

SPRAWOZDANIE					PROSZĘ PODAĆ NR GRUPY:						
					ZIISS1	3	5	1	2	IO	
IMIĘ	NAZWISKO	Temat ćwiczenia zgodny z wykazem tematów:			PONIŻEJ PROSZĘ PODAĆ TERMIN ZAJĘĆ:			ROK:			
MICHAŁ	WARSZAWSKI	Podstawy programowania Kalkulator z wyświetlaczem LCD						2023 r.			
					PN	WT	SR	CZ	PT	SB	ND
					GODZINA ROZPOCZĘCIA ZAJĘĆ:						11 : 30

Wprowadzenie teoretyczne:

Opisz rodzaje klawiatur wykorzystywanych w technice mikroprocesorowej

Istnieje kilka rodzajów klawiatur wykorzystywanych w tym kontekście, zależnie od ich konstrukcji i zastosowania.

Membranowe klawiatury: Te klawiatury składają się z cienkich warstw membran, które są rozmieszczone pomiędzy dwiema foliami przewodzącymi. Kiedy przycisk jest naciśnięty, warstwy membrany się stykają, co zamyka obwód i wysyła sygnał do mikroprocesora. Klawiatury membranowe są popularne ze względu na ich niski koszt produkcji i płaski profil.

Mechaniczne klawiatury: Te klawiatury używają mechanicznych przełączników pod klawiszami. Przełączniki te mają fizyczne sprężyny i styki, co daje użytkownikowi taktyczne odczucie i charakterystyczny dźwięk przy naciśnięciu klawisza. Klawiatury mechaniczne są cenione przez osoby, które preferują precyzyjne i wygodne odczucie podczas pisania.

Klawiatury dotykowe: Nie posiadają fizycznych klawiszy, zamiast tego wykorzystują ekran dotykowy do rejestracji nacisków palców. Są powszechnie stosowane w urządzeniach mobilnych, takich jak smartfony, tablety czy ekranowe terminale. Klawiatury dotykowe są intuicyjne, ale nie zawsze oferują takie same odczucia jak tradycyjne klawiatury.

Klawiatury scissor-switch: Są to klawiatury, które łączą pewne cechy klawiatur membranowych i mechanicznych. Posiadają klawisze oparte na mechanicznych przełącznikach, ale mają też specjalny mechanizm złącza nożycowego, co sprawia, że są one bardziej płaskie i wygodne do użytku.

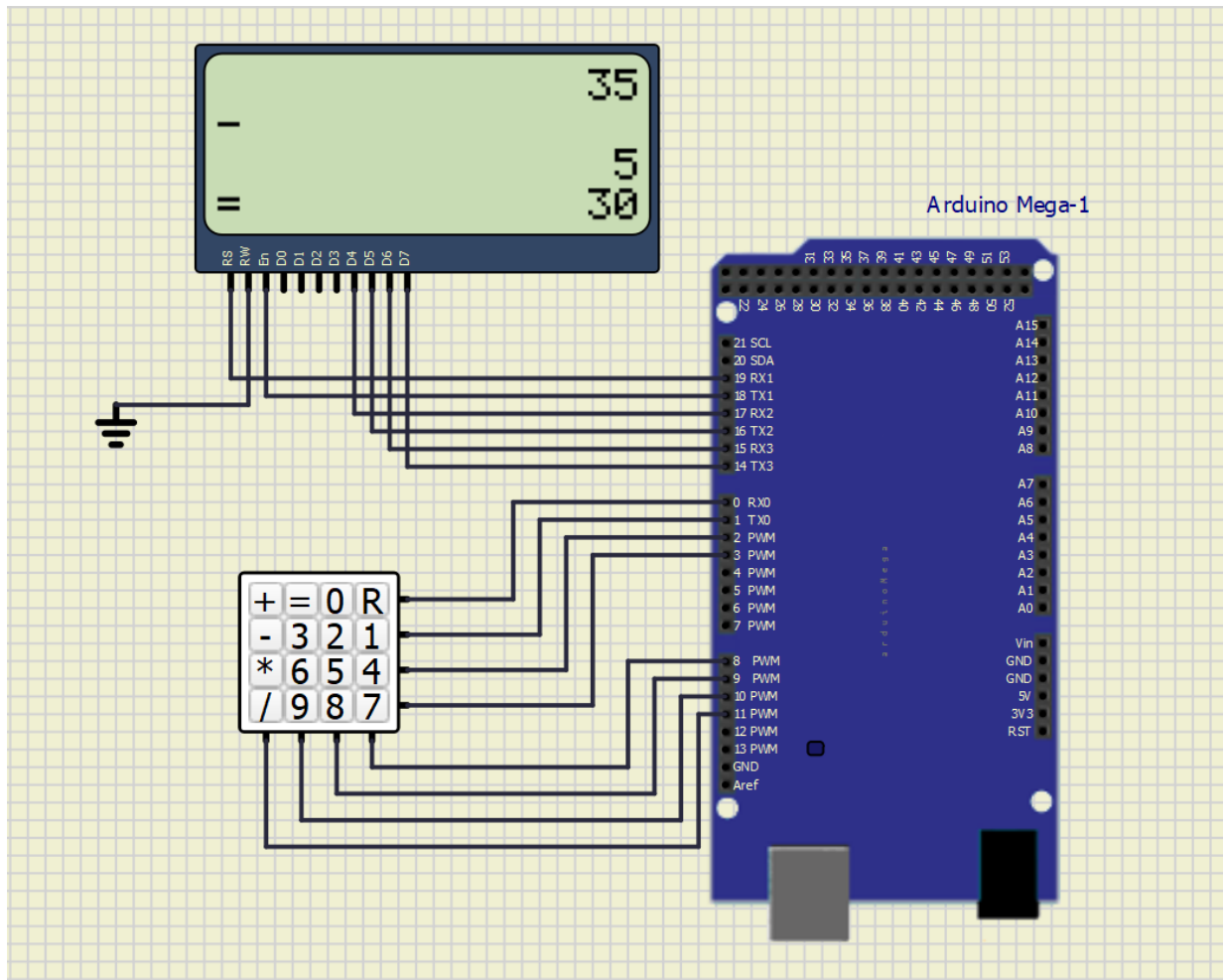
Klawiatury elastomerowe: Te klawiatury używają elastycznych gumowych klawiszy, które są naciskane, aby zamknąć obwód i przekazać sygnał. Są stosunkowo tanie i mogą być używane w różnych aplikacjach.

Klawiatury z touchpadem lub trackballem: Niektóre klawiatury mikroprocesorowe zawierają dodatkowe urządzenia w postaci touchpadów lub trackballi, umożliwiające obsługę kursora na ekranie. To szczególnie przydatne w przypadku urządzeń, gdzie mysz może być niewygodna lub niemożliwa do użycia.

Zadanie 1

Zaprojektuj układ przedstawiony na rysunku. W skład układu wchodzi klawiatura numeryczna 4x4 wyświetlacz LED 16x5 oraz płyta Andino Mega z procesorem 2560. Zaprogramuj kalkulator, w którym możliwe będzie wykonanie wszystkich podstawowych operacji arytmetycznych przedstawionych na klawiaturze.

Schemat układu:



Listing programu:

```
#include <LiquidCrystal.h>
#include <Keypad.h>

// Inicjalizacja obiektu LiquidCrystal
const int rs = 19, en = 18, d4 = 17, d5 = 16, d6 = 15, d7 = 14;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// Konfiguracja klawiatury
const byte ROWS = 4;
const byte COLS = 4;
byte rowPins[ROWS] = {0, 1, 2, 3};
byte colPins[COLS] = {8, 9, 10, 11};

// Definicja klawiszy na klawiaturze
char keys[ROWS][COLS] = {
  {'R', '0', '=', '+'},
  {'1', '2', '3', '-'},
```

```
{'4', '5', '6', '*'},
{'7', '8', '9', '/'}];

// Inicjalizacja obiektu Keypad
Keypad klawiatura = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

// Deklaracje zmiennych globalnych
int firstNums[8];
int secondNums[8];
char sign[1];
int firstLp = 0;
int secondLp = 0;
bool letFirstNum = true;
bool letSign = false;
bool letSecondNum = false;

void setup()
{
    // Konfiguracja pinów jako wejścia
    for (int i = 0; i < ROWS; ++i)
    {
        pinMode(rowPins[i], INPUT);
        pinMode(colPins[i], INPUT);
    }

    // Inicjalizacja ekranu LCD
    lcd.begin(16, 4);
    lcd.setCursor(15, 0);
    lcd.print("0");
}

void loop()
{
    // Odczyt klawisza z klawiatury
    char klawisz = klawiatura.getKey();
    if (klawisz != NO_KEY)
    {
        // Obsługa klawiszy 0-9
        if (klawisz >= '0' && klawisz <= '9')
        {
            // Logika dla pierwszej liczby
            if (letFirstNum == true && firstLp < 8)
            {
                if (letSign == false)
                {
                    lcd.clear();
                }
                firstNums[firstLp] = convertToNumber(klawisz);
                firstLp++;
                lcd.setCursor(16 - firstLp, 0);
                for (int i = 0; i < firstLp; ++i)
                {
                    lcd.print(firstNums[i]);
                }
                letSign = true;
            }

            // Logika dla drugiej liczby
            if (letSecondNum == true && secondLp < 8)
            {
                secondNums[secondLp] = convertToNumber(klawisz);
                secondLp++;
                lcd.setCursor(16 - secondLp + 4, 2);
                for (int j = 0; j < secondLp; ++j)
                {
```

```
        lcd.print(secondNums[j]);
    }
    letSign = false;
}
}
// Obsługa klawiszy +, -, *, /
else if (klawisz == '+' || klawisz == '-' || klawisz == '*' || klawisz == '/')
{
    if (letSign == true)
    {
        sign[0] = klawisz;
        lcd.setCursor(0, 1);
        lcd.print(sign[0]);
        letFirstNum = false;
        letSecondNum = true;
    }
}
// Obsługa klawisza "="
else if (klawisz == '=' && letSign == false)
{
    lcd.setCursor(4, 3);
    lcd.print("=");

    // Obliczenia i wyświetlanie wyniku
    int result = calculate();
    if (result == 0)
    {
        lcd.setCursor(16, 3);
        lcd.print('0');
    }
    else
    {
        int length = 0;
        int temp = result;
        while (temp != 0)
        {
            temp /= 10;
            ++length;
        }
        lcd.setCursor(20 - length, 3);
        lcd.print(result);
    }

    // Resetowanie zmiennych po obliczeniach
    letFirstNum = true;
    letSign = false;
    letSecondNum = false;
    clearArrays();
}
// Obsługa klawisza "R" (reset)
else if (klawisz == 'R')
{
    lcd.clear();
    letFirstNum = true;
    letSign = false;
    letSecondNum = false;
    clearArrays();
}
}

// Funkcja do przeliczania znaków na liczby
int convertToNumber(int klawisz)
{
    return klawisz - '0';
}
```

```
// Funkcja do obliczeń matematycznych
int calculate()
{
    long first = 0;
    long second = 0;
    long ten = 1;

    // Przeliczanie pierwszej liczby
    for (int i = firstLp - 1; i >= 0; --i)
    {
        first = first + firstNums[i] * ten;
        ten *= 10;
    }

    ten = 1;

    // Przeliczanie drugiej liczby
    for (int i = secondLp - 1; i >= 0; --i)
    {
        second = second + secondNums[i] * ten;
        ten *= 10;
    }

    // Obliczenia wyniku
    long result = 0;
    switch (sign[0])
    {
        case '+':
            result = first + second;
            break;
        case '-':
            result = first - second;
            break;
        case '*':
            result = first * second;
            break;
        case '/':
            // Sprawdzenie, czy druga liczba nie jest zerem przed dzieleniem
            if (second != 0)
            {
                result = first / second;
            }
            else
            {
                // Wyświetlenie komunikatu o błędzie
                lcd.setCursor(0, 3);
                lcd.print("Error");
                delay(2000);
                lcd.clear();
                return 0;
            }
            break;
        default:
            break;
    }

    return result;
}

// Funkcja do czyszczenia tablic
void clearArrays()
{
    // Wyzerowanie tablic pierwszej i drugiej liczby
    for (int i = firstLp - 1; i >= 0; --i)
    {
```

```
        firstNums[i] = 0;
    }
    for (int i = secondLp - 1; i >= 0; --i)
    {
        secondNums[i] = 0;
    }
    // Reset liczników
    firstLp = 0;
    secondLp = 0;
}
```

Wnioski:

Program został napisany w języku Arduino i wykorzystuje biblioteki LiquidCrystal i Keypad do obsługi wyświetlacza i klawiatury. Zdefiniowano obiekty dla LCD i klawiatury, a także zmienne globalne do przechowywania liczb, operatora i innych informacji potrzebnych do działania kalkulatora. Program obsługuje wprowadzanie cyfr, operatorów (+, -, *, /), oraz przycisków R (reset) i = (wynik). Wprowadzone liczby są przechowywane w tablicach i używane do wykonania operacji arytmetycznych. Wyświetlanie wyników odbywa się na wyświetlaczu LCD. Program korzysta z biblioteki Keypad do obsługi klawiatury numerycznej 4x4. Odczyt klawiszy odbywa się w pętli loop(), gdzie każdy naciśnięty klawisz jest interpretowany i odpowiednio reaguje na niego program. Program uwzględnia obsługę błędów, takich jak dzielenie przez zero, i wyświetla komunikat "Error" na wyświetlaczu w przypadku wystąpienia błędu. Zastosowano funkcje pomocnicze do konwersji znaków na liczby, obliczeń matematycznych oraz czyszczenia tablic. Programu realizuje podstawowe funkcje kalkulatora.