

# University of Engineering and Technology, Peshawar

Department of Computer Systems Engineering.

Course 307: CSE-307 MicroProcessor Based System Design

**Zubair Khan**

Section

Batch

**Submitted to**



**19 PWCSE 1797**

A

21 (Spring\_2022)

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(0xc);
    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(1000);while (C1==0); return '7'; }
    if (C2 == 0) { delay(1000); while (C2==0); return '8'; }
    if (C3 == 0) { delay(1000); while (C3==0); return '9'; }
    if (C4 == 0) { delay(1000); while (C4==0); return '/'; }
    RowA = 1; RowB = 0; RowC = 1; RowD = 1; //Test Row B
    if (C1 == 0) { delay(1000); while (C1==0); return '4'; }
    if (C2 == 0) { delay(1000); while (C2==0); return '5'; }
    if (C3 == 0) { delay(1000); while (C3==0); return '6'; }
    if (C4 == 0) { delay(1000); while (C4==0); return 'x'; }
    RowA = 1; RowB = 1; RowC = 0; RowD = 1; //Test Row C
    if (C1 == 0) { delay(1000); while (C1==0); return '1'; }
    if (C2 == 0) { delay(1000); while (C2==0); return '2'; }
    if (C3 == 0) { delay(1000); while (C3==0); return '3'; }
    if (C4 == 0) { delay(1000); while (C4==0); return '-'; }
    RowA = 1; RowB = 1; RowC = 1; RowD = 0; //Test Row D
    if (C1 == 0) { delay(1000); while (C1==0); return '*'; }
    if (C2 == 0) { delay(1000); while (C2==0); return '0'; }
    if (C3 == 0) { delay(1000); while (C3==0); return '='; }
    if (C4 == 0) { delay(1000); 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



