Student Name : W.K.G.K Jayawardana

Student Number : EC/2021/006

Partner Name : S.M.A.D.R Pabasarani

Partner Number : EC/2021/007

Date Performed : 13/06/2025

Date of Submission : 13/06/2025

Experiment No: 11

BECS 31421

**MICROCONTROLLER-BASED SIMPLE CALCULATOR**

**DISCUSSION**

In this experiment, a simple microcontroller-based calculator was implemented using the PIC16F628A microcontroller, a 16×2 LCD, and a 4×4 keypad. The primary goal was to perform basic arithmetic operations using two-digit numerical inputs entered via the keypad.

The system was programmed in MikroC PRO using essential libraries such as LCD, Keypad 4×4, C\_String, and Conversion to facilitate communication between the microcontroller and the external devices. The keypad was scanned using polling, and ASCII codes were used to display characters on the LCD, ensuring compatibility and clarity.

User inputs were processed in a structured sequence: the user entered two digits to form the first number, selected an operator, then entered another two digits to form the second number. Once the '=' key was pressed, the microcontroller executed the operation and displayed the result.

However, in the final implementation, **only addition was successfully functional**, even though the program structure includes cases for subtraction, multiplication, and division. This limitation was due to the keypad mapping in the switch-case structure, where all three keys (8, 9, and 11) were incorrectly assigned the value **9**, preventing unique detection of '-', '\*', and '/' operations. As a result, the system could not correctly identify any operation other than '+'.

In summary, the experiment demonstrated the core principles of microcontroller-based arithmetic processing and peripheral interfacing. While limited to addition due to input mapping issues, the project effectively showed how LCDs, keypads, and microcontrollers can be combined to perform interactive computing tasks.

**SOURCE CODE**

unsigned int kp = 0;

int val = 0;

int PW1 = 0, PW2 = 0, PW3 = 0, PW4 = 0, PW5 = 0;

int number1 = 0, number2 = 0;

long Answer = 0;

char answerx[15];

int state = 0;

#define DIVISION '/'

#define MULTIPLY '\*'

#define SUBTRACT '-'

#define ADD '+'

char keypadPort at PORTB;

sbit LCD\_RS at RA0\_bit;

sbit LCD\_EN at RA1\_bit;

sbit LCD\_D4 at RA2\_bit;

sbit LCD\_D5 at RA3\_bit;

sbit LCD\_D6 at RA7\_bit;

sbit LCD\_D7 at RA6\_bit;

sbit LCD\_RS\_Direction at TRISA0\_bit;

sbit LCD\_EN\_Direction at TRISA1\_bit;

sbit LCD\_D4\_Direction at TRISA2\_bit;

sbit LCD\_D5\_Direction at TRISA3\_bit;

sbit LCD\_D6\_Direction at TRISA7\_bit;

sbit LCD\_D7\_Direction at TRISA6\_bit;

void ResetCalculator() {

PW1 = PW2 = PW3 = PW4 = PW5 = number1 = number2 = Answer = 0;

state = 0;

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Out(1, 1, "Enter Values:");

Lcd\_Cmd(\_LCD\_SECOND\_ROW);

Lcd\_Cmd(\_LCD\_BLINK\_CURSOR\_ON);

}

void main() {

CMCON = 0x07;

TRISA = 0x00;

PORTA = 0x00;

Keypad\_Init();

Lcd\_Init();

ResetCalculator();

while (1) {

do {

kp = Keypad\_Key\_Click();

Delay\_ms(50); // Debounce Delay

} while (!kp);

// Keypad Mapping

switch (kp) {

case 1: val = 7; break;

case 2: val = 4; break;

case 3: val = 1; break;

case 4: val = 0; break;

case 5: val = 8; break;

case 6: val = 5; break;

case 7: val = 2; break;

case 8: val = 9; break;

case 9: val = 9; break;

case 10: val = 6; break;

case 11: val = 3; break;

case 12: val = '='; break; // '=' key

case 13: val = '/'; break;

case 14: val = '\*'; break;

case 15: val = '-'; break;

case 16: val = '+'; break;

default: val = -1; break;

}

// Input Digits

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

if (state == 0) {

PW1 = val;

Lcd\_Chr\_CP(PW1 + '0');

state = 1;

} else if (state == 1) {

PW2 = val;

Lcd\_Chr\_CP(PW2 + '0');

number1 = PW1 \* 10 + PW2;

state = 2;

} else if (state == 3) {

PW4 = val;

Lcd\_Chr\_CP(PW4 + '0');

state = 4;

} else if (state == 4) {

PW5 = val;

Lcd\_Chr\_CP(PW5 + '0');

number2 = PW4 \* 10 + PW5;

state = 5;

}

}

// Input Operator

if ((val == '+' || val == '-' || val == '\*' || val == '/') && state == 2) {

PW3 = val;

Lcd\_Chr\_CP(PW3); // show operator

state = 3;

}

// Equals (=) to calculate

if (val == '=' && state == 5) {

switch (PW3) {

case ADD: Answer = number1 + number2; break;

case SUBTRACT: Answer = number1 - number2; break;

case MULTIPLY: Answer = number1 \* number2; break;

case DIVISION:

if (number2 != 0)

Answer = number1 / number2;

else

Answer = 0; // handle divide by zero

break;

}

Lcd\_Cmd(\_LCD\_CLEAR);

Lcd\_Out(1, 1, "Answer =");

IntToStr(Answer, answerx);

Lcd\_Out(2, 1, answerx);

// Wait for clear (C) key to reset

while (1) {

kp = Keypad\_Key\_Click();

Delay\_ms(50);

if (kp == 1) break; // Press key 1 (e.g., mapped as C) to clear

}

ResetCalculator();

}

}

}

**SIMULATION SCREENSHOTS**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A computer screen shot of a circuit board

AI-generated content may be incorrect.**

**A computer screen shot of a computer

AI-generated content may be incorrect.**