**University of Engineering and Technology, Peshawar**

Department of Computer Systems Engineering.

*Course 307: CSE-307 MicroProcessor Based System Design*



**Zubair Khan 19** PWCSE **1797**

Section A

Batch 21 (Spring\_2022)

**Submitted to** Dr. Bilal Habib

**TASK \_#\_: 06**

**Source Code**

#include <reg51.h>

#include <stdio.h>

#define ERROR 0 *// Any value other than 0 to 9 is good here*

*//Function declarations*

**void** ports\_initialization(**void**);

**void** delay(**int**);

**void** Lcd\_initialization(**void**);

**void** write\_command(**int**);

**void** write\_data(**char**);

**void** Return(**void**);

**char** READ\_SWITCHES(**void**);

**char** get\_key(**void**);

**int** get\_number(**char**);

**char** get\_operation(**char**);

**void** display\_result(**int**);

sbit RowA = P1^0;

sbit RowB = P1^1;

sbit RowC = P1^2;

sbit RowD = P1^3;

sbit C1 = P1^4;

sbit C2 = P1^5;

sbit C3 = P1^6;

sbit C4 = P1^7;

sbit E = P3^6;

sbit RS = P3^7;

*// Main program*

**void** main(**void**)

{

**char** key;

**int** number1,number2;

**char** operation;

 ports\_initialization();

 Lcd\_initialization();

 write\_command(0x80);

 while(1)

 {

 key = get\_key();

 number1=get\_number(key);

 if (number1 != ERROR)

 {

write\_data(key);

key=get\_key();

operation=get\_operation(key);

write\_data(operation);

if(operation!=ERROR)

{

 key=get\_key();

 number2=get\_number(key);

 if(number2!=ERROR)

 {

 write\_data(key);

 key=get\_key();

 if(key=='=')

 {

 write\_data(key);

 switch(operation){

 case '+': display\_result(number1+number2) ; break;

 case '-': display\_result(number1-number2); break;

 case 'x': display\_result(number1\*number2); break;

 case '/': display\_result(number1/number2); break;

 default: write\_command(0x01);

}

 }

 }

}

 }

 else{

 write\_command(0x01);

 }

 }

}

**void** ports\_initialization(**void**){

P0 = 0x00;

P1 = 0xf0;

P2 = 0x00;

P3 = 0x00;

}

**void** delay(**int** d){

**int** i;

 for(i=0;i<d;i++);

}

**void** write\_data(**char** s){

 RS = 1;

 P2 = s;

 E = 1;

 delay(150);

 E = 0;

 delay(150);

}

**void** write\_command(**int** a){

 RS = 0;

 P2 = a;

 E = 1;

 delay(150);

 E = 0;

 delay(150);

}

**void** Lcd\_initialization(**void**){

 delay(15000);

 write\_command(0x30);

 delay(4500);

 write\_command(0x30);

 delay(300);

 write\_command(0x30);

 delay(650);

 write\_command(0x38);

 write\_command(0x0c);

 write\_command(0x01);

 write\_command(0x06);

}

**void** Return(**void**){

 write\_command(0x02);

 delay(1500);

}

**char** Read\_Switches(**void**){

RowA = 0; RowB = 1; RowC = 1; RowD = 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 '/'; }

RowA = 1; RowB = 0; RowC = 1; RowD = 1; *//Test Row B*

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'; }

RowA = 1; RowB = 1; RowC = 0; RowD = 1; *//Test Row C*

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 '-'; }

RowA = 1; RowB = 1; RowC = 1; RowD = 0; *//Test Row D*

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

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(**void**)

{

**char** key = 'n';

while(key=='n')

key = READ\_SWITCHES();

return key;

}

**int** get\_number(**char** character)

{

 switch(character){

case '0': return 0; break;

case '1': return 1; break;

case '2': return 2; break;

case '3': return 3; break;

case '4': return 4; break;

case '5': return 5; break;

case '6': return 6; break;

case '7': return 7; break;

case '8': return 8; break;

case '9': return 9; break;

default:return ERROR; break;

 }

}

**char** get\_operation(**char** key){

 if(key=='+' || key=='-' || key=='x' || key=='/')

 {

 return key;

 }

 else{

return ERROR;

 }

}

**void** display\_result(**int** number)

{

**unsigned** **char** Unit = 0;

**unsigned** **char** Tenth= 0;

 if(number<0)

 {

number = -1\*number;

write\_data('-');

 }

 Tenth = (number/10);

 if( Tenth != 0)

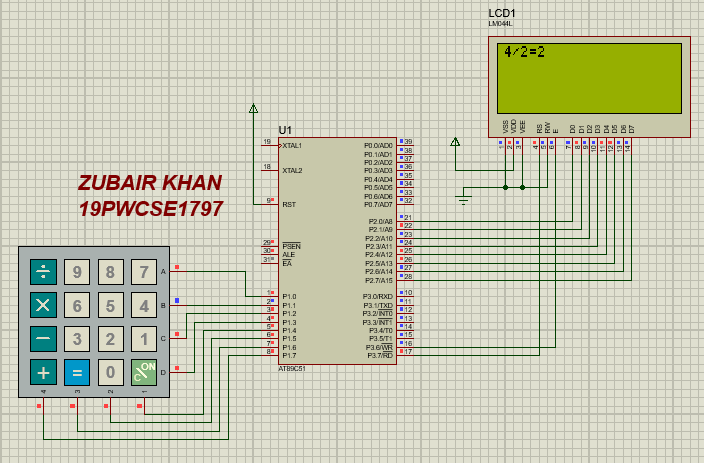
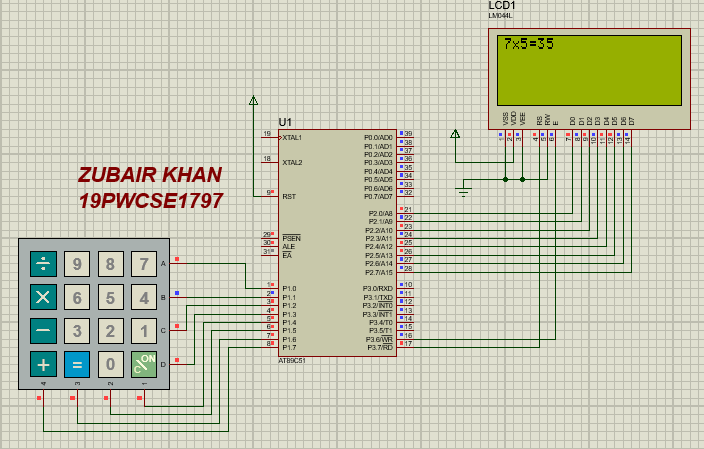
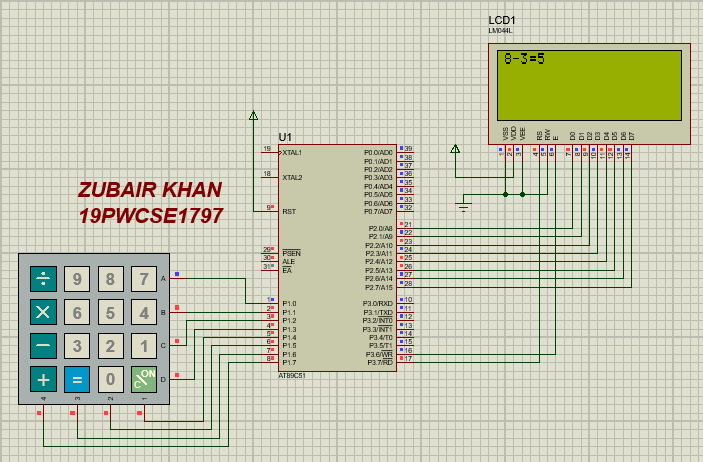
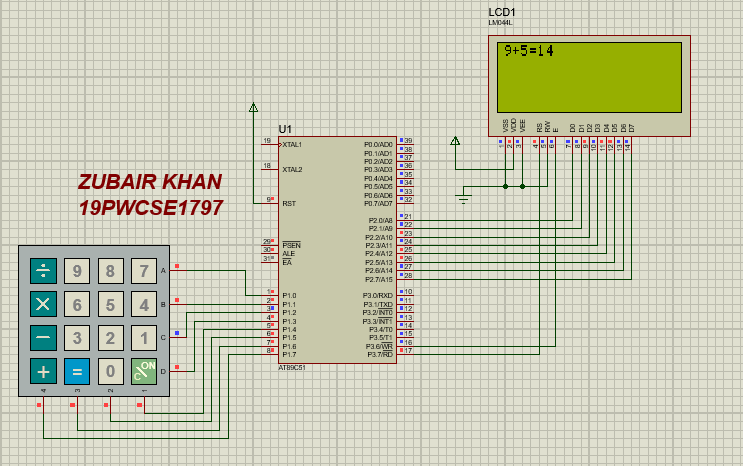
write\_data(Tenth+0x30); *// Make Char of TenthDigit and then display it on LCD*

 Unit = number - Tenth\*10;

 write\_data(Unit+0x30); *// Make Char of UnitDigit and then display it on LCD*

}

**Outputs**

****