**Microcontroller-Based Simple Calculator**

STUDENT NAME: G.P.D. THAMARA

STUDENT NUMBER: EC/2021/005

**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 = 1; break;

case 2: val = 4; break;

case 3: val = 7; break;

case 4: ResetCalculator(); continue;

case 5: val = 2; break;

case 6: val = 5; break;

case 7: val = 8; break;

case 8: val = 0; break;

case 9: val = 3; break;

case 10: val = 6; break;

case 11: val = 9; break;

case 12: val = '='; break;

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

}

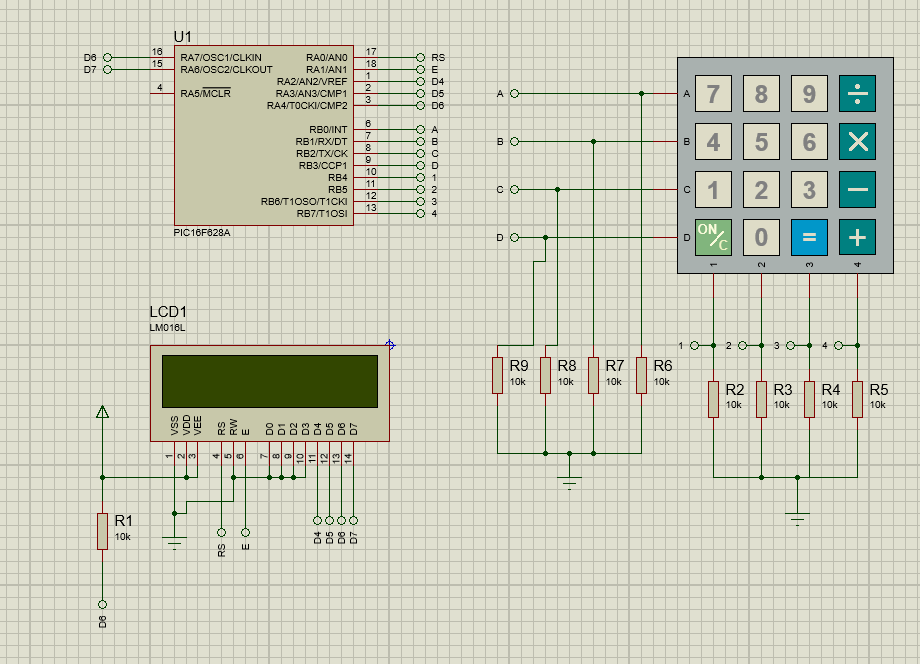
ResetCalculator();

}

}

}

**Circuit**

****

**Observations**

A screenshot of a computer

AI-generated content may be incorrect.

**Discussion**

In this experiment, we successfully interfaced a 16×2 LCD and a 4×4 keypad with the PIC16F628A microcontroller using MikroC. The primary objective was to input characters via the keypad and display them on the LCD screen using ASCII encoding. The LCD functions were initialized through the MikroC LCD library, along with additional essential libraries such as Conversion, Keypad4x4, and C\_String, which facilitated smooth interaction between hardware and software components.

The 4×4 keypad matrix, consisting of rows and columns, was scanned using a polling method. By systematically setting each row low and checking the column inputs, the microcontroller could identify the specific key pressed. Each key press corresponded to an ASCII code, which was then sent to the LCD for display. For example, pressing '1' transmitted ASCII code 49, and the LCD correctly rendered the character '1'. Arithmetic operators like '+', '-', '\*', and '/' also produced the expected symbols through their respective ASCII values.

This exercise reinforced the importance of understanding ASCII encoding, keypad scanning, and LCD interfacing techniques in embedded systems. It also highlighted how efficient library usage in MikroC simplifies the process of developing interactive microcontroller-based applications, particularly those involving text-based inputs and displays.