Міністерство освіти і науки України

**Прикарпатський національний університет**

**імені В.Стефаника**

*Факультет математики та інформатики*

*Кафедра інформаційних технологій*

*Програмування вбудованих систем*

Лабораторна робота № 1

Тема: «Ввід аналогових сигналів. Робота з РКІ»

Варіант : 16

Виконав: **Петрів В. В.**

Група ІПЗ-13

Дата: 11 жовтня 2023р.

Викладач: Лазарович І.М.

Івано-Франківськ - 2023

**Завдання**

Використовуючи теоретичний матеріал лекцій 1-4 скласти схему для

двохканального вимірювання напруги на основі Arduino UNO з кнопкою та

виводом на рідкокристалічний індикатор та в послідовний порт (рисунок 1).

Рисунок 1 – Рекомендована схема для виконання роботи перший канал вимірювання повинен міряти напругу з потенціометра, другий – напругу живлення Arduino, з використанням резисторного подільника, його параметри задано у таблиці 1, Uвходу=5В.

Виміряне значення напруги з кожного каналу вимірювання потрібно вивести на РКІ в 2 рядки. Частота оновлення екрану (виміряних значень) – 0.5 - 2 Гц, яка циклічно змінюється при натисканні на кнопку з кроком 0.5. Програму реалізувати без delay().

Кожен рядок на РКІ повинен містити миттєве значення напруги (3 знаки після коми), усереднене значення напруги на основі 10-100 вимірів (3 знаки після коми), відхилення між цими значеннями, розмір вибірки для усереднення.

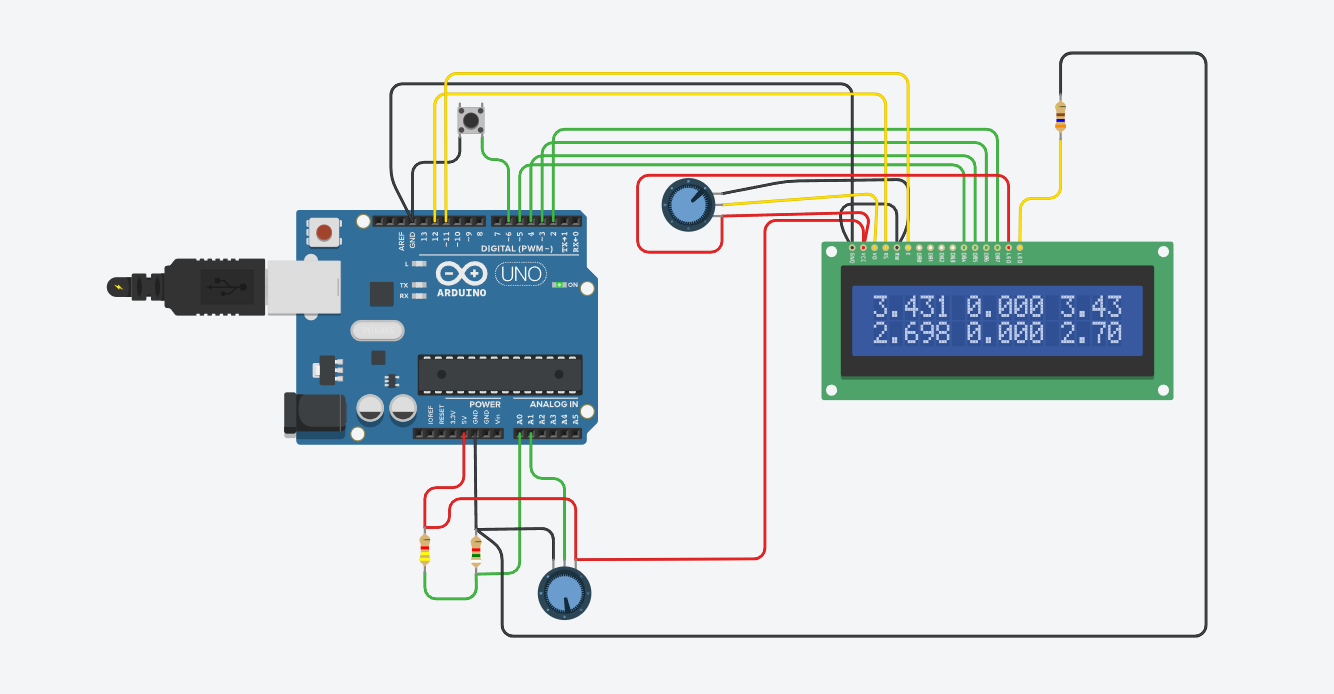
Розмір вибірки змінювати циклічно з кроком 10 після кожного виміру.

Для можливості дослідження отриманих даних разом з РКІ, інформацію у вказаному вигляді виводити також в послідовний порт.

Дослідити як змінюється це відхилення залежно від розміру вибірки для усереднення.

В звіті навести тему, сформульоване завдання, програмний код, схему з’єднань (можна фото), скріншот результатів виведення з послідовного порта (не менше 15-20 рядків), графік залежності відхилення між миттєвим та усередненим значенням від об’єму вибірки, висновки.

**Схема**

****

**Програмний код**

#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

unsigned long previousMillis = 0;

float hertz = 0.5;

double intervalHertz = (1. / hertz) \* 1000;

double sum\_p = 0, average\_p = 0, sum\_r = 0, average\_r = 0;

double current\_p, current\_r;

double k = 5. / 1023;

const long interval = 300;

const int buttonPin = 6;

int lastButtonState = HIGH;

unsigned long lastDebounceTime = 0;

unsigned long debounceDelay = 10;

int buttonState;

int size = 0, amount = 0;

void setup() {

lcd.begin(16, 2);

Serial.begin(9600);

pinMode(A0, INPUT);

pinMode(A1, INPUT);

pinMode(buttonPin, INPUT\_PULLUP);

}

void loop() {

unsigned long currentMillis = millis();

int reading = digitalRead(buttonPin);

if (reading != lastButtonState) {

lastDebounceTime = millis();

}

if ((millis() - lastDebounceTime) > debounceDelay) {

if (reading != buttonState) {

buttonState = reading;

if (buttonState == LOW) {

if (hertz == 2) {

hertz = 0.5;

intervalHertz = (1. / hertz) \* 1000;

} else {

hertz += 0.5;

intervalHertz = (1. / hertz) \* 1000;

}

}

}

}

if (currentMillis - previousMillis >= intervalHertz) {

previousMillis = currentMillis;

current\_p = analogRead(A1) \* k;

sum\_p += current\_p;

current\_r = analogRead(A0) \* k;

sum\_r += current\_r;

amount++;

if (amount % 10 == 0) {

size = amount;

average\_r = sum\_r / amount;

average\_p = sum\_p / amount;

if (size > 100) {

size = 10;

}

}

lcd.clear();

lcd.setCursor(0, 1);

lcd.print(current\_p, 3);

lcd.print(" ");

lcd.print(average\_p, 3);

lcd.print(" ");

lcd.print(current\_p - average\_p);

lcd.setCursor(0, 0);

lcd.print(current\_r, 3);

lcd.print(" ");

lcd.print(average\_r, 3);

lcd.print(" ");

lcd.print(current\_r - average\_r);

Serial.print("\n\n");

Serial.print("\ncurrent U=");

Serial.print(current\_r, 3);

Serial.print("\t average U=");

Serial.print(average\_r, 3);

Serial.print("\t diverse =");

Serial.print(current\_r - average\_r);

Serial.print("\t size =");

Serial.print(size);

Serial.print("\ncurrent U=");

Serial.print(current\_p, 3);

Serial.print("\t average U=");

Serial.print(average\_p, 3);

Serial.print("\t diverse =");

Serial.print(current\_p - average\_p);

Serial.print("\t size =");

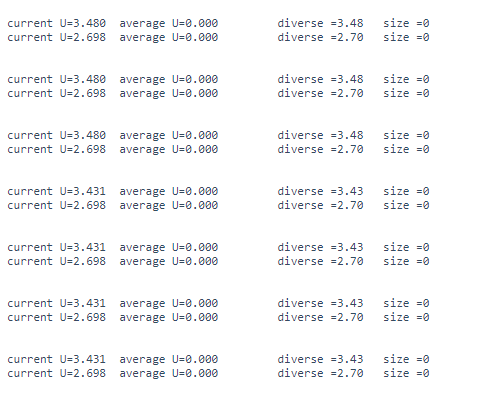
Serial.print(size);

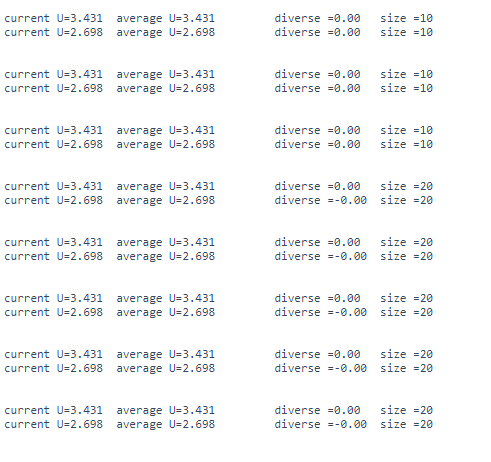
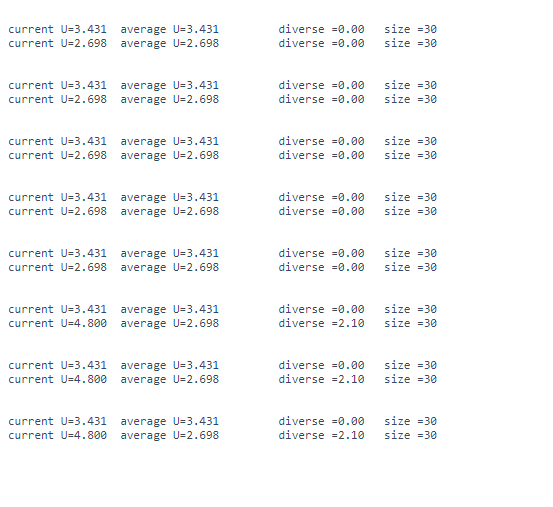
lastButtonState = reading;

}

}

**Результати виведення**









Найближче значення за рядом номіналів – 3,32 kOm

**Висновок**

Під час даної роботи ми вивчили, що таке «брязкіт» кнопки і як його позбутись, чим погане використання delay() і який алгоритм покращення. Як зчитувати дані з аналогових входів контролера Arduino, обробляти ці дані та їх виводити на рідкокристалічний індикатор на базі контролера HD44780, а також вивід у послідовний порт.