**LCD and Keypad**

**Task 06**

****

Spring 2022

CSE-307 Microprocessor Based system Design

Submitted by: **Ashfaq Ahmad**

Registration No: **19PWCSE1795**

Class Section: **B**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Student Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Submitted to:

**Dr: Bilal Habib**

**June** 6, 2022

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

**Task:**

In this assignment, you are required to design a calculator that should be able to do the following operations, Addition, Subtraction, Multiplication, and Division.

**Source Code:**

//it will take only one digit before and after operator

#include <reg51.h>

#include <stdio.h>

//Functions 7

void Ports\_init();

void lcd\_init();

void delay(int);

void writedata(char);

void writecmd(int);

char read\_keypad();

char get\_key();

//pins names

sbit RA=P1^0;   //rows

sbit RB=P1^1;

sbit RC=P1^2;

sbit RX=P1^3;     //i can't write RD b/c i think it's a keyword.

sbit C1=P1^4;  //coloumns

sbit C2=P1^5;

sbit C3=P1^6;

sbit C4=P1^7;

sbit RS=P3^0;  //control pins..

sbit E=P3^1;

//veriables

unsigned int index=0,int\_result;

char arr[3],char\_result;

//main function

void main()

{

    char key;

    lcd\_init();

    Ports\_init();

    writecmd(0x94);   //Sets cursor to line 3 of display

    writedata('A');

    writedata('S');

    writedata('H');

    writedata('F');

    writedata('A');

    writedata('Q');

    writedata(' ');

    writedata('A');

    writedata('H');

    writedata('M');

    writedata('A');

    writedata('D');

    writecmd(0xD4);    //Sets cursor to line 4 of display

    writedata('M');

    writedata('B');

    writedata('S');

    writedata('D');

    writedata(' ');

    writedata('T');

    writedata('A');

    writedata('S');

    writedata('K');

    writedata('0');

    writedata('6');

    writecmd(0x80);   //force cursor to the begining of first line.

    while(1)

    {

        key=get\_key();

        if(key=='C')

        {

            writecmd(0x01);

            arr[0]=arr[1]=arr[2]='n';

            int\_result=0;

            index=0;

        }

        else if(key=='=')

        {

            writedata('=');

            if(arr[1]=='+')

            {

                int\_result=(arr[0]-'0')+(arr[2]-'0');   //-'0' mean converting char to int.

            }

            else if(arr[1]=='-')

            {

                int\_result=(arr[0]-'0')-(arr[2]-'0');

            }

            else if(arr[1]=='x')

            {

                int\_result=(arr[0]-'0')\*(arr[2]-'0');

            }

            else if(arr[1]=='/')

            {

                int\_result=(arr[0]-'0')/(arr[2]-'0');

            }

            if(int\_result>9)

            {

                int arr1[2];

                arr1[0]=int\_result%10;

                arr1[1]=int\_result/10;

                char\_result=arr1[1]+'0';  //+'0' mean converting int to char.

                writedata(char\_result);

                char\_result=arr1[0]+'0';

                writedata(char\_result);

            }

            else

            {

                char\_result=int\_result+'0';

                writedata(char\_result);

            }

        }

        else

            writedata(key);

    }

}

void delay(unsigned int x)

{

    int i;

    for(i=0; i<x; i++);

}

void writecmd(int c)

{

    RS=0;

    P2=c;

    E=1;

    delay(150);

    E=0;

    delay(150);

}

void writedata(char d)

{

    RS=1;

    P2=d;

    E=1;

    delay(150);

    E=0;

    delay(150);

}

void lcd\_init()

{

  delay(15000);

    writecmd(0x30);

    delay(4500);

    writecmd(0x30);

    delay(300);

    writecmd(0x30);

    delay(650);

    writecmd(0x38);

    writecmd(0x0C);

    writecmd(0x01);

    writecmd(0x06);

}

void Ports\_init()   //this function don't need. if we skip it still it will work.

{

    P2=0x00;    //Data lines

    P1=0xF0;   //make columns as input and rows as output.

    E=0;      //initially 0. if we wanna latch command or data we will make it 1 and then 0. data will latch.

    RS=0;     //command register when 0 and data register when 1

}

char read\_keypad()

{

    RA=0; RB=1; RC=1; RX=1;

    if(C1==0){delay(10000); while(C1==0); return '7';}

    if(C2==0){delay(10000); while(C2==0); return '8';}

    if(C3==0){delay(10000); while(C3==0); return '9';}

    if(C4==0){delay(10000); while(C4==0); return '/';}

    RA=1; RB=0; RC=1; RX=1;

    if(C1==0){delay(10000); while(C1==0); return '4';}

    if(C2==0){delay(10000); while(C2==0); return '5';}

    if(C3==0){delay(10000); while(C3==0); return '6';}

    if(C4==0){delay(10000); while(C4==0); return 'x';}

    RA=1; RB=1; RC=0; RX=1;

    if(C1==0){delay(10000); while(C1==0); return '1';}

    if(C2==0){delay(10000); while(C2==0); return '2';}

    if(C3==0){delay(10000); while(C3==0); return '3';}

    if(C4==0){delay(10000); while(C4==0); return '-';}

    RA=1; RB=1; RC=1; RX=0;

    if(C1==0){delay(10000); while(C1==0); return 'C';}

    if(C2==0){delay(10000); while(C2==0); return '0';}

    if(C3==0){delay(10000); while(C3==0); return '=';}

    if(C4==0){delay(10000); while(C4==0); return '+';}

    return 'n';

}

char get\_key()

{

    char key='n';

    while(key=='n')

    {

        key=read\_keypad();

    }

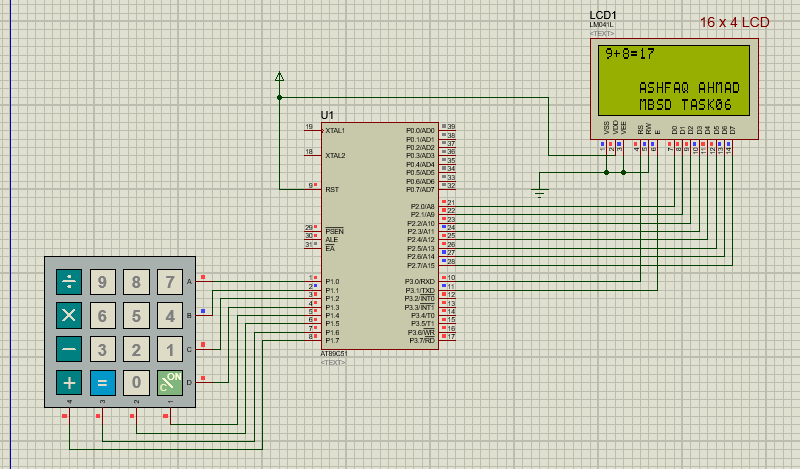
  arr[index++]=key;   //post increment.

    return key;

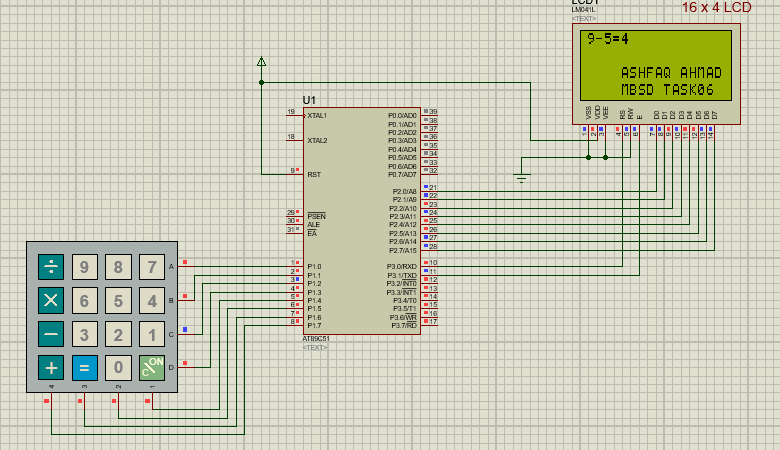
}

**Circuit Diagram:**

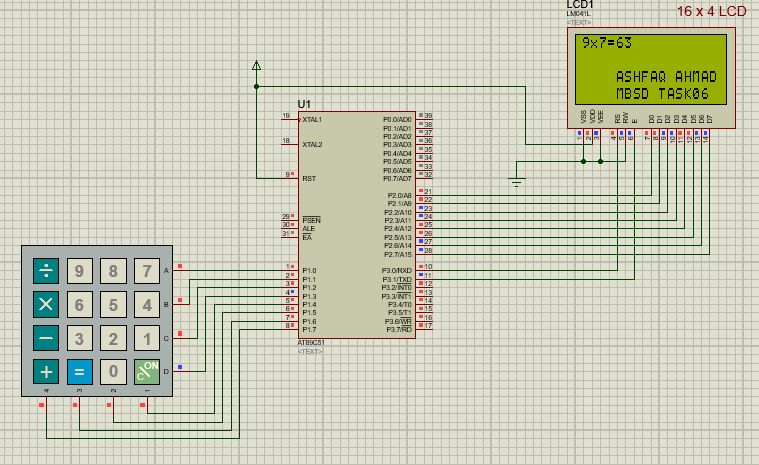
**(Addition)**



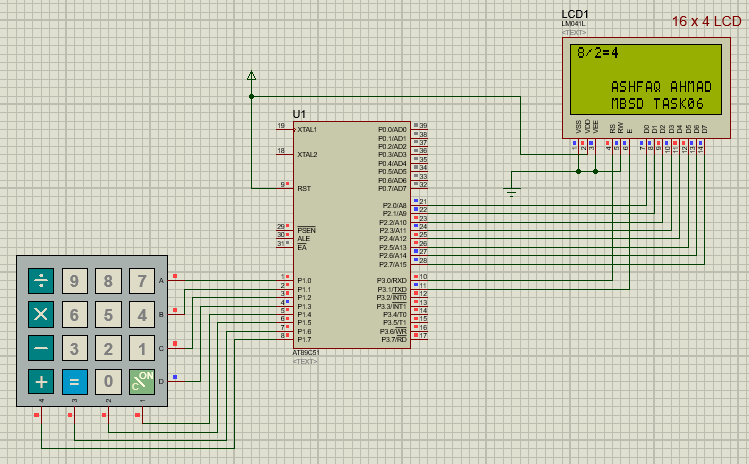
Subtraction:

****

**Multiplication:**

****

**Division:**

****