd\_busy() & (flag <= 10)); }//防止忙检测进入死循环

LCD\_RS = 0;

LCD\_RW = 0;

LCD\_EN = 0;

delays(5);

P2 = cmd;

delays(5);

LCD\_EN = 1;

delays(5);

LCD\_EN = 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*写数据子程序\*\*\*\*\*\*\*\*\*\*\*\*/

void lcd\_wdat(uchar dat)

{

{while (lcd\_busy() & (flag <= 10)); }//防止忙检测进入死循环

LCD\_RS = 1;

LCD\_RW = 0;

LCD\_EN = 0;

delays(5);

P2 = dat;

delays(5);

LCD\_EN = 1;

delays(5);

LCD\_EN = 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*LCD初始化子程序\*\*\*\*\*\*\*\*\*\*\*\*/

void lcd\_init()

{

delays(5); //等待 LCD 电源稳定

lcd\_wcmd(0x38); //功能设定指令中 DL=1,N=1,F=0,8 位数据宽度,16\*2 显示，

//5\*7 点阵字符

delays(5);

lcd\_wcmd(0x0c); //显示开关控制指令中 D=1,C=0,B=1,显示开，关光标,不闪烁

delays(5);

lcd\_wcmd(0x06); //进入模式设置指令中 I/D=1,S=0,地址自动增加

delays(5);

lcd\_wcmd(0x01); //清除 LCD 显示内容,清屏指令 DB7-DB0 部分为 01H

delays(5);

}

/\*\*\*\*\*\*\*\*\*\*\*主程序\*\*\*\*\*\*\*\*\*\*\*/

void main()

{

key = 20;

order = 1;

num1 = num2 = m = temp = temp2 = minus = error = remain = 0;

lcd\_init();

while (1)

{

P3 = 0xF0;//行输出0

temp = P3;//读取列输入

temp = temp & 0xF0;//保留有效位

if (temp != 0xF0)

{

delays(10);//延时10毫秒

P3 = 0xF0;//重新行输出0

temp = P3;//读取列输入

temp = temp & 0xF0;//保留有效位

if (temp != 0xF0)

{

key = temp;//记录按键所在列

temp = temp | 0x0F;//变为列输出0

P3 = temp;

temp = P3 & 0x0F;//记录按键所在行

key = temp | key;//合并键码

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

{

if (key == KEY\_TABLE[j])

{

key = j;

break;

}

}

}

}

else

{

if (key == 20)delays(5);//保证无按键时不会进行任何操作

else {

if (key == 14)//复位操作

{

delays(5);

lcd\_init();

key = 20;

order = 1;

num1 = num2 = temp = m = temp2 = minus = error = remain = 0;

}

for (j = 0; j < 6; j++)

{

disresult[j] = disremain[j] = 0xf;

}

switch (order)

{

case 1:

{if ((key >= 0) & (key <= 9))

{

if (m == 0)

{

lcd\_wcmd(0x80); m++;

}//从第一行第一列开始输入，且保证输入第一个数过程中不再光标归位

delays(5);

lcd\_wdat(key + 48);

num1 = num1 \* 10 + key;

key = 20;

}

if ((key >= 10) & (key <= 13))

{

order++;

opr = key;

switch (opr)

{

case 10:lcd\_wdat(opr + 0x21); break;

case 11:lcd\_wdat(opr + 0x22); break;

case 12:lcd\_wdat(opr + 0x1e); break;

case 13:lcd\_wdat(opr + 0x22); break;//分别将加减乘除号转为ASCII码

}

}

}; break;

case 2:

{if ((key >= 0) & (key <= 9))

{

lcd\_wdat(key + 48);

delays(1000);

num2 = num2 \* 10 + key;

key = 20;

}

if (key == 15)

{

order = 1;

key = 20;

switch (opr)

{

case 10:result = num1 + num2; break;

case 11:if (num1 >= num2)result = num1 - num2; else { minus = 1; result = num2 - num1; }break;

case 12:result = num1 \* num2; break;

case 13:if (num2 == 0)error = 1; else { result = num1 / num2; remain = num1 % num2; }break;

default:break;

}

if (error == 1)

{

lcd\_wcmd(0xc0);

j = 0;

while (line1[j] != '\0')

{

lcd\_wdat(line1[j]);

delays(5);

j++;

}

}//除0时直接报错，不进行其他任何操作

else

{

lcd\_wcmd(0xc0);

delays(5);

lcd\_wdat(0x3d);//上三行显示输出一个等号

if (minus == 1)lcd\_wdat(0x2d);//若结果为负先输出一个负号

for (j = 0; j < 6; j++)

{

disresult[5 - j] = result % 10;

result = result / 10;

if (result == 0)break;

}

for (j = 0; j < 6; j++)

{

if (disresult[j] != 0xf)lcd\_wdat(disresult[j] + 48); delays(5);

}//以上两个循环将结果转十进制并显示

if (remain != 0)

{

j = 0;

while (line2[j] != '\0')

{

lcd\_wdat(line2[j]);

delays(5);

j++;

}

for (j = 0; j < 6; j++)

{

disremain[5 - j] = remain % 10;

remain = remain / 10;

if (remain == 0)break;

}

for (j = 0; j < 6; j++)

{

if (disremain[j] != 0xf)lcd\_wdat(disremain[j] + 48); delays(5);

}//若除法有余数，以上两个循环将余数转十进制并显示

}

}

}}; break;

}

}

}

}

}