**Лабораторна робота №1**

**Виконав: Коваль С.В.**

**Група ІПЗ-21**

**Дата: 12 вересня 2024р.**

**Викладач: Ігор Лазарович**

**Івано-Франківськ – 2024**

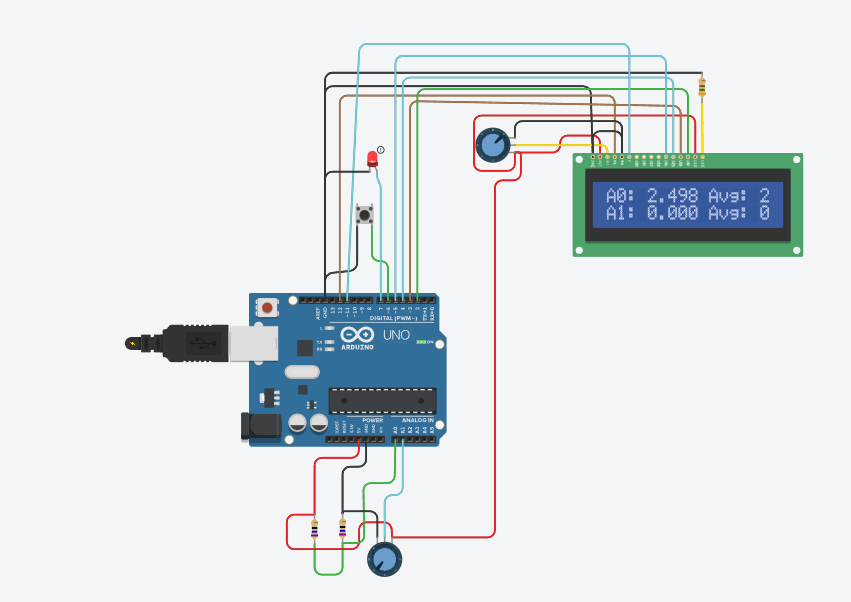
**Завдання:**

Написати програму та створити схему, що зчитує напругу з двох пінів (A0 і A1) для вимірювання опору потенціометра та обчислення коефіцієнту подільника напруги, виводить миттєві та середні значення напруги на LCD-дисплей і серійний порт. Кнопка змінює частоту оновлення між 0,5 герца і 2 герців.

Також визначити коефіцієнт подільника



**Схема:**

**Код:**

**#include <LiquidCrystal.h>**

**#define pinbtn 6**

**#define ledPin 7**

**LiquidCrystal lcd(12, 11, 5, 4, 3, 2);**

**const int maxSamples = 100;**

**int sampleCount = 10;**

**float measurementsA0[maxSamples];**

**float measurementsA1[maxSamples];**

**int currentSample = 0;**

**float instantA0 = 0;**

**float instantA1 = 0;**

**unsigned long updateInterval = 2000;**

**unsigned long lastUpdate = 0;**

**unsigned long lastdebT = 0;**

**int lastButState = HIGH;**

**float R1 = 76;**

**float K;**

**void setup() {**

**pinMode(pinbtn, INPUT\_PULLUP);**

**pinMode(ledPin, OUTPUT);**

**Serial.begin(9600);**

**lcd.begin(16, 2);**

**lcd.print("Initializing...");**

**lcd.clear();**

**}**

**void loop() {**

**unsigned long curmil = millis();**

**int reading = digitalRead(pinbtn);**

**if (curmil - lastdebT > 50) {**

**if (reading != lastButState) {**

**lastButState = reading;**

**lastdebT = curmil;**

**if (lastButState == LOW) {**

**if (updateInterval == 2000) {**

**updateInterval = 500;**

**} else {**

**updateInterval = 2000;**

**}**

**}**

**}**

**}**

**if (curmil - lastUpdate > updateInterval) {**

**instantA0 = 5.0 \* analogRead(A0) / 1023; // Миттєва напруга на A0**

**instantA1 = 5.0 \* analogRead(A1) / 1023; // Миттєва напруга на A1**

**float R\_fixed = 10000.0;**

**float R2 = (R\_fixed \* instantA1) / (5.0 - instantA1);**

**K = R2 / (R1 + R2);**

**Serial.print("\nK: ");**

**Serial.println(K, 3);**

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

**measurementsA0[currentSample] = instantA0;**

**measurementsA1[currentSample] = instantA1;**

**currentSample++;**

**if (currentSample >= sampleCount) {**

**float avgA0 = calculateAverage(measurementsA0, sampleCount);**

**float avgA1 = calculateAverage(measurementsA1, sampleCount);**

**float deviationA0 = instantA0 - avgA0;**

**float deviationA1 = instantA1 - avgA1;**

**printMeasurements(instantA0, avgA0, deviationA0, instantA1, avgA1, deviationA1);**

**displayMeasurements(instantA0, avgA0, instantA1, avgA1);**

**currentSample = 0;**

**sampleCount = (sampleCount >= maxSamples) ? 10 : sampleCount + 10; // Змінюємо розмір вибірки з кроком 10**

**}**

**lastUpdate = curmil;**

**}**

**if (updateInterval == 500) { // Якщо частота оновлення 500 мс**

**digitalWrite(ledPin, HIGH); // Увімкнути світлодіод**

**} else {**

**digitalWrite(ledPin, LOW); // Вимкнути світлодіод**

**}**

**}**

**float calculateAverage(float\* measurements, int count) {**

**float total = 0;**

**for (int i = 0; i < count; i++) {**

**total += measurements[i];**

**}**

**return total / count;**

**}**

**void printMeasurements(float instA0, float avgA0, float devA0, float instA1, float avgA1, float devA1) {**

**Serial.print("A0 Inst: ");**

**Serial.print(instA0, 3);**

**Serial.print(" V, A0 Avg: ");**

**Serial.print(avgA0, 3);**

**Serial.print(" V, Deviation: ");**

**Serial.print(devA0, 3);**

**Serial.println(" V");**

**Serial.print("A1 Inst: ");**

**Serial.print(instA1, 3);**

**Serial.print(" V, A1 Avg: ");**

**Serial.print(avgA1, 3);**

**Serial.print(" V, Deviation: ");**

**Serial.print(devA1, 3);**

**Serial.println(" V");**

**}**

**void displayMeasurements(float instA0, float avgA0, float instA1, float avgA1) {**

**lcd.clear();**

**lcd.setCursor(0, 0);**

**lcd.print("A0: ");**

**lcd.print(instA0, 3);**

**lcd.print(" Avg: ");**

**lcd.print(avgA0, 3);**

**lcd.setCursor(0, 1);**

**lcd.print("A1: ");**

**lcd.print(instA1, 3);**

**lcd.print(" Avg: ");**

**lcd.print(avgA1, 3);**

**}**