AMRITA VISHWA VIDYAPEETHAM

Amrita School of Engineering

Amritapuri Campus, Kollam - 690525



**19CSE303 Embedded Systems**

**Project Report (Group 2)**

**Real-Time Calculator**

**on LPC2148**

Group Members:

|  |  |
| --- | --- |
| ROHITH KRISHNAN | AM.EN.U4CSE21049 |
| AISHWARYA B NAIR | AM.EN.U4CSE21006 |
| SANGEETH SASIDHARAN | AM.EN.U4CSE21051 |
| ALFIN JOSEPH | AM.EN.U4CSE21008 |
| M. CHAITANYA RAJU | AM.EN.U4CSE21034 |

**Contents**

|  |  |  |
| --- | --- | --- |
| Sl. No | Title | Page No. |
| 1 | Abstract | 3 |
| 2 | Components Used | 4 |
| 3 | Block Diagram | 5 |
| 4 | Working | 6 |
| 5 | Code | 7 |
| 6 | Results | 14 |
| 7 | Conclusion | 16 |

**Abstract**

Objective:

The project utilizes the capabilities of the LPC2148 microcontroller to process user input from the keypad, perform mathematical operations on the entered numbers, and display the results on the LCD screen. This design allows for a basic, functional calculator experience.

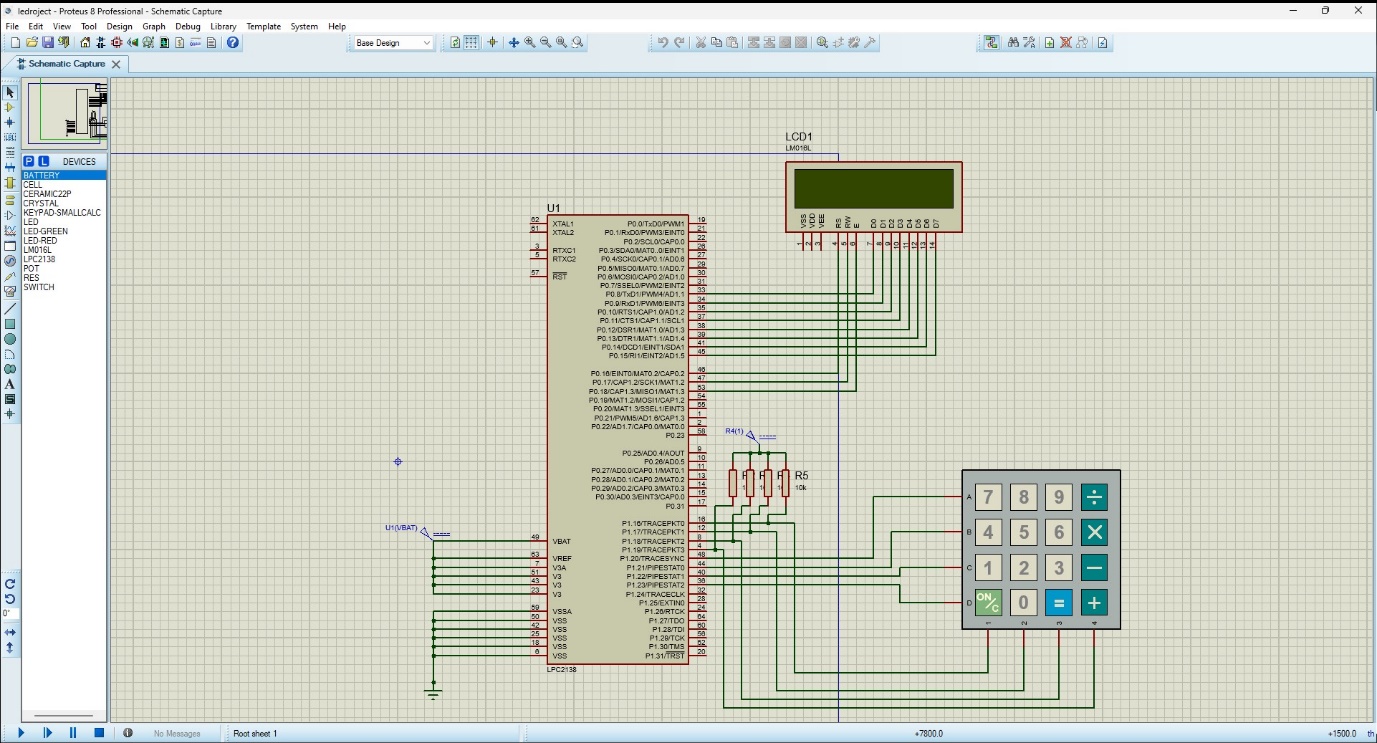
Applications:

The calculator project utilizing the LPC2148 microcontroller emerges as a versatile and indispensable tool across diverse domains, showcasing its profound impact on advancing technological innovations and enhancing operational efficiencies. In educational settings, it serves as a pivotal educational aid, fostering hands-on learning experiences in embedded systems and electronic design principles. Within the financial sector, the system facilitates complex financial analyses, risk assessments, and strategic decision-making processes, driving financial initiatives and planning strategies. In industrial automation applications, the calculator project enables precise control and monitoring of manufacturing processes, machinery operations, and production workflows, leading to enhanced productivity and cost savings. Furthermore, in healthcare management, it supports accurate medical assessments, treatment planning, and patient care coordination, thereby improving healthcare outcomes and patient safety. Additionally, within home automation, the system provides smart control solutions for managing household appliances, security systems, and energy consumption, transforming traditional homes into intelligent living spaces. Lastly, in research and development sectors, it serves as a valuable asset for scientists, engineers, and innovators, facilitating experimental simulations, data analysis, and prototype testing, fostering scientific discoveries and industry innovations. Collectively, these applications underscore the calculator project's versatility, significance, and potential to drive technological advancements across multiple industries and sectors.

**Components**

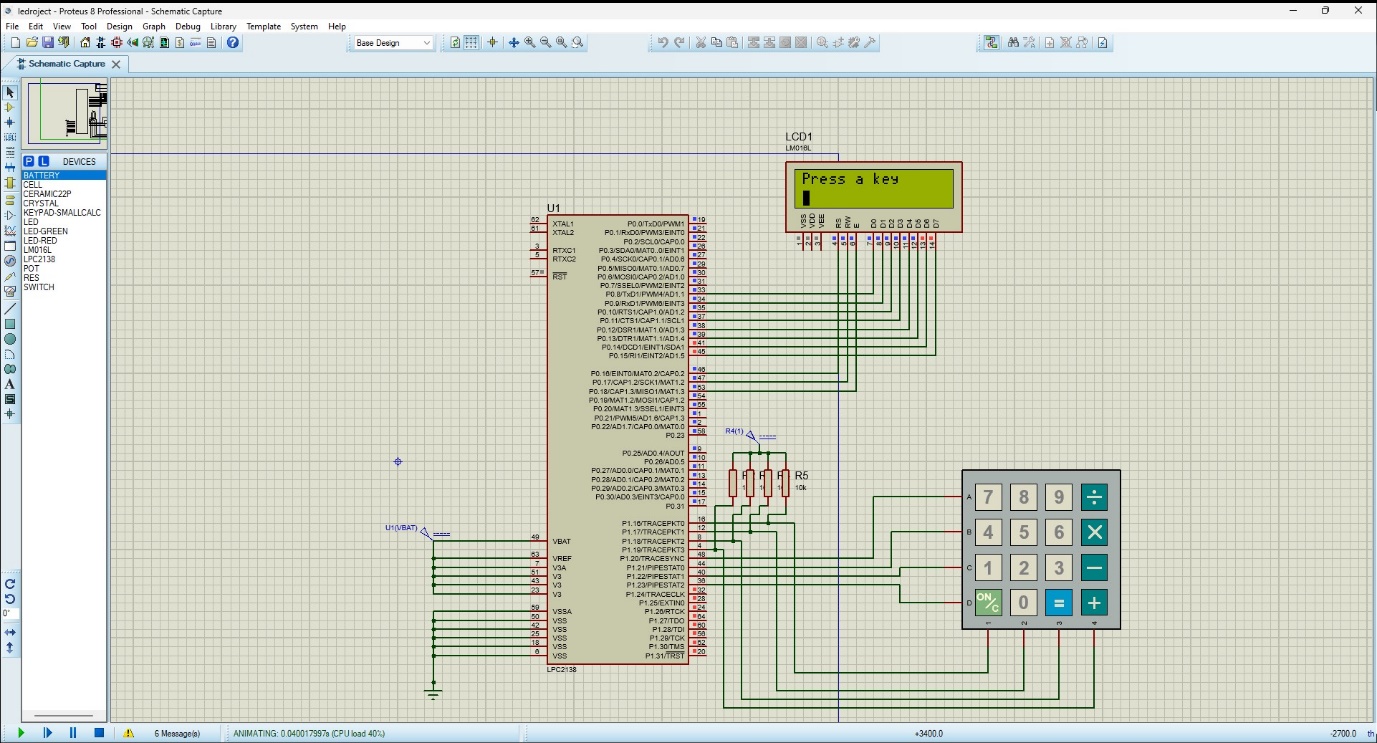
* Keypad-SmallCalc: The custom-designed Keypad-SmallCalc circuit board stands as the user interface hub of the calculator. This integral component enables users to input numerical values and operational commands effortlessly, while the integrated display visually communicates the results of computations, offering a user-friendly and efficient interaction experience.
* LM016L: The LM016L is a 16x2 character LCD display module used in the calculator circuit. It serves as the visual interface, displaying alphanumeric characters and computation results.
* LPC2148: Serving as the computational brain of the calculator, the LPC2148 microcontroller is the central processing unit (CPU). It adeptly manages the fetching, decoding, and execution of instructions, orchestrating the calculator's operations with precision and efficiency.
* Resistors: Strategically placed resistors within the circuit play a multifaceted role. They not only limit the current to LEDs but also act as protective elements for the microcontroller, contributing to the overall stability and safety of the calculator.

**Block diagram**



The calculator system consists of a complex electronic circuit that connects a LPC2148 microcontroller, a Keypad-SmallCalc, and an LM016L LCD display. It performs basic arithmetic operations using a complex electronic circuit. The circuit consists of several components that are interconnected by wires. The LPC2148 microcontroller acts as the main controller that handles the input and output signals. The LM016L LCD display serves as the visual interface, displaying computation results with clarity. It can display alphanumeric characters, symbols, and custom characters. The Keypad-SmallCalc allows the user to enter the input for the calculator. The keypad is connected to the microcontroller through the data lines, which are coloured green in the diagram. The LPC2148 microcontroller functions as the computational epicentre, efficiently managing tasks such as instruction fetching, decoding, and execution. The LEDs illuminate to signal the activation status and contribute to the overall user experience by providing visual cues and feedback during various calculator operations. These resistors enhance the safety and reliability of the calculator system. Additionally, an electronic switch serves as a pivotal component for controlling the flow of electric current, enabling the efficient activation and deactivation of the calculator. The calculator system works by interpreting the keystrokes from the keypad, processing the information, and displaying the results on the LCD, demonstrating a sophisticated interplay between user interaction and electronic operations. This carefully designed configuration ensures optimal performance and smooth communication within the calculator system.

**Working**



Pin Configuration for LCD:

* **Pin P0.16 (GPIO 16):** Register Select (RS)
* **Pin P0.17 (GPIO 17):** Read/Write (RW)
* **Pin P0.18 (GPIO 18):** Enable (EN)
* **P0.8 to P0.15 (GPIO 8 to GPIO 15):** Data Pins

Pin Configuration for Keypad:

* **P1.16 (GPIO 16):** Column 1 (col1)
* **P1.17 (GPIO 17):** Column 2 (col2)
* **P1.18 (GPIO 18):** Column 3 (col3)
* **P1.19 (GPIO 19):** Column 4 (col4)
* **P1.20 (GPIO 20):** Row 1 (row1)
* **P1.21 (GPIO 21):** Row 2 (row2)
* **P1.22 (GPIO 22):** Row 3 (row3)
* **P1.23 (GPIO 23):** Row 4 (row4)

Upon initialization, the user interface of the calculator presents a prompt on the LM016L LCD display, inviting the user to press a key on the 4x4 Keypad-SmallCalc for input. As users engage with the keypad, selecting numerical values, operators, or command keys, the LPC2148 microcontroller takes centre stage. Functioning as the central processing unit, it interprets the signals generated by the Keypad-SmallCalc, efficiently decoding the pressed key to identify the corresponding input.

The microcontroller, having deciphered the user input, proceeds to perform the necessary arithmetic operations and data processing. This involves executing calculations with precision and managing intermediate results according to the programmed logic within the microcontroller. The results of these computations are then dynamically displayed on the LM016L LCD, providing users with a real-time visual representation of the ongoing calculations.

Throughout this operational cycle, the calculator system offers visual cues and updates through LED indicators and the LCD display. This visual feedback not only indicates the operational status but also ensures that users are continuously informed about the results of their inputs. The user interaction loop remains open, allowing users to input new values, operators, or commands, fostering an interactive and iterative calculation environment.

The comprehensive orchestration of the Keypad-SmallCalc, LPC2148 microcontroller, and LM016L LCD display establishes a seamless and iterative process, translating user input into calculated results and providing real-time feedback on the calculator's display.

[**Code**](https://amritauniv-my.sharepoint.com/:u:/g/personal/amenu4cse21006_am_students_amrita_edu/EbmUJM_QDLlFllgZDeYZpIUBBnAmpPXC9y4Tel6BUGpnmQ?e=ygnBoT)

#include <lpc21xx.h>

#define LCD (0xff << 8)

#define RS (1 << 16)

#define RW (1 << 17)

#define EN (1 << 18)

#define col1 (1 << 16)

#define col2 (1 << 17)

#define col3 (1 << 18)

#define col4 (1 << 19)

#define row1 (1 << 20)

#define row2 (1 << 21)

#define row3 (1 << 22)

#define row4 (1 << 23)

void delay(unsigned int time);

void lcd\_ini(void);

void lcd\_print(char \*str);

void lcd\_cmd(unsigned char command);

void lcd\_dat(unsigned int data);

unsigned char keypad(void);

void keypad\_delay(void);

void addDigits(char a, char b, char\* result) {

    int intA = a - '0';

    int intB = b - '0';

    int sum = intA + intB;

    if (sum > 9) {

        result[0] = '1'

        result[1] = (sum % 10) + '0';

        result[2] = '\0';

    } else {

        result[0] = sum + '0';

        result[1] = '\0';

    }

}

void subtractDigits(char a, char b, char\* result) {

    int intA = a - '0';

    int intB = b - '0';

    int diff = intA - intB;

    if (diff < 0) {

        result[0] = '-';

        result[1] = (-diff) + '0';

        result[2] = '\0';

    } else {

        result[0] = diff + '0';

        result[1] = '\0';

    }

}

void multiplyDigits(char a, char b, char\* result) {

    int intA = a - '0';

    int intB = b - '0';

    int product = intA \* intB;

    result[0] = (product / 10) + '0';

    result[1] = (product % 10) + '0';

    result[2] = '\0';

}

void divideDigits(char a, char b, char\* result) {

    int intA = a - '0';

    int intB = b - '0';

    if (intB == 0) {

        result[0] = 'E';

        result[1] = '\0';

    } else {

        int quotient = intA / intB;

        int remainder = intA % intB;

                if (remainder != 0 ){

        result[0] = quotient + '0';

        result[1] = remainder + '0';

        result[2] = '\0';

                }{

                    result[0]=quotient+'0';

                    result[1]='\0';

                }

    }

}

int main(void)

{

    unsigned char a;

    int b=-1,t=-1,k=-1,c=0;

    char result[3];

    PINSEL0 = 0x00000000;

    IODIR0 = 0xffffffff;

    PINSEL1 = 0x00000000;

    IODIR1 = 0X00f00000;

    lcd\_ini();

    lcd\_print("Press a key");

    lcd\_cmd(0xc0);

    while (1)

    {

              a = keypad();

        lcd\_dat(a);

                keypad\_delay();

                 if (b == -1){

                    b = a;

                }

                else if(t == -1){

                    t = a;

                }

                else if (b != -1 && t != -1 ){

                    k = a;

                    if ((char)t == '+')

                    {

                        lcd\_ini();

                        addDigits(b, k, result);

                        b = -1;

                        t = -1;

                        k=-1;

                        lcd\_print(result);

                        keypad\_delay();

                    }

                    else if((char)t == '-'){

                        lcd\_ini();

                        subtractDigits(b, k, result);

                        b = -1;

                        t = -1;

                        k=-1;

                        lcd\_print(result);

                        keypad\_delay();

                    }

                    else if ((char)t == '\*'){

                        lcd\_ini();

                        multiplyDigits(b, k, result);

                        b = -1;

                        t = -1;

                        k = -1;

                        lcd\_print(result);

                        keypad\_delay();

                    }

                    else if((char)t == '/'){

                        lcd\_ini();

                        divideDigits(b, k, result);

                        b = -1;

                        t = -1;

                        k=-1;

                        lcd\_print(result);

                        keypad\_delay();

                }

                }

    }

    return 0;

}

void keypad\_delay(void)

{

    unsigned int t1, t2;

    for (t1 = 0; t1 < 500; t1++)

        for (t2 = 0; t2 < 1275; t2++)

            ;

}

unsigned char keypad(void)

{

    unsigned char key;

    IOCLR1 |= (row1 | row2 | row3 | row4 | col1 | col2 | col3 | col4);

    while (1)

    {

        IOCLR1 |= row1;

        IOSET1 = (row2 | row3 | row4);

        if ((IOPIN1 & col1) == 0)

        {

            key = '7';

            keypad\_delay();

            return key;

        }

        else if ((IOPIN1 & col2) == 0)

        {

            key = '8';

            keypad\_delay();

            return key;

        }

        else if ((IOPIN1 & col3) == 0)

        {

            key = '9';

            keypad\_delay();

            return key;

        }

        else if ((IOPIN1 & col4) == 0)

        {

            key = '/';

            keypad\_delay();

            return key;

        }

        IOCLR1 |= row2;

        IOSET1 = (row1 | row3 | row4);

        if ((IOPIN1 & col1) == 0)

        {

            key = '4';

            keypad\_delay();

            return key;

        }

        else if ((IOPIN1 & col2) == 0)

        {

            key = '5';

            keypad\_delay();

            return key;

        }

        else if ((IOPIN1 & col3) == 0)

        {

            key = '6';

            keypad\_delay();

            return key;

        }

        else if ((IOPIN1 & col4) == 0)

        {

            key = '\*';

            keypad\_delay();

            return key;

        }

        IOCLR1 |= row3;

        IOSET1 = (row1 | row2 | row4);

        if ((IOPIN1 & col1) == 0)

        {

            key = '1';

            keypad\_delay();

            return key;

        }

        else if ((IOPIN1 & col2) == 0)

        {

            key = '2';

            keypad\_delay();

            return key;

        }

        else if ((IOPIN1 & col3) == 0)

        {

            key = '3';

            keypad\_delay();

            return key;

        }

        else if ((IOPIN1 & col4) == 0)

        {

            key = '-';

            keypad\_delay();

            return key;

        }

        IOCLR1 |= row4;

        IOSET1 = (row1 | row2 | row3);

        if ((IOPIN1 & col1) == 0)

        {

            lcd\_cmd(0x01);

            keypad\_delay();

        }

        else if ((IOPIN1 & col2) == 0)

        {

            key = '0';

            keypad\_delay();

            return key;

        }

        else if ((IOPIN1 & col3) == 0)

        {

            key = '=';

            keypad\_delay();

            return key;

        }

        else if ((IOPIN1 & col4) == 0)

        {

            key = '+';

            keypad\_delay();

            return key;

        }

    }

}

void lcd\_cmd(unsigned char command)

{

    IO0CLR |= (RS | RW | EN | LCD);

    IO0SET |= (command << 8);

    IO0CLR |= RS;

    IO0CLR |= RW;

    IO0SET |= EN;

    delay(2);

    IO0CLR |= EN;

    delay(3);

}

void lcd\_dat(unsigned int data)

{

    IO0CLR |= (RS | RW | EN | LCD);

    IO0SET |= (data << 8);

    IO0SET |= RS;

    IO0CLR |= RW;

    IO0SET |= EN;

    delay(2);

    IO0CLR |= EN;

    delay(3);

}

void lcd\_print(char \*str)

{

    while (\*str != '\0')

    {

        lcd\_dat(\*str);

        str++;

    }

}

void delay(unsigned int time)

{

    unsigned int t1, t2;

    for (t1 = 0; t1 < time; t1++)

        for (t2 = 0; t2 < 1275; t2++)

            ;

}

void lcd\_ini(void)

{

    delay(5);

    lcd\_cmd(0X38);

    lcd\_cmd(0X0f);

      lcd\_cmd(0X06);

    lcd\_cmd(0X01);

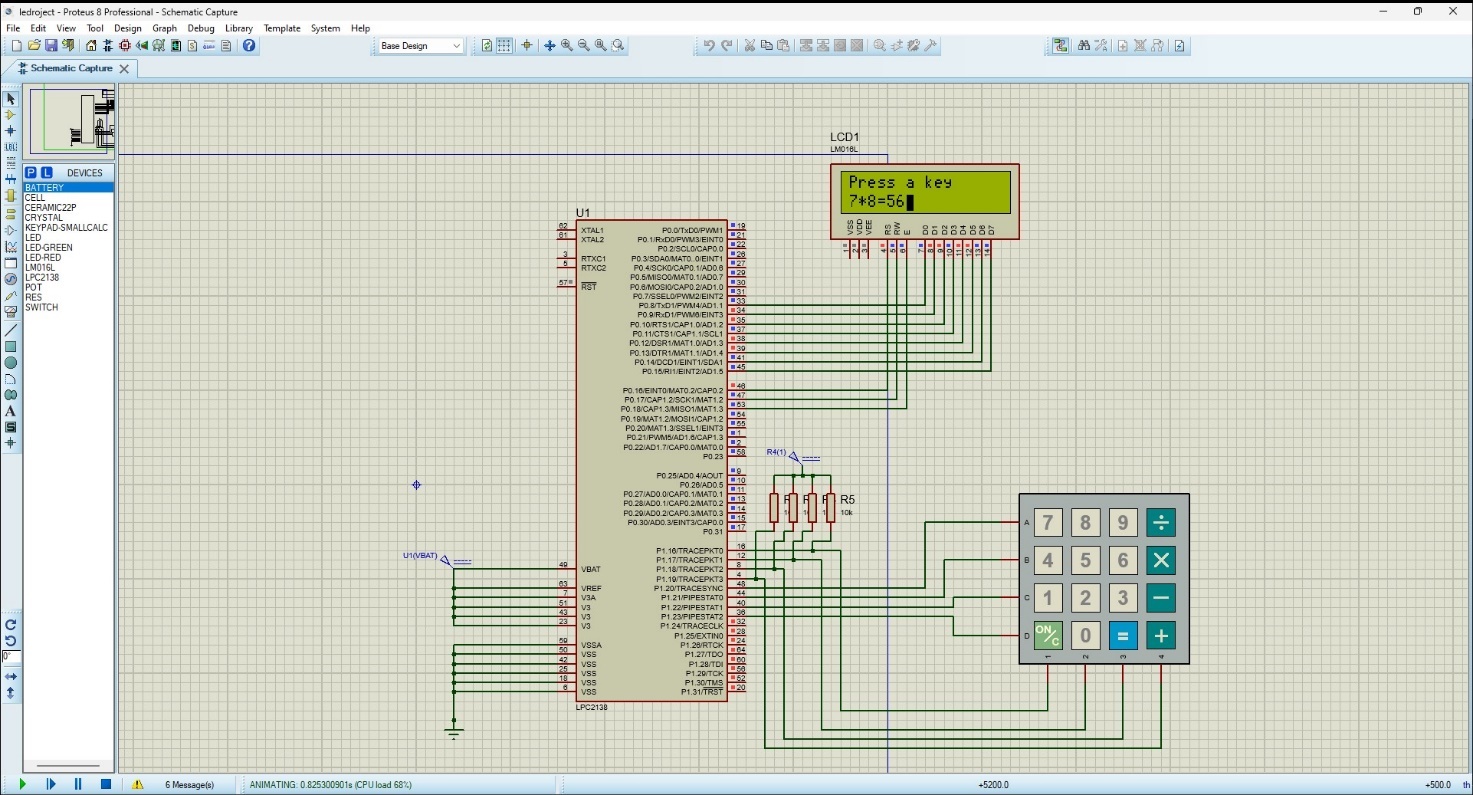
    delay(5);

    lcd\_cmd(0X80);

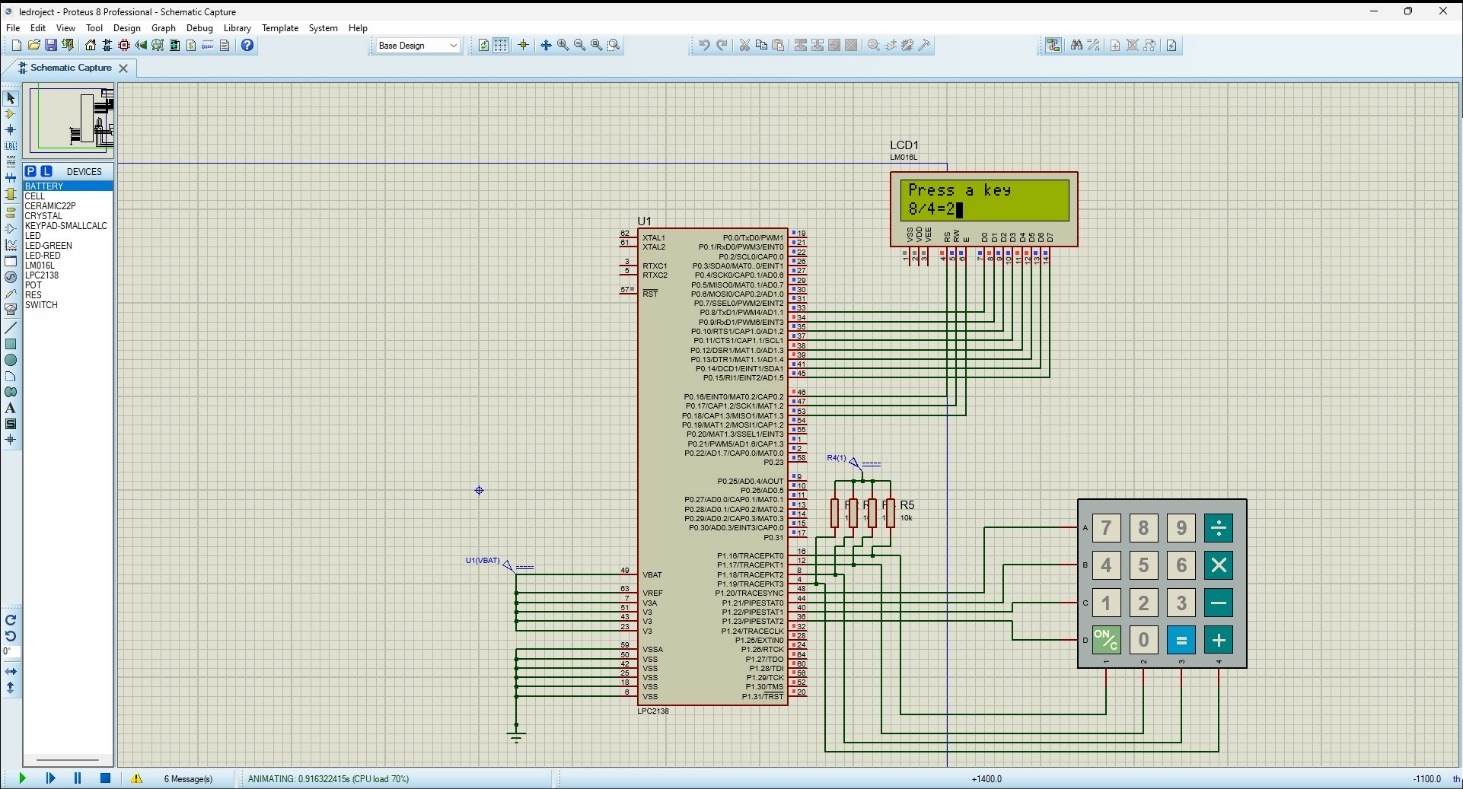
}

**Results**

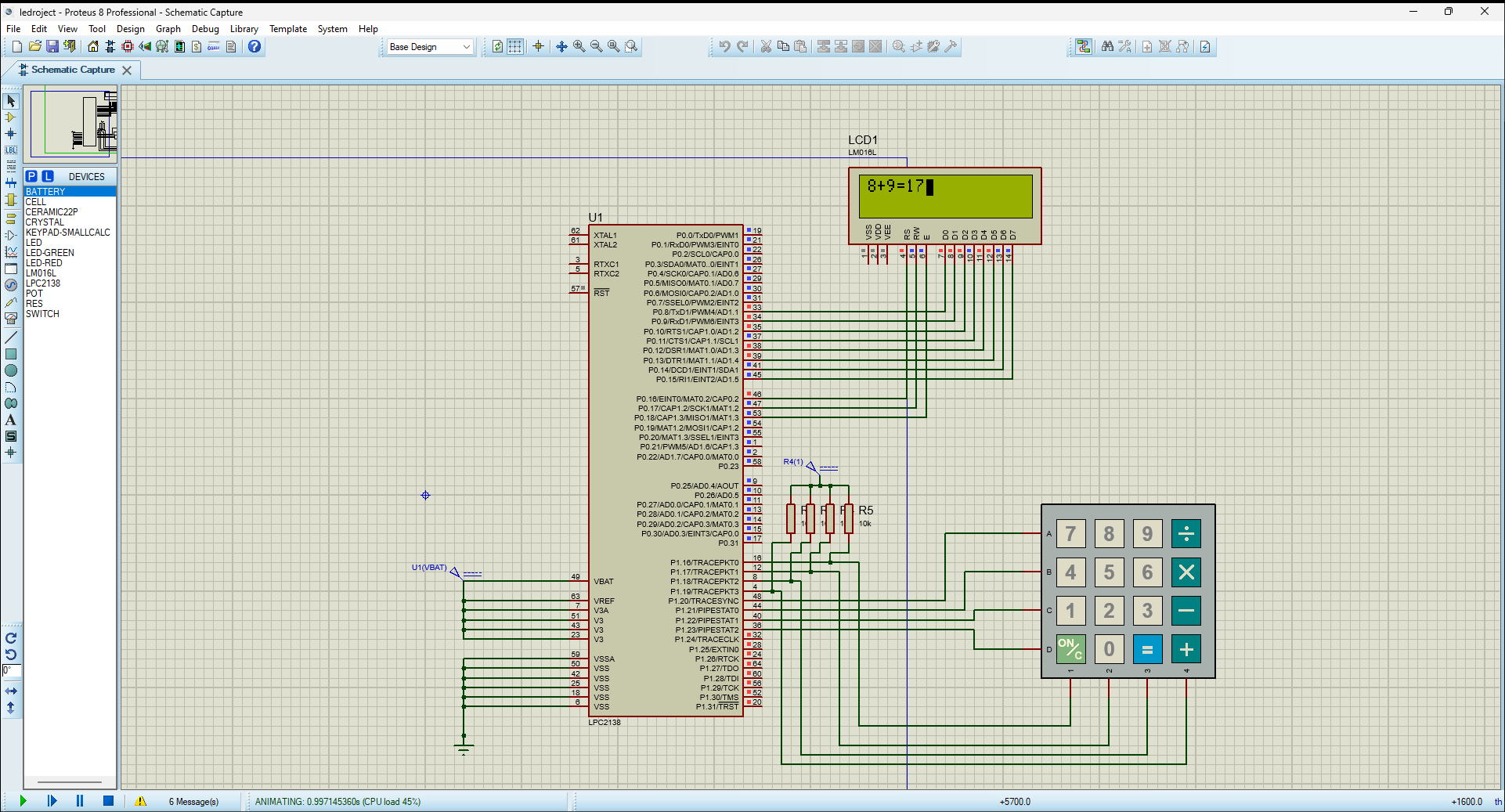
Multiplication:



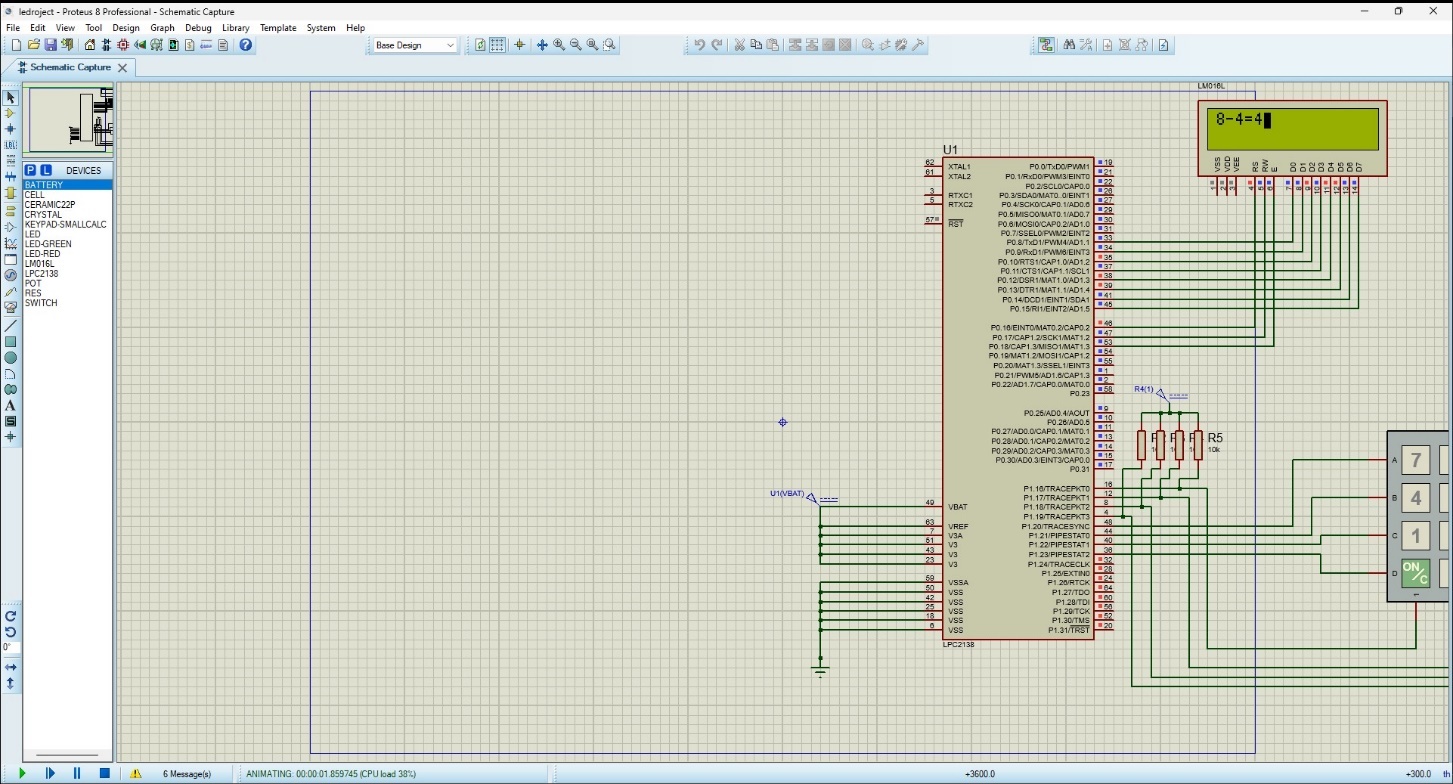
Division:



Addition:



Subtraction:



**Conclusion**

The calculator project showcases the intricate design and harmonious integration of the Keypad-SmallCalc, LPC2148 microcontroller, and LM016L LCD display. The system creates a user-friendly and interactive sequence of operations: the LM016L LCD displays a prompt for user input from the 4x4 Keypad-SmallCalc, where the LPC2148 microcontroller accurately decodes the keystrokes. This central processing unit then performs the calculations and data processing, and shows the results on the LM016L LCD, thus providing a responsive and engaging user experience.

The project’s success relies on the LPC2148 microcontroller’s remarkable power. Based on the ARM7TDMI-S architecture, it has a clock frequency of up to 60 MHz, which enables fast execution of instructions and effective multitasking. This computational power is supported by the microcontroller’s various peripherals, such as UART, SPI, and I2C, which enhance its flexibility for different applications.

In summary, this calculator project, with its careful design and use of the LPC2148 microcontroller, exemplifies the combination of electronic engineering and programming skills. It not only performs complex arithmetic operations with precision but also offers users an intuitive and responsive interface, illustrating the potential of the LPC2148 microcontroller in developing sophisticated embedded systems.