

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Навчально-науковий інститут інноваційних освітніх технологій
Кафедра інформаційно-обчислювальних систем і управління

Гуменюк Іван Ігорович

**Програмний модуль у середовищі візуального програмування для
опрацювання даних вимірювання напруги / Software module in visual
programing for processing voltage measurements**

Спеціальність 122 – Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

Кваліфікаційна робота

Виконав студент групи КНз-41
І. І. Гуменюк

Науковий керівник:
к.т.н., професор, В.В.Кочан

Кваліфікаційну роботу допущено до
захисту

«___»_____ 2026 р.

Завідувач кафедри

_____ Н.В. Дзюбановська

Тернопіль – 2026

Навчально-науковий інститут інноваційних освітніх технологій
Кафедра інформаційно-обчислювальних систем і управління
Освітній ступінь «бакалавр»
спеціальність 122 – Комп'ютерні науки
освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Н.В. Дзюбановська
«_____» _____ 20__ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ
Гуменюку Івану Ігорович
(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи: програмний модуль в середовищі візуального програмування для опрацювання даних вимірювання напруги / Software module in visual programing for processing voltage measurements

керівник роботи В.В. Кочан, к.т.н., професор
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджений наказом по університету від 01 грудня 2025 року № 894.

2. Строк подання студентом закінченої кваліфікаційної роботи 25 травня 2026 р.

3. Вихідні дані до кваліфікаційної роботи: завдання на кваліфікаційну роботу студента, наукові статті, технічна література.

4. Основні питання, які потрібно розробити:

- провести аналіз отримання та опрацювання даних вимірювання напруги;
- провести огляд середовищ візуального програмування для роботи з вимірювальними даними;
- сформулювати постановку задачі;
- розробити загальну структуру програмного модуля;
- розробити алгоритми опрацювання даних вимірювання напруги;
- розробити інформаційне забезпечення модуля;
- провести вибір середовища візуального програмування та засобів реалізації;
- розробити інтерфейс користувача програмного модуля;
- провести тестування розробленого програмного модуля.

5. Перелік графічного матеріалу в роботі:

- блок - діаграма вимірювального модуля;
- алгоритм функціонування модуля;
- алгоритм опрацюванням масиву значень напруги.

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 01 грудня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Затвердження теми кваліфікаційної роботи, ознайомлення з літературними джерелами та складання плану роботи	до 01.01.2026 р.	
2	Написання 1 розділу кваліфікаційної роботи	до 01.02.2026 р.	
3	Написання 2 розділу кваліфікаційної роботи	до 15.03.2026 р.	
4	Написання 3 розділу кваліфікаційної роботи	до 01.05.2026 р.	
5	Представлення попереднього варіанту кваліфікаційної роботи, перевірка та внесення змін керівником	до 15.05.2026 р.	
6	Опрацювання зауважень та представлення завершеного варіанту кваліфікаційної роботи. Підготовка супроводжуючих документів	до 25.05.2026 р.	
7	Перевірка кваліфікаційної роботи на оригінальність тексту	до 30.05.2026 р.	
8	Оформлення кваліфікаційної роботи та отримання допуску до захисту	до 10.06.2026 р.	
9	Подання кваліфікаційної роботи до захисту на засіданні атестаційної комісії	до 10.06.2026 р.	

Студентка _____ І. І. Гуменюк
(підпис) (прізвище та ініціали)

Керівник кваліфікаційної роботи _____ В.В. Кочан
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи: 52 с., 13 рис., 4 додатки, 35 джерел.

Метою кваліфікаційної роботи є розробка програмного модуля в середовищі візуального програмування для опрацювання даних вимірювання напруги, що забезпечує введення або завантаження значень напруги, перевірку їх коректності, обчислення основних параметрів, графічне відображення результатів, контроль допустимих меж та збереження даних у файл.

Об'єкт дослідження: процес опрацювання даних вимірювання напруги в комп'ютеризованій вимірювальній системі.

Предмет дослідження: система візуального програмування, для введення, перевірки, обчислення, графічного відображення та збереження даних.

Методи дослідження: аналіз та опрацювання даних вимірювання напруги, методи візуального програмування, структурне проектування, алгоритмічна декомпозиція, моделювання та функціональне тестування.

Результатом роботи є розроблений програмний модуль в середовищі LabVIEW для опрацювання даних вимірювання напруги. В роботі сформовано загальну структуру модуля, розроблено алгоритми функціонування та перевірки вхідних даних, визначено інформаційне забезпечення, створено інтерфейс користувача у вигляді лицьової панелі та виконано тестування основних режимів роботи. Програмний модуль забезпечує обчислення мінімального, максимального та середнього значення напруги, контроль виходу значень за допустимі межі, побудову графіка зміни напруги та збереження результатів.

Практичне значення роботи полягає в тому, що розроблений модуль може бути використаний як базовий віртуальний прилад для роботи з даними вимірювання напруги.

Ключові слова: LABVIEW, ВІЗУАЛЬНЕ ПРОГРАМУВАННЯ, ПРОГРАМНИЙ МОДУЛЬ, ВИМІРЮВАННЯ НАПРУГИ, ВІРТУАЛЬНИЙ ПРИЛАД, ЛИЦЬОВА ПАНЕЛЬ, БЛОК-ДІАГРАМА, МАСИВ ДАНИХ, ГРАФІК НАПРУГИ, КОНТРОЛЬ МЕЖ, CSV.

ANNOTATION

Explanatory note to the qualification thesis: 52 pages, 13 figures, 4 appendices, 35 references.

The purpose of the qualification thesis is to develop a software module in a visual programming environment for processing voltage measurement data, which provides input or loading of voltage values, verification of their correctness, calculation of the main parameters, graphical representation of results, control of permissible limits, and saving data to a file.

Object of research: the process of processing voltage measurement data in a computerized measurement system.

Subject of research: a visual programming system for input, verification, calculation, graphical representation, and saving of data.

Research methods: analysis and processing of voltage measurement data, visual programming methods, structural design, algorithmic decomposition, modeling, and functional testing.

The result of the work is a developed software module in the LabVIEW environment for processing voltage measurement data. The work presents the general structure of the module, develops algorithms for operation and input data verification, defines the information support, creates a user interface in the form of a front panel, and tests the main operating modes. The software module provides calculation of the minimum, maximum, and average voltage values, control of values exceeding the permissible limits, construction of a voltage change graph, and saving of results.

The practical significance of the work is that the developed module can be used as a basic virtual instrument for working with voltage measurement data.

Keywords: LABVIEW, VISUAL PROGRAMMING, SOFTWARE MODULE, VOLTAGE MEASUREMENT, VIRTUAL INSTRUMENT, FRONT PANEL, BLOCK DIAGRAM, DATA ARRAY, VOLTAGE GRAPH, LIMIT CONTROL, CSV.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області та середовищ візуального програмування для опрацювання даних вимірювання напруги.....	10
1.1 Особливості отримання та опрацювання даних вимірювання напруги	10
1.2 Огляд середовищ візуального програмування для роботи з вимірювальними даними	12
1.3 Постановка задачі.....	14
2 Алгоритмічне та інформаційне забезпечення програмного модуля.....	17
2.1 Загальна структура програмного модуля	17
2.2 Алгоритми опрацювання даних вимірювання напруги	19
2.3 Інформаційне забезпечення модуля	23
3 Програмно-технологічне забезпечення програмного модуля	28
3.1 Вибір середовища візуального програмування та засобів реалізації	28
3.2 Інтерфейс користувача програмного модуля	30
3.3 Тестування розробленого програмного модуля	35
Висновки	39
Список використаних джерел	40
Додаток А Блок - діаграма вимірювального модуля	45
Додаток Б Алгоритм функціонування модуля	46
Додаток В Алгоритм опрацюванням масиву значень напруги	47
Додаток Д Апробація отриманих результатів	48

ВСТУП

В сучасних вимірювальних системах [1] дедалі рідше обмежуються ручним зчитуванням показників. Натомість дані одразу збирають, опрацьовують, відображають на графіках і зберігають для подальшого аналізу [14, 26]. Такий підхід особливо потрібний під час роботи з електричними величинами, зокрема з напругою. Напруга є одним із базових параметрів електронних та електротехнічних систем, оскільки за її значенням можна оцінювати стан джерел живлення, електронних схем, сенсорних вузлів, вимірювальних каналів і різних технічних об'єктів. В багатьох випадках потрібно не тільки зафіксувати поточне значення напруги, а і побачити, як воно змінюється в часі, визначити мінімум, максимум і середнє значення, а також перевірити, чи не виходить сигнал за допустимі межі.

Традиційне вимірювання напруги за допомогою окремого вольтметра є зручним для разового контролю, однак воно не завжди забезпечує достатні можливості для аналізу серій вимірювань. Якщо значення напруги змінюються в часі [2], користувачу потрібно мати засіб, який дозволяє накопичувати виміряні дані, подавати їх у вигляді графіка [3], виконувати базові обчислення та автоматично визначати відхилення [5]. Тому для таких задач зручніше використовувати комп'ютеризовані вимірювальні системи, які працюють не з одним значенням, а з цілою послідовністю вимірювань.

Для створення таких систем зручно застосовувати середовища візуального програмування. Їхня перевага полягає в тому, що робота програми подається через графічні блоки. Для вимірювальних задач це зручно, бо структура програми нагадує сам процес вимірювання. Одним із найпоширеніших середовищ такого типу є LabVIEW [17], яке орієнтоване на створення віртуальних приладів, роботу з вимірювальними даними, графічне подання сигналів [19], взаємодію з користувачем і збереження результатів [24].

Актуальність теми зумовлена потребою в простому програмному модулі, який бере на себе основні дії з опрацювання даних напруги: перевірку, розрахунки, побудову графіка та збереження результатів. Такий модуль дозволяє зменшити

кількість ручних дій, підвищити зручність аналізу результатів, швидко виявляти відхилення від допустимого діапазону та зберігати отримані дані у файлі. Для реалізації модуля обрано LabVIEW, оскільки в цьому середовищі вже є потрібні інструменти для роботи з масивами, графіками, елементами інтерфейсу та файлами результатів.

Метою кваліфікаційної роботи є розробка програмного модуля у середовищі візуального програмування для опрацювання даних вимірювання напруги

Для досягнення поставленої мети необхідно виконати такі завдання:

- провести аналіз особливостей отримання та опрацювання даних вимірювання напруги;
- розглянути середовища візуального програмування для роботи з вимірювальними даними;
- обґрунтувати доцільність використання LabVIEW для реалізації програмного модуля;
- сформулювати постановку задачі розробки програмного модуля;
- розробити загальну структуру програмного модуля;
- описати алгоритми опрацювання даних вимірювання напруги;
- сформулювати інформаційне забезпечення програмного модуля;
- обрати засоби реалізації програмного модуля;
- розробити інтерфейс користувача програмного модуля;
- провести тестування розробленого програмного модуля.

Об'єктом дослідження є процес опрацювання даних вимірювання напруги у комп'ютеризованій вимірювальній системі.

Предметом дослідження є програмний модуль у середовищі LabVIEW, призначений для введення, перевірки, обчислення, графічного відображення та збереження даних вимірювання напруги.

Результати кваліфікаційної роботи апробовані та опубліковані у матеріалах XI Міжнародної наукової конференції «Традиційні та інноваційні підходи до наукових досліджень» (м. Луцьк, Україна).

Методи дослідження включають структурне проектування програмного модуля, алгоритмічну декомпозицію процесу опрацювання даних, моделювання роботи модуля на тестових наборах значень напруги та функціональне тестування розробленого інтерфейсу.

Практичне значення одержаних результатів полягає у створенні програмного модуля, який може бути використаний як основа для віртуального вимірювального приладу в середовищі LabVIEW. Розроблений модуль забезпечує введення або завантаження значень напруги, перевірку їх коректності, обчислення основних параметрів, побудову графіка зміни напруги, контроль допустимих меж і збереження результатів.

Кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА СЕРЕДОВИЩ ВІЗУАЛЬНОГО ПРОГРАМУВАННЯ ДЛЯ ОПРАЦЮВАННЯ ДАНИХ ВИМІРЮВАННЯ НАПРУГИ

1.1 Особливості отримання та опрацювання даних вимірювання напруги

Вимірювання напруги є одним із найпоширеніших завдань в електроніці, електротехніці, автоматизованих системах контролю та лабораторних дослідженнях. Напруга характеризує різницю електричних потенціалів між двома точками кола, тому її значення дає змогу оцінювати стан електронного пристрою, якість живлення, рівень сигналу, роботу сенсорного вузла або поведінку досліджуваної схеми. В сучасних вимірювальних системах важливо не лише отримати числове значення напруги, а й забезпечити його подальше опрацювання: перевірку коректності, фільтрацію, збереження, побудову графіків та формування узагальнених показників.

Значення напруги можна отримувати кількома способами. Найпростіший варіант — використати цифровий вольтметр, який показує значення напруги на екрані. Але цього недостатньо, коли потрібно простежити зміну напруги в часі, порівняти кілька серій вимірювань або автоматично знайти значення поза допустимими межами. В таких випадках краще використовувати комп'ютеризовану вимірювальну систему [1, 11], де аналоговий сигнал напруги спочатку перетворюється у цифрові відліки [14, 26].

Напруга як аналоговий сигнал може змінюватися безперервно, тому для комп'ютерного аналізу її потрібно подати у вигляді окремих відліків. Для подальшого опрацювання в комп'ютерній системі такий сигнал дискретизують, тобто отримують окремі відліки у визначені моменти часу. На якість таких даних впливають розрядність АЦП [8, 11], частота дискретизації, вибраний діапазон вимірювання, схема підключення та рівень електричних завад [26].

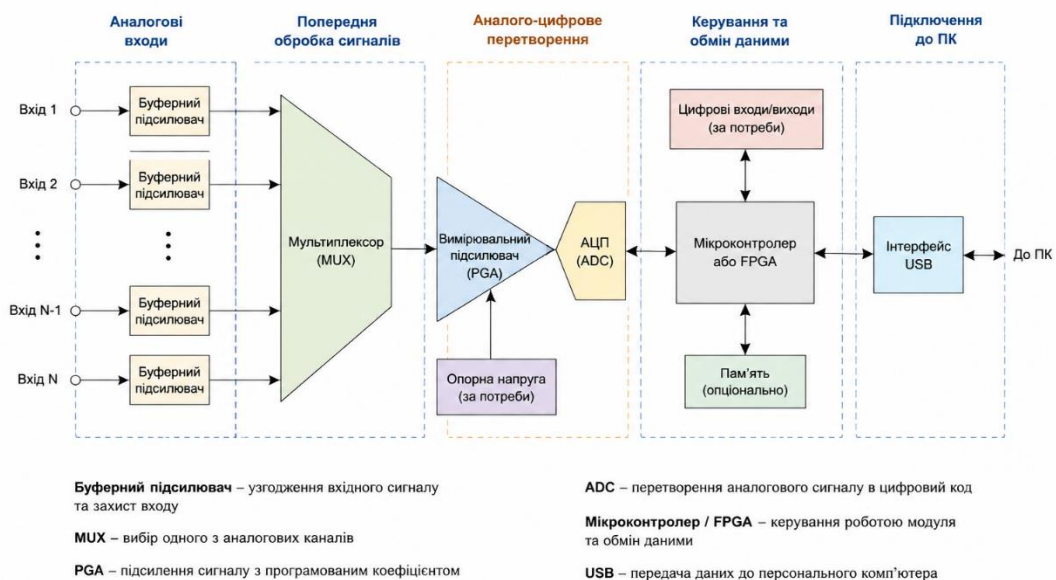


Рисунок 1.1 – Структура каналу збирання даних напруги

Якщо частота дискретизації є недостатньою, частина інформації про зміну сигналу може бути втрачена. Якщо ж діапазон вимірювання вибрано неправильно, можливе перевантаження входу або зниження точності результатів.

В системі збирання даних сигнал напруги проходить кілька послідовних етапів.

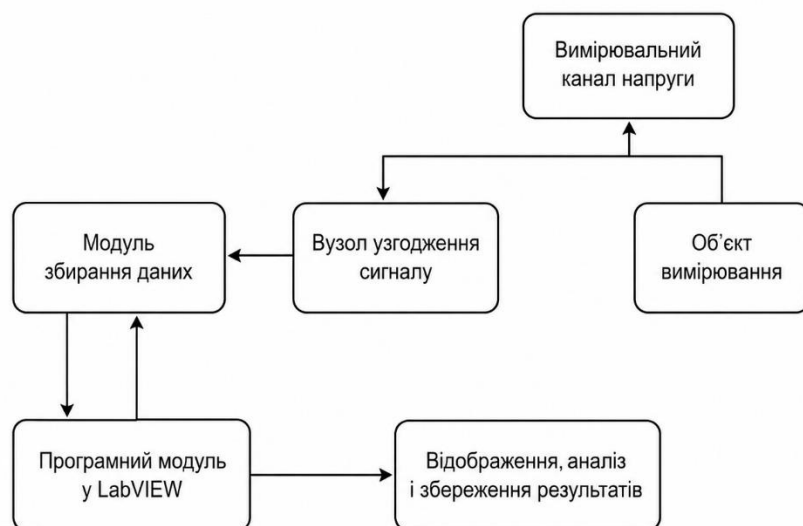


Рисунок 1.2 – Узагальнена структура отримання та опрацювання даних вимірювання напруги

Спочатку сигнал надходить на вхід вимірювального модуля, де може виконуватися його масштабування, підсилення або захист від перенапруги. Далі аналоговий сигнал подається на аналого-цифровий перетворювач [11], після чого формується масив цифрових значень [26]. Ці значення можуть передаватися до персонального комп'ютера через USB [10, 21], Ethernet або інший інтерфейс [22, 23]. На наступному етапі програмний модуль виконує опрацювання даних: обчислення мінімального, максимального та середнього значення напруги, визначення відхилень, побудову часової діаграми, збереження результатів у файл або підготовку звіту.

Окремо потрібно перевіряти, наскільки отримані дані придатні для подальшого аналізу. Під час вимірювання [8, 9] можуть виникати похибки, пов'язані з нестабільністю джерела сигналу, шумами, неточністю вимірювального пристрою, неправильним підключенням або впливом зовнішніх умов [26]. Через це в програмі варто передбачити перевірку пропусків, різких стрибків, некоректних значень і виходу напруги за встановлений діапазон. Для зменшення впливу випадкових коливань можуть використовуватися усереднення, ковзне середнє або інші прості методи згладжування.

Для таких задач добре підходить LabVIEW [4, 5], оскільки воно створене саме для побудови віртуальних приладів [17], роботи з вимірювальними сигналами та швидкого відображення результатів на графіках [19].

1.2 Огляд середовищ візуального програмування для роботи з вимірювальними даними

Для роботи з вимірювальними даними можна використовувати різні програмні середовища. Вони відрізняються способом побудови програми, можливостями візуалізації та підтримкою вимірювального обладнання. Серед таких засобів варто виділити середовища візуального програмування, де логіка роботи задається не текстовим кодом, а графічними блоками та зв'язками між ними. Такий підхід є зручним для створення вимірювальних програм [17], оскільки

структура програми наближається до логіки реального вимірювального процесу [19, 24]: отримання сигналу, його перетворення, обчислення параметрів, побудова графіка та збереження результатів.

Одним із найпоширеніших середовищ для створення віртуальних вимірювальних приладів є LabVIEW [16]. Це середовище орієнтоване на роботу з вимірювальним обладнанням, платами збирання даних, датчиками, лабораторними стендами та промисловими системами контролю. У LabVIEW програма створюється як віртуальний прилад, що має дві основні частини: лицьову панель і блок-діаграму [33].

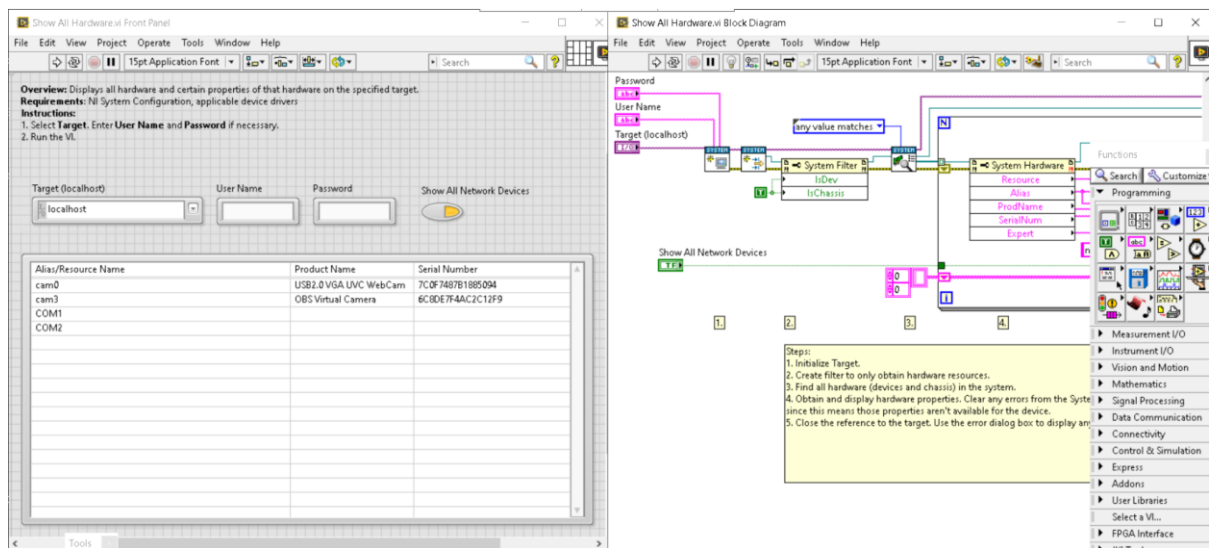


Рисунок 1.3 – Компоненти віртуального приладу: передня панель і блок-діаграма

Передня панель [17, 19] використовується як інтерфейс користувача. На ній розміщуються кнопки, перемикачі, числові поля, індикатори, графіки та інші елементи взаємодії. Блок-діаграма [17] відповідає за алгоритмічну частину програми [18], тобто визначає послідовність обробки даних, зв'язки між функціональними блоками та порядок виконання операцій [19].

LabVIEW [19] зручне тим, що в ньому вже є інструменти для графіків, масивів, запису у файл, простих розрахунків і взаємодії з вимірювальними пристроями [31]. Це дає змогу створювати програмний модуль, який не тільки приймає значення напруги, але й одразу відображає їх у зручному вигляді, виконує

первинне опрацювання, визначає мінімальні, максимальні та середні значення, а також дає можливість зберегти результати для подальшого аналізу.

Для роботи з вимірювальними даними також можуть застосовуватися інші програмні середовища. Наприклад, MATLAB/Simulink [13] є зручним для математичного моделювання, побудови алгоритмів цифрової обробки сигналів та аналізу великих масивів даних [27]. Але для створення інтерактивного вимірювального інтерфейсу та безпосередньої взаємодії з обладнанням він часто потребує додаткових модулів і налаштувань. Multisim [25] використовується переважно для моделювання електронних схем і віртуальних вимірювань у навчальному процесі, але він більше орієнтований на імітацію роботи схем, ніж на створення повноцінного програмного модуля збору й опрацювання реальних вимірювальних даних. Node-RED [28] також можна розглядати як візуальне середовище [29], яке зручно застосовувати для потокового обміну даними, роботи з мережевими пристроями та простими системами моніторингу, однак для лабораторних вимірювань напруги воно менш спеціалізоване, ніж LabVIEW.

Після порівняння середовищ основним засобом реалізації обрано LabVIEW. Його перевагами є графічний принцип побудови програми, наявність зручної передньої панелі, підтримка вимірювальних пристроїв, засоби візуалізації сигналів, робота з файлами та можливість швидкої побудови віртуального приладу. Саме тому в межах цієї кваліфікаційної роботи LabVIEW доцільно використати як основне середовище для створення програмного модуля, що забезпечує введення або отримання даних напруги, їх опрацювання, графічне подання та збереження результатів.

1.3 Постановка задачі

Проведений аналіз показав, що для цієї теми доцільно розробити окремий модуль у LabVIEW, який буде працювати з даними вимірювання напруги. Це середовище підходить для роботи, оскільки поєднує графічний інтерфейс, блокову логіку програми, візуалізацію сигналів і засоби збереження результатів.

Основна проблема полягає в тому, що окреме значення напруги мало що показує без контексту: потрібно бачити послідовність вимірювань і зміну сигналу в часі. Для практичного використання їх потрібно впорядкувати, перевірити, опрацювати, подати у вигляді графіка та зберегти для подальшого аналізу. Якщо ці дії виконувати вручну, зростає ймовірність помилок, збільшується час обробки результатів і ускладнюється аналіз зміни напруги в часі. Тому актуальним є створення програмного модуля, який автоматизує основні етапи роботи з вимірювальними даними.

В межах роботи потрібно створити модуль LabVIEW [4, 5], який приймає значення напруги та виконує їх базове опрацювання [6, 24]. Модуль має забезпечувати введення або завантаження значень напруги, їх первинну перевірку, обчислення основних параметрів, графічне відображення зміни сигналу та збереження результатів. Для роботи з модулем потрібна зрозуміла лицьова панель, через яку користувач зможе запускати опрацювання, задавати межі напруги та переглядати результати.

Вхідними даними для програмного модуля є послідовність значень напруги. Такі дані можуть бути отримані з вимірювального каналу, завантажені з файлу або сформовані у вигляді тестового масиву для перевірки працездатності програми. На етапі розробки достатньо використати тестові дані. Це дає змогу перевірити алгоритм без підключення реального вимірювального пристрою.

В процесі опрацювання даних програмний модуль повинен виконувати кілька основних функцій:

- забезпечити приймання або введення масиву значень напруги;
- виконати перевірку коректності даних, зокрема виявлення порожніх значень, некоректних числових даних або значень, що виходять за межі допустимого діапазону;
- модуль має обчислювати мінімальне, максимальне та середнє значення напруги;
- побудову графіка зміни напруги, що дозволить користувачу візуально оцінити характер сигналу;

– збереження результатів опрацювання у файл для подальшого використання.

Лицьова панель програмного модуля повинна містити елементи керування запуском і зупинкою опрацювання, поля для відображення числових результатів, графік зміни напруги, індикатори стану та елементи для задання допустимих меж напруги. Це дозволить зробити модуль зрозумілим для користувача та наблизити його до принципу роботи віртуального вимірювального приладу.

Результатом роботи має бути працездатний модуль LabVIEW для введення, перевірки, аналізу та збереження даних напруги. Модуль повинен показати, що засобами візуального програмування можна швидко створити зрозумілий інструмент для роботи з вимірювальними даними. Надалі такий модуль може бути розширений [21, 22] шляхом підключення реального вимірювального обладнання, додавання фільтрації сигналу, формування звітів або реалізації розширених методів аналізу [23, 32].

2 АЛГОРИТМІЧНЕ ТА ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ПРОГРАМНОГО МОДУЛЯ

2.1 Загальна структура програмного модуля

Оскільки розробка виконується в LabVIEW [17, 19], модуль подано як віртуальний прилад з лицьовою панеллю та блок-діаграмою. Лицьова панель виконує роль інтерфейсу користувача, а блок-діаграма реалізує алгоритмічну частину програми. Таке розділення є зручним: користувач працює з лицьовою панеллю, а вся логіка обчислень виконується на блок-діаграмі.

Модуль орієнтований на роботу з послідовністю значень напруги. Тому в структурі модуля передбачено всі основні дії: введення даних, перевірку, розрахунок показників, побудову графіка, контроль меж і запис результатів.

Структура модуля складається з кількох функціональних блоків: введення даних, налаштування параметрів, перевірки, обчислень, контролю меж, візуалізації, збереження та керування.

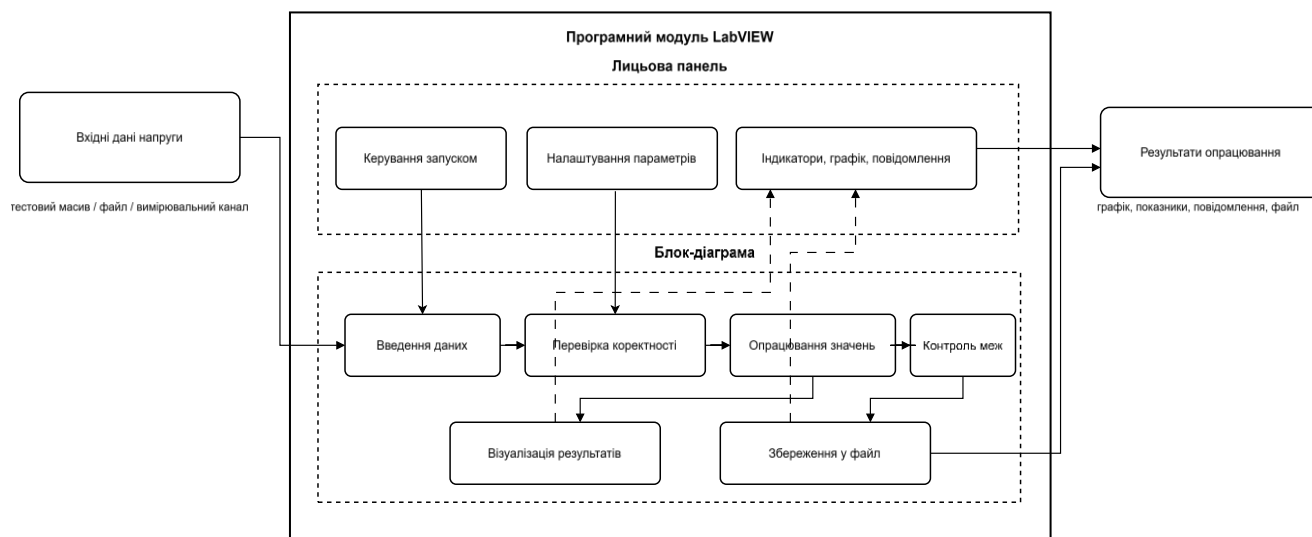


Рисунок 2.1 – Загальна структура програмного модуля опрацювання даних вимірювання напруги

Блок введення даних призначений для отримання значень напруги, які надалі передаються на опрацювання.

Ці дані можуть формуватися як тестовий масив [12], вводитися користувачем або завантажуватися з файлу [13]. Надалі структура модуля може бути розширена для підключення різних вимірювальних пристроїв [23] або модуля збирання даних [32]. Це зручно на етапі розробки, тому що спочатку можна перевірити роботу програми на тестових даних, а вже потім підключати реальний вимірювальний пристрій.

Блок налаштування параметрів забезпечує введення користувачем основних умов опрацювання. До таких параметрів можна віднести допустимі мінімальне та максимальне значення напруги, кількість відліків, інтервал між вимірюваннями, шлях для збереження файлу результатів або режим роботи програми. Завдяки цьому блоку користувач може змінювати межі напруги, кількість відліків або режим роботи без редагування блок-діаграми.

Блок перевірки коректності даних виконує первинний контроль вхідної інформації. Він перевіряє, чи є у вхідному наборі пропуски, порожні поля або значення, які не можна використати в розрахунках. Якщо виявлено помилку, користувач повинен отримати відповідне повідомлення або індикатор стану. Така перевірка зменшує ризик отримати неправильні результати через помилки у вхідних даних.

Основна обробка виконується в блоці опрацювання вимірних значень. В ньому виконуються основні обчислення над масивом значень напруги. До базових показників належать мінімальне значення, максимальне значення, середнє значення та, за потреби, кількість відліків, що виходять за допустимі межі. Отримані результати передаються на елементи індикації лицьової панелі та можуть бути записані до файлу.

Блок контролю допустимих меж призначений для порівняння кожного значення напруги із заданими користувачем межами. Якщо значення виходить за встановлений діапазон, програма повинна зафіксувати це як відхилення. Цей блок потрібний не лише для відображення значень, а й для перевірки, чи відповідає сигнал заданому діапазону.

Блок візуалізації забезпечує графічне подання результатів. Основним елементом візуалізації є графік, на якому видно зміну напруги за відліками або в часі. Крім графіка, на лицьовій панелі можуть бути розміщені числові індикатори мінімального, максимального та середнього значення напруги, індикатор стану програми, повідомлення про помилки та індикатор виходу значень за допустимі межі.

Блок збереження результатів відповідає за запис опрацьованих даних у файл. До файлу можуть бути внесені початкові значення напруги, результати обчислень, інформація про допустимі межі та виявлені відхилення. Збереження результатів потрібне для того, щоб пізніше порівняти кілька вимірювань або використати дані у звіті.

Блок керування роботою програми забезпечує запуск, зупинку та повторне виконання опрацювання. На лицьовій панелі для цього доцільно передбачити кнопки запуску, зупинки, очищення попередніх результатів і збереження даних. У блок-діаграмі ці елементи мають бути пов'язані з відповідними структурами керування виконанням програми.

Структура модуля побудована так, щоб дані послідовно проходили шлях від введення до графіка, числових результатів і файлу. У спрощеному вигляді цей процес можна подати так: введення або завантаження значень напруги, налаштування параметрів, перевірка даних, обчислення основних показників, контроль допустимих меж, побудова графіка та збереження результатів. Така структура є достатньою для реалізації базового програмного модуля і водночас залишає можливість для подальшого розширення, наприклад додавання фільтрації сигналу, підключення реального вимірювального обладнання або автоматичного формування звіту.

2.2 Алгоритми опрацювання даних вимірювання напруги

Оскільки модуль реалізується в LabVIEW, його алгоритм описано як послідовність дій, які потім відтворюються на блок-діаграмі. Такий підхід дозволяє

пов'язати елементи лицьової панелі з обчислювальними блоками та забезпечити зрозумілий порядок проходження даних через програму:

1. Початок роботи програмного модуля.
2. Ініціалізація елементів лицьової панелі LabVIEW.
3. Вибір режиму отримання даних:
 - 3.1. Якщо обрано режим тестового масиву, то сформувати набір тестових значень напруги.
 - 3.2. Якщо обрано режим ручного введення, то зчитати значення, введені користувачем.
 - 3.3. Якщо обрано режим завантаження з файлу, то відкрити файл і зчитати масив значень напруги.
4. Зчитати параметри опрацювання: нижню межу напруги, верхню межу напруги, кількість відліків, інтервал вимірювання та шлях для збереження результатів.
5. Передати отриманий масив значень напруги до блоку перевірки коректності даних.
6. Виконати перевірку коректності вхідних даних.
7. Якщо дані некоректні, то:
 - 7.1. Вивести повідомлення про помилку на лицьову панель.
 - 7.2. Заблокувати подальше опрацювання.
 - 7.3. Перейти до пункту 14.
8. Якщо дані коректні, то передати масив значень до блоку опрацювання.
9. Обчислити мінімальне, максимальне та середнє значення напруги.
10. Виконати контроль допустимих меж напруги.
11. Відобразити результати на лицьовій панелі:
 - 11.1. Побудувати графік зміни напруги.
 - 11.2. Вивести мінімальне, максимальне та середнє значення.
 - 11.3. Вивести кількість значень, що вийшли за допустимі межі.
12. Перевірити, чи потрібно зберегти результати у файл.

13. Якщо збереження увімкнено, то записати результати опрацювання у файл.

14. Очікувати подальших дій користувача: повторити опрацювання, очистити результати або завершити роботу.

15. Кінець роботи програмного модуля.

Робота модуля починається з підготовки параметрів і елементів інтерфейсу. На цьому етапі користувач задає режим отримання даних, допустимі межі напруги, кількість відліків, інтервал між вимірюваннями та шлях для збереження результатів. Після цього виконується запуск процедури введення або завантаження даних. Значення напруги можуть бути отримані з тестового масиву, введені користувачем вручну або завантажені з файлу [13]. Це дає змогу перевірити роботу алгоритму без обов'язкового підключення реального вимірювального пристрою.

Після отримання вхідних даних виконується їх первинна перевірка. На цьому етапі програма перевіряє, чи є у масиві дані, з якими можна працювати далі. Якщо масив порожній або містить некоректні значення, програма повинна сформулювати повідомлення про помилку і не переходити до етапу обчислень. Також потрібно перевіряти, чи належать отримані значення до числового типу [26], чи не містять пропусків і чи відповідають очікуваному діапазону вимірювання. Без такої перевірки програма може виконати розрахунки над неповними або неправильними даними, що спотворить результат.

Основний етап алгоритму пов'язаний з опрацюванням масиву значень напруги:

1. Отримати масив значень напруги.
2. Перевірити, чи масив не є порожнім.
3. Якщо масив порожній, то:
 - 3.1. Сформулювати повідомлення «Дані відсутні».
 - 3.2. Повернути ознаку помилки.
 - 3.3. Завершити перевірку.
4. Перевірити кожне значення масиву на належність до числового типу.
5. Якщо виявлено нечислове або пропущене значення, то:

- 5.1. Сформулювати повідомлення «Некоректне значення у вхідних даних».
- 5.2. Повернути ознаку помилки.
- 5.3. Завершити перевірку.
- 6. Перевірити, чи нижня допустима межа не перевищує верхню допустиму межу.
- 7. Якщо нижня межа більша за верхню, то:
 - 7.1. Сформулювати повідомлення «Некоректно задано межі напруги».
 - 7.2. Повернути ознаку помилки.
 - 7.3. Завершити перевірку.
- 8. Якщо всі перевірки пройдено успішно, то:
 - 8.1. Передати масив значень на подальше опрацювання.
 - 8.2. Повернути ознаку успішної перевірки.
- 9. Завершити алгоритм перевірки.

Для кожного набору даних програмний модуль [13, 24] повинен визначати мінімальне, максимальне та середнє значення напруги. Мінімальне значення дозволяє встановити найнижчий рівень сигналу в межах отриманого набору даних. Максимальне значення показує найбільший зафіксований рівень напруги. Середнє значення використовується для загальної оцінки рівня сигналу за весь період вимірювання. Для базового аналізу цього достатньо: користувач бачить нижню, верхню та середню оцінку рівня напруги.

Окремою частиною алгоритму є контроль допустимих меж напруги. Користувач задає нижню та верхню межу, після чого кожне значення вхідного масиву порівнюється з цими параметрами. Якщо значення менше за нижню межу або більше за верхню межу, воно позначається як відхилення. Програма повинна підраховувати кількість таких значень і передавати інформацію про відхилення на лицьову панель. За кількістю відхилень користувач одразу бачить, чи залишався сигнал у заданому діапазоні.

Після виконання обчислень алгоритм переходить до візуалізації результатів. Основним способом подання даних є графік зміни напруги [19, 31]. На ньому користувач може побачити характер зміни сигналу, наявність різких стрибків,

спадів або виходу значень за допустимий діапазон. Крім графіка, на лицьовій панелі відображаються числові результати: мінімальне, максимальне та середнє значення, а також кількість виявлених відхилень. За потреби можуть використовуватися індикатори стану, які повідомляють про нормальний режим роботи або про наявність помилки.

Останній етап роботи алгоритму — запис результатів у файл [24]. Якщо користувач обрав режим запису у файл, програмний модуль формує набір вихідних даних, який включає початкові значення напруги, результати обчислення, допустимі межі та інформацію про виявлені відхилення. Збереження результатів дозволяє використовувати отримані дані для подальшого аналізу, порівняння декількох вимірювань або підготовки звітних матеріалів. Якщо запис у файл не потрібний, програма завершує цикл опрацювання після відображення результатів на лицьовій панелі.

Реалізований у LabVIEW алгоритм дає змогу отримати простий віртуальний прилад, де користувач задає параметри, запускає обробку й одразу бачить результат на панелі.

2.3 Інформаційне забезпечення модуля

Інформаційне забезпечення описує, які саме дані використовує модуль і як вони переходять між його блоками. У цій роботі це важливо, бо всі частини модуля пов'язані через дані: значення напруги надходять на блок-діаграму, обробляються, відображаються на панелі та за потреби записуються у файл.

Програмний модуль розглядається як віртуальний прилад, який працює з даними вимірювання напруги. Тому основними інформаційними об'єктами є значення напруги, параметри опрацювання, проміжні результати перевірки, результати обчислень, службові повідомлення та файл збереження результатів. Разом ці дані утворюють інформаційну модель модуля: від початкового масиву напруги до графіка, числових показників і файлу результатів.

Основним вхідним інформаційним об'єктом є масив значень напруги. В базовій реалізації ці значення можуть бути сформовані як тестовий масив, введені користувачем вручну або завантажені з файлу. На етапі розробки це практично, бо модуль можна перевірити навіть без реального вимірювального пристрою. Надалі цей самий інформаційний об'єкт може використовуватися для приймання даних із вимірювального каналу або модуля збирання даних. Тобто внутрішня логіка модуля не залежить від конкретного джерела даних, а працює з уніфікованим масивом числових значень.

Значення напруги подаються як одновимірний числовий масив [19]. Кожен елемент такого масиву відповідає одному відліку виміряної напруги. Наприклад, якщо було отримано десять вимірювань, то масив містить десять числових значень. Порядок розміщення елементів [18, 19] в масиві відповідає порядку їх отримання або завантаження. Такий порядок потрібний для графіка: точки мають відображатися саме в тій послідовності, у якій були отримані вимірювання.

Крім масиву значень напруги, до вхідної інформації належать параметри опрацювання. Вони задаються користувачем на лицьовій панелі програмного модуля та визначають умови подальшої роботи алгоритму. До таких параметрів належать нижня допустима межа напруги, верхня допустима межа напруги, кількість відліків, інтервал між вимірюваннями, режим отримання даних, а також шлях для збереження результатів. Нижня та верхня межі використовуються для контролю допустимого діапазону напруги. Кількість відліків дозволяє оцінити обсяг вхідного масиву. Інтервал між вимірюваннями може використовуватися для формування часової шкали графіка. Режим отримання даних визначає, звідки саме надходять значення: з тестового набору, файлу або іншого джерела. Шлях до файлу потрібний для збереження результатів після завершення опрацювання.

Проміжна інформація формується під час виконання алгоритмів програмного модуля. Вона не завжди безпосередньо відображається користувачу, але використовується блок-діаграмою LabVIEW для керування логікою виконання програми. До проміжних даних можна віднести ознаку коректності вхідного масиву, службовий стан виконання, поточне значення під час перегляду масиву,

лічильник оброблених відліків, лічильник виявлених відхилень, а також масив або список значень, які вийшли за допустимі межі. Ці проміжні дані допомагають програмі вирішити, що робити далі: продовжити обробку, зупинитися, показати помилку або перейти до збереження результатів.

Окрему роль відіграє інформація про коректність вхідних даних. Перед виконанням основних обчислень програмний модуль має перевірити, чи не є масив порожнім, чи всі значення є числовими, чи не містяться в ньому пропуски та чи правильно задано допустимі межі. Якщо нижня межа більша за верхню, подальший контроль діапазону втрачає зміст. Тому така ситуація повинна бути зафіксована як помилка. Результатом перевірки є службова ознака, яка може мати два стани: дані коректні або дані некоректні. В разі виявлення помилки програма повинна сформувати текстове повідомлення для користувача та припинити виконання основних розрахунків.

Після успішної перевірки даних формується група вихідних інформаційних об'єктів. До них належать мінімальне значення напруги, максимальне значення напруги, середнє значення напруги, кількість відхилень, стан сигналу та графік зміни напруги. Мінімальне значення показує найнижчий рівень напруги серед усіх отриманих відліків. Максимальне значення дозволяє визначити найбільший зафіксований рівень сигналу. Середнє значення використовується для узагальненої оцінки рівня напруги за весь період вимірювання. Кількість відхилень показує, скільки значень вийшло за встановлені межі. Стан сигналу може подаватися у вигляді текстового повідомлення, наприклад «Норма» або «Виявлено відхилення».

Одним із головних результатів роботи модуля є графік зміни напруги. Він дозволяє користувачу не тільки бачити окремі числові значення, але й оцінити характер зміни сигналу в часі або за номером відліку. Якщо використовується лише номер відліку, горизонтальна вісь графіка відображає послідовність вимірювань. Якщо задано інтервал між вимірюваннями, можна сформувати умовну часову шкалу. Вертикальна вісь графіка відображає значення напруги. На графіку користувач швидко помічає стрибки, просідання, стабільні ділянки та вихід сигналу за допустимі межі. Основні інформаційні об'єкти подано в таблиці 2.1.

Таблиця 2.1 – Основні інформаційні об’єкти програмного модуля

№	Інформаційний об’єкт	Тип даних	Призначення
1	Масив значень напруги	числовий масив	Зберігання вхідних вимірних значень
2	Номер відліку	ціле число або масив чисел	Формування послідовності точок для графіка
3	Час вимірювання	числове значення або масив	Побудова часової шкали за наявності інтервалу
4	Нижня допустима межа	числове значення	Контроль мінімального допустимого рівня напруги
5	Верхня допустима межа	числове значення	Контроль максимального допустимого рівня напруги
6	Кількість відліків	ціле число	Визначення обсягу вхідного масиву
7	Інтервал вимірювання	числове значення	Формування часової шкали сигналу
8	Режим отримання даних	текстове або перелічуване значення	Вибір джерела вхідних даних
9	Ознака коректності даних	логічне значення	Дозвіл або заборона подальшого опрацювання
10	Мінімальне значення напруги	числове значення	Відображення найменшого значення в наборі
11	Максимальне значення напруги	числове значення	Відображення найбільшого значення в наборі
12	Середнє значення напруги	числове значення	Узагальнена оцінка рівня сигналу
13	Кількість відхилень	ціле число	Фіксація кількості значень поза межами
14	Стан сигналу	текстове значення	Повідомлення про норму або відхилення
15	Шлях до файлу	текстове значення	Визначення місця збереження результатів
16	Файл результатів	табличні дані	Збереження значень і результатів опрацювання

В середовищі LabVIEW [18, 19] ці інформаційні об’єкти можуть бути реалізовані за допомогою стандартних елементів лицьової панелі та блок-діаграми. Масив напруги може передаватися між функціональними блоками як числовий масив. Нижня та верхня допустимі межі можуть задаватися через числові елементи керування. Режим отримання даних можна реалізувати за допомогою перемикача

або випадваючого списку. Результати обчислень можна подавати через числові індикатори. Графік зміни напруги доцільно відображати через графічний елемент, який показує послідовність значень у вигляді кривої. Службові повідомлення можуть виводитися через текстове поле або індикатор стану.

Файл результатів [24] є окремим елементом інформаційного забезпечення, оскільки він забезпечує збереження результатів [26] після завершення роботи програмного модуля. До файлу доцільно записувати не тільки початкові значення напруги, але й параметри, з якими виконувалося опрацювання. Це дозволяє в майбутньому правильно інтерпретувати отримані результати. Тому файл має містити значення напруги, номер або час відліку, задані межі, мінімальне, максимальне та середнє значення, кількість відхилень і загальний стан сигналу.

Для базової версії модуля найпростіше зберігати результати у табличному файлі. У такому файлі кожен рядок відповідає одному відліку, а в стовпцях можна подати номер, значення напруги, задані межі та ознаку відхилення. Окремо в кінці файлу або в службовій частині можна записати узагальнені показники: мінімальне, максимальне, середнє значення та кількість відхилень. Така структура є простою, зрозумілою і придатною для подальшого відкриття у табличних редакторах або повторного аналізу.

Інформаційні потоки модуля можна описати так: значення напруги та параметри користувача спочатку надходять до блоку перевірки, після чого формується перевірений масив даних. Далі цей масив передається до блоку обчислень і блоку контролю допустимих меж. Результати обчислень передаються на лицьову панель для відображення, а також до блоку збереження. Якщо користувач активував запис у файл, результати додатково перетворюються у табличну структуру та записуються у вибране місце. Якщо запис не потрібний, модуль завершує цикл після відображення результатів.

3 ПРОГРАМНО-ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ ПРОГРАМНОГО МОДУЛЯ

3.1 Вибір середовища візуального програмування та засобів реалізації

LabVIEW [19, 24] використовується для створення віртуальних приладів і автоматизації вимірювань, тому воно добре підходить для цієї роботи. На відміну від звичайного текстового програмування [18, 19], у LabVIEW логіка програми будується на блок-діаграмі за допомогою функціональних елементів і з'єднань між ними. Це зручно для задачі вимірювання напруги, бо весь процес легко розкласти на послідовні дії: отримати дані, перевірити їх, виконати розрахунки, побудувати графік і зберегти результат.

Важливою перевагою LabVIEW є наявність двох основних частин віртуального приладу: лицьової панелі та блок-діаграми. Лицьова панель (рисунок 3.1) використовується для взаємодії користувача з програмним модулем.

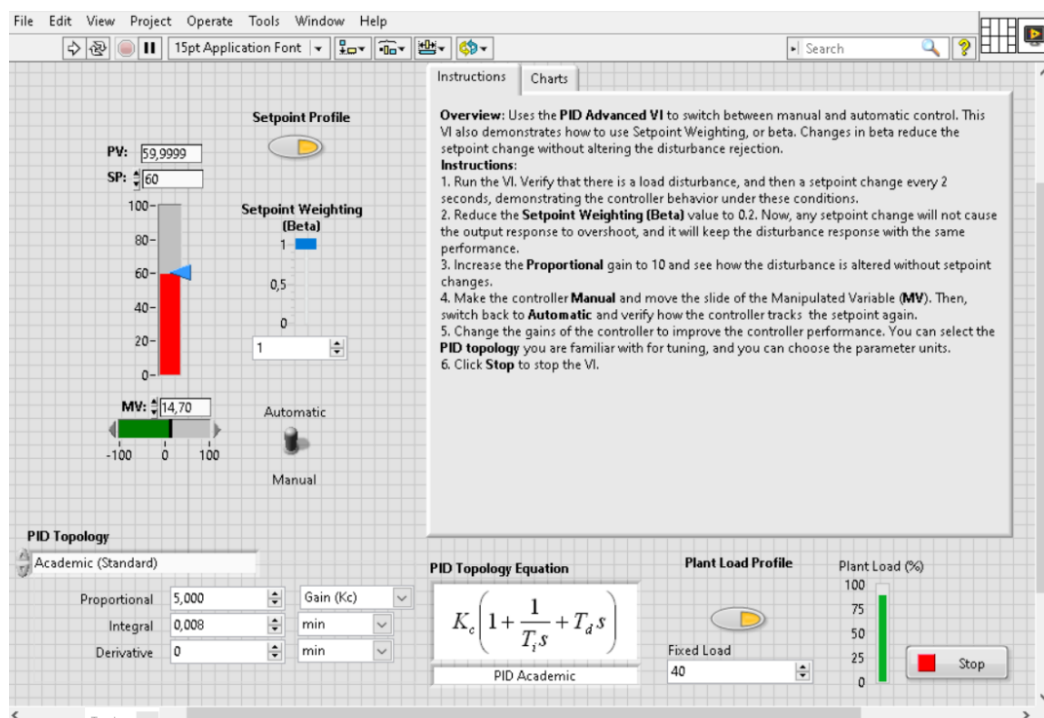


Рисунок 3.1 – Лицьова панель LabVIEW

На ній можна розмістити кнопки запуску й зупинки, числові поля для введення допустимих меж напруги, індикатори мінімального, максимального та

середнього значення, графік зміни напруги, повідомлення про стан сигналу та елементи для збереження результатів.

Блок-діаграма (рисунк 3.2), у свою чергу, реалізує внутрішню логіку роботи модуля: зчитування даних, перевірку масиву, виконання розрахунків, формування графіка та запис результатів у файл.

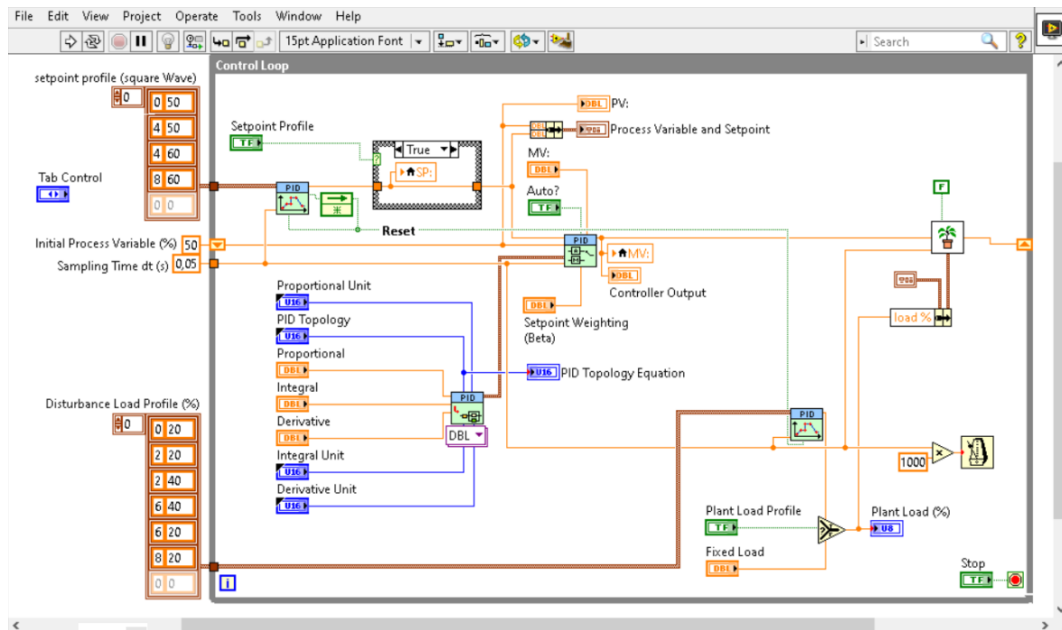


Рисунок 3.2 – Панель блоків LabVIEW

В LabVIEW вже є стандартні інструменти для роботи з числовими значеннями та масивами. Дані вимірювання напруги в межах програмного модуля подаються як одновимірний числовий масив, у якому кожен елемент відповідає окремому відліку напруги. В LabVIEW можна зручно виконувати операції над такими масивами [31]: визначати мінімальне та максимальне значення, обчислювати середнє значення, переглядати елементи масиву в циклі, порівнювати їх із заданими межами та формувати результати для подальшого відображення.

Ще одна причина вибору LabVIEW — зручна побудова графіків і індикаторів. Для задачі опрацювання напруги важливо не тільки отримати числові результати, але й побачити зміну сигналу у вигляді графіка. Це дозволяє швидше оцінити характер сигналу, помітити різкі зміни, стабільні ділянки або вихід значень за допустимі межі.

У цій роботі основна увага приділена програмній частині, а не створенню нового вимірювального пристрою. Основна увага приділяється програмній частині, тому вхідні дані можуть бути отримані кількома способами: сформовані як тестовий масив, введені користувачем або завантажені з файлу. Це практично, бо алгоритми можна перевіряти на тестових і файлових даних без постійного підключення апаратури.

Результати зручно зберігати у CSV-файл, оскільки його легко відкрити в табличному редакторі. Такий спосіб є простим і зручним, тому що дозволяє зберігати як початкові значення напруги, так і результати їх опрацювання: мінімальне, максимальне та середнє значення, кількість відхилень і стан сигналу. Надалі такий файл можна відкрити у табличному редакторі або використати для повторного аналізу. У LabVIEW для цього можна застосувати стандартні функціональні блоки [19] запису даних у файл.

Для розробки та перевірки модуля достатньо звичайного комп'ютера або ноутбука з установленим LabVIEW. Оскільки базова реалізація працює з тестовими або файловими даними, додаткове спеціалізоване обладнання на початковому етапі не є обов'язковим. Проте структура модуля [23, 32] повинна бути побудована так, щоб у майбутньому можна було додати блок отримання даних із вимірювального пристрою без повної переробки програми.

3.2 Інтерфейс користувача програмного модуля

Інтерфейс є основною частиною взаємодії з модулем: через нього користувач задає параметри, запускає обробку, переглядає графік і зберігає результати. Оскільки модуль створюється в LabVIEW, інтерфейс реалізовано на лицьовій панелі [17, 19] віртуального приладу. Лицьова панель має бути простою, зрозумілою та логічно поділеною на окремі функціональні зони.

Інтерфейс має зробити роботу з даними напруги простою: від вибору джерела даних до перегляду результатів. Користувач повинен мати можливість швидко вибрати джерело вхідних даних, задати допустимі межі напруги, запустити

опрацювання, переглянути графік зміни сигналу, оцінити основні числові показники та за потреби зберегти результати. Тому елементи інтерфейсу краще згрупувати: окремо параметри, окремо кнопки керування, окремо графік і результати.

Ліва частина панелі відведена для вибору джерела даних і введення значень напруги (рисунки 3.3). У цьому блоці користувач може вибрати режим отримання значень напруги.

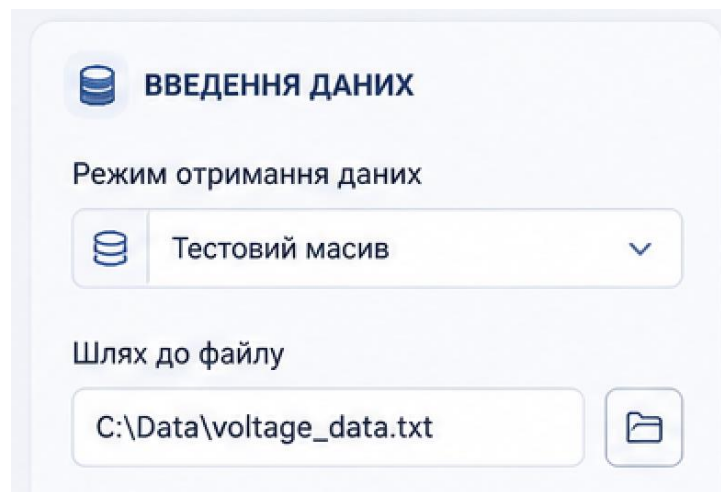


Рисунок 3.3 – Блок введення даних

Для базової реалізації доцільно передбачити кілька режимів: використання тестового масиву, ручне введення значень або завантаження даних із файлу. Такий підхід дозволяє перевіряти роботу програмного модуля без обов'язкового підключення реального вимірювального обладнання. Якщо використовується файл, на панелі має бути елемент для вибору шляху до файлу. Якщо використовується тестовий масив, програма може автоматично сформувати набір значень для демонстрації роботи алгоритму.

Другою частиною інтерфейсу є блок налаштування параметрів опрацювання (рисунки 3.4). У ньому розміщуються числові поля для введення нижньої та верхньої допустимої межі напруги.

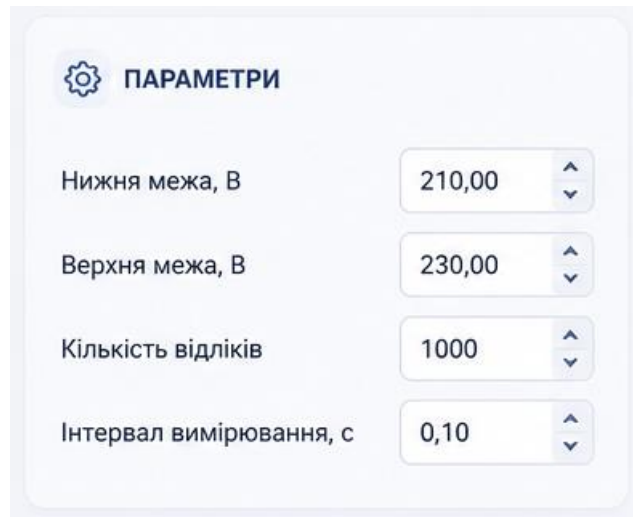


Рисунок 3.4 – Блок налаштування

Ці межі використовуються для автоматичного визначення значень, які виходять за встановлений діапазон. Також доцільно передбачити поле для введення кількості відліків або інтервалу між вимірюваннями. Інтервал між вимірюваннями може використовуватися для формування умовної часової шкали графіка. Усі ці параметри користувач повинен задавати до запуску опрацювання.

Третьою частиною інтерфейсу є блок керування роботою програмного модуля (рисунок 3.5).

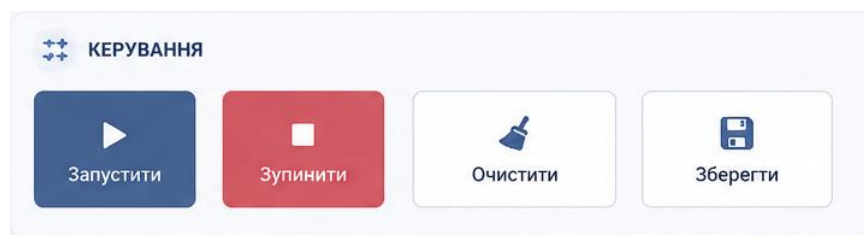


Рисунок 3.5 – Блок управління

До нього належать кнопки запуску, зупинки, очищення результатів і збереження даних. Кнопка запуску ініціює виконання алгоритму опрацювання. Кнопка зупинки дозволяє завершити роботу або перервати виконання процесу. Кнопка очищення потрібна для видалення попередніх результатів із графіка та числових індикаторів перед новим запуском. Кнопка збереження використовується

для запису результатів у файл. Завдяки такому набору кнопок користувач швидко розуміє, як запустити, зупинити, очистити або зберегти результати.

В центрі панелі розміщено графік зміни напруги (рисунок 3.6). Він відображає послідовність значень напруги за номером відліку або за умовною часовою шкалою.

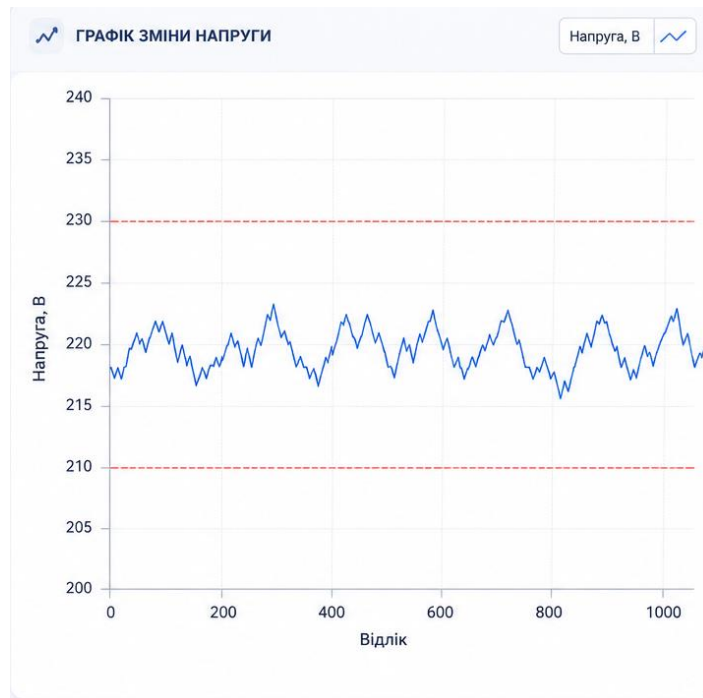


Рисунок 3.6 – Блок графік зміни напруги

За графіком видно, чи сигнал стабільний, чи має різкі стрибки, просідання або підвищення.

Окремий блок лицьової панелі повинен містити числові індикатори результатів опрацювання. До таких індикаторів належать мінімальне значення напруги, максимальне значення напруги, середнє значення та кількість відхилень від допустимого діапазону. Мінімальне значення показує найнижчий рівень сигналу в наборі даних, максимальне — найвищий рівень, а середнє — узагальнений рівень напруги. Кількість відхилень дозволяє оцінити, скільки значень не відповідають заданим межам.

Для зручності на панелі передбачено індикатор стану роботи програми. Він може відображати повідомлення про поточний режим роботи програми,

наприклад: «Очікування запуску», «Опрацювання виконується», «Дані коректні», «Виявлено відхилення», «Помилка вхідних даних» або «Результати збережено». Це допомагає користувачу швидко зрозуміти, що зараз відбувається: програма очікує запуску, обробляє дані, знайшла помилку або завершила роботу.

Якщо користувач задав некоректні межі напруги, не вибрав файл або вхідний масив не містить даних, програма повинна вивести зрозуміле повідомлення. Так користувач бачить не лише факт помилки, а й її причину.

Загалом інтерфейс користувача програмного модуля доцільно поділити на такі зони: зона введення даних, зона налаштування параметрів, зона керування, зона графічного відображення, зона числових результатів і зона службових повідомлень (рисунк 3.7).

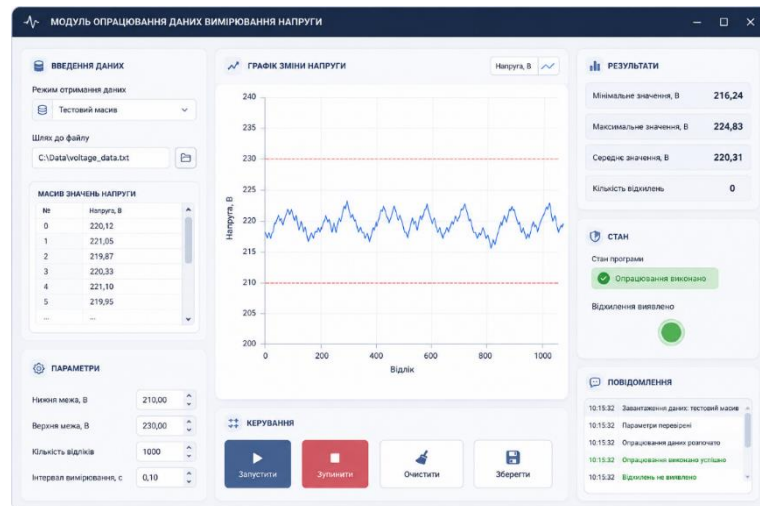


Рисунок 3.7 – Макет лицьової панелі програмного модуля

Таке компонування робить панель зрозумілою: користувач рухається від введення параметрів до запуску та перегляду результатів. Користувач спочатку задає джерело даних і параметри, потім запускає опрацювання, після чого переглядає графік, числові результати та повідомлення про стан.

Розроблений інтерфейс має забезпечити просту роботу з модулем без зайвих дій з боку користувача. Використання лицьової панелі LabVIEW дозволяє поєднати елементи керування [19, 31], індикатори, графік і службові повідомлення в одному вікні.

3.3 Тестування розробленого програмного модуля

Після створення модуля було перевірено його роботу в кількох сценаріях. Під час тестування перевіряли, чи правильно модуль приймає дані, виконує розрахунки, будує графік, контролює межі напруги та записує результати у файл. Програмний модуль призначений для опрацювання даних вимірювання напруги, тестування охоплювало як нормальні режими роботи, так і ситуації з помилковими або граничними даними.

На першому етапі було перевірено запуск програмного модуля та коректність відображення елементів лицьової панелі. Під час запуску користувач повинен бачити всі основні елементи інтерфейсу: блок вибору джерела даних, поля налаштування нижньої та верхньої межі напруги, поле кількості відліків, поле інтервалу вимірювання, кнопки керування, графік зміни напруги, числові індикатори результатів і повідомлення про стан програми. Така перевірка показує, чи користувач одразу бачить усі потрібні елементи для роботи з модулем.

На другому етапі було протестовано роботу модуля з тестовим масивом значень напруги (рисунк 3.8).

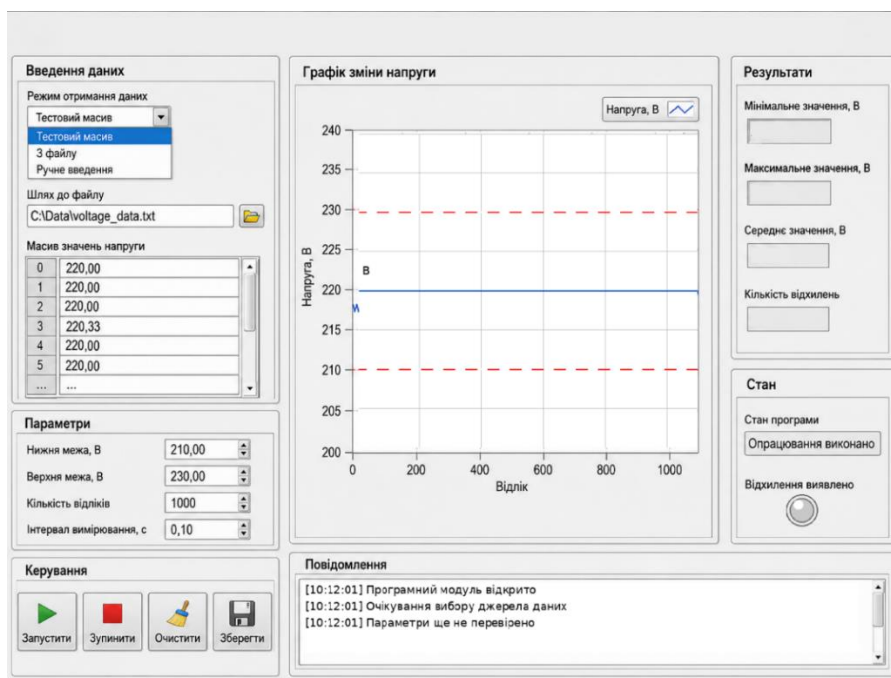


Рисунок 3.8 – Опрацювання тестового масиву значень напруги

Для перевірки використано набір числових значень, який імітує реальні вимірювання напруги. Після натискання кнопки запуску програма повинна прийняти масив, перевірити його коректність, обчислити мінімальне, максимальне та середнє значення, побудувати графік і вивести результати на лицьову панель. Цей режим дає змогу перевірити базову роботу модуля [12, 13] без підключення вимірювального пристрою.

Окремо було перевірено правильність обчислення основних параметрів напруги. Для цього до модуля подавалися контрольні набори даних із заздалегідь відомими мінімальними, максимальними та середніми значеннями. Наприклад, для масиву значень 218 В, 220 В, 222 В, 224 В програма повинна визначити мінімальне значення 218 В, максимальне значення 224 В, а середнє значення 221 В. Збіг із ручним розрахунком підтверджує, що блок обчислень працює правильно.

Наступним етапом було тестування контролю допустимих меж (рисунок 3.9).

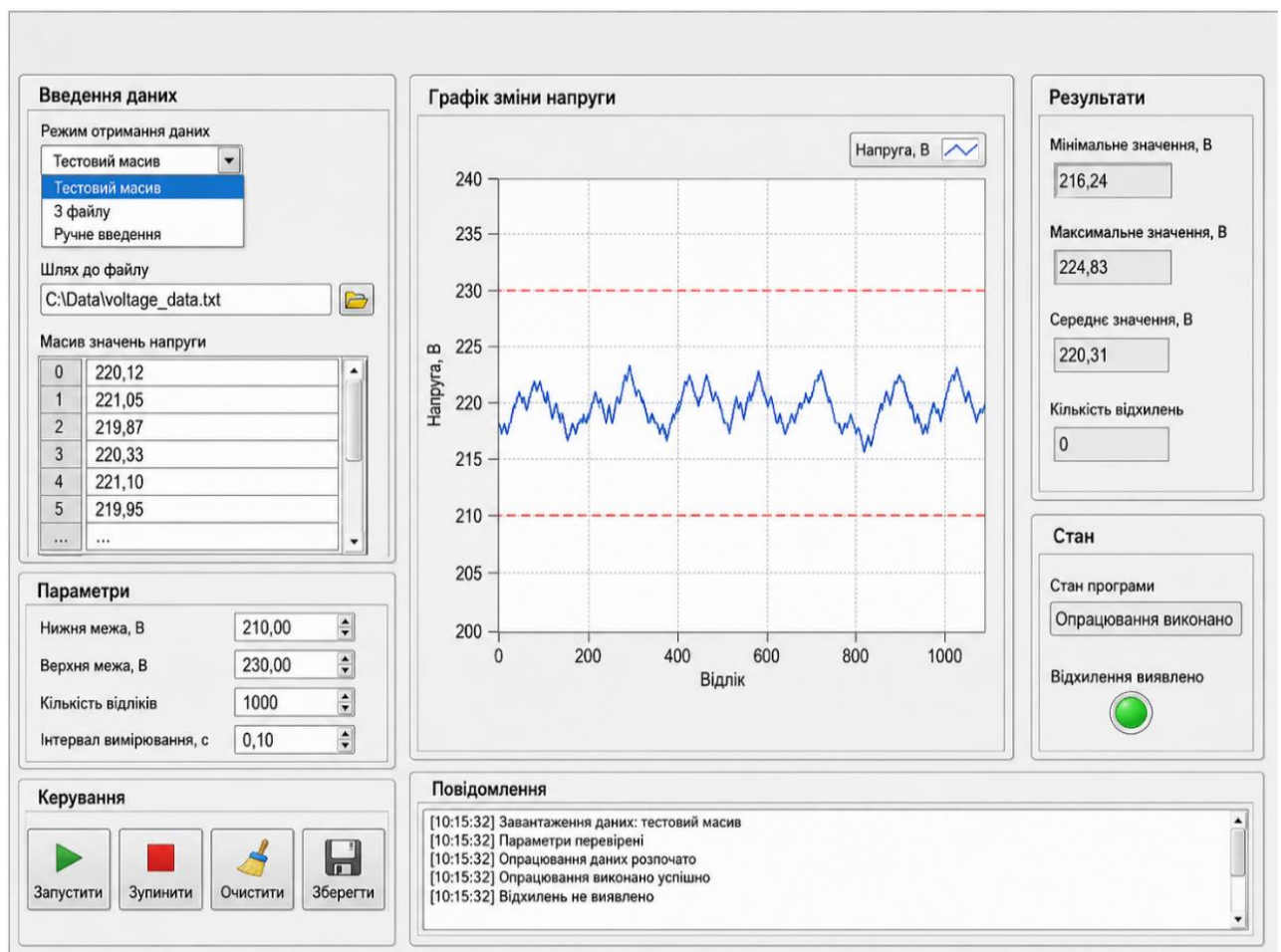


Рисунок 3.9 – Тестування контролю допустимих меж

Для цього задавалися нижня та верхня межі напруги, наприклад 210 В і 230 В. У першому тестовому випадку всі значення перебували в межах заданого діапазону, тому програма повинна була сформувати стан «Норма» і показати нульову кількість відхилень. У другому випадку до масиву були додані значення, що виходять за межі допустимого діапазону, наприклад 205 В або 235 В. У цьому випадку модуль фіксує відхилення, збільшує їх кількість і показує повідомлення на панелі.

Також було перевірено реакцію програмного модуля на некоректні вхідні дані (рисунок 3.10).

До таких ситуацій належать порожній масив значень, неправильно задані межі напруги, коли нижня межа більша за верхню, а також відсутність вибраного файлу для завантаження даних. У таких випадках програма зупиняє обробку і повідомляє користувача, що саме потрібно виправити. Це захищає користувача від неправильних результатів і спрощує пошук помилки.

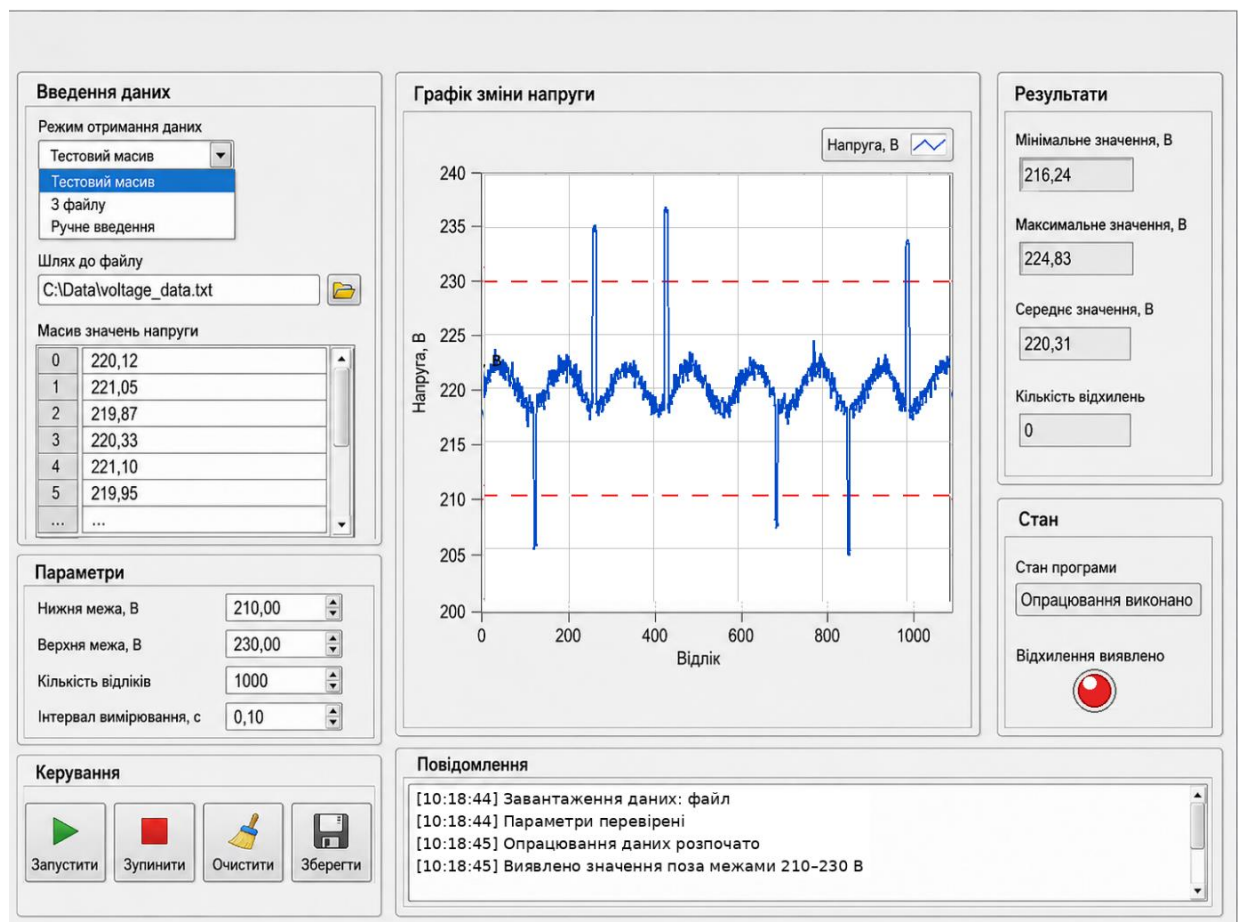


Рисунок 3.10 – Реакція програмного модуля на некоректні вхідні дані

На завершальному етапі було перевірено збереження результатів у файл. Після обробки користувач натискає кнопку збереження, і програма записує у файл вхідні значення та отримані результати.

Тестування підтвердило, що модуль виконує основні дії: приймає дані, перевіряє їх, обчислює параметри, контролює межі, будує графік і зберігає результати. Проведені перевірки показали, що модуль можна використовувати як базовий віртуальний прилад для роботи з даними напруги.

ВИСНОВКИ

В кваліфікаційній роботі розглянуто створення програмного модуля для опрацювання даних вимірювання напруги в середовищі LabVIEW.

1. Було проаналізовано особливості отримання та опрацювання даних вимірювання напруги. Встановлено, що ручне зчитування окремих значень не є достатньо зручним у випадках, коли потрібно аналізувати зміну сигналу в часі.

2. Було розглянуто середовища візуального програмування для роботи з вимірювальними даними. Для реалізації поставленої задачі використано LabVIEW, бо воно орієнтоване на створення віртуальних приладів, має зручні засоби для побудови лицьової панелі, блок-діаграми, графіків, роботи з масивами та збереження результатів.

3. На основі проведеного аналізу було розроблено структуру програмного модуля для опрацювання даних вимірювання напруги.

4. Було сформовано алгоритми роботи програмного модуля.

5. Було розроблено інформаційне забезпечення модуля. Визначено основні інформаційні об'єкти.

6. Було створено інтерфейс користувача програмного модуля у вигляді лицьової панелі. Інтерфейс поділено на зони введення даних, налаштування параметрів, керування, графічного відображення, числових результатів і службових повідомлень.

7. Тестування показало коректну роботу програмного модуля в основних сценаріях. Перевірено роботу з тестовим масивом значень напруги, правильність обчислення мінімального, максимального та середнього значення, контроль допустимих меж, реакцію на некоректні параметри та збереження результатів у файл.

В результаті виконання роботи створено модуль LabVIEW, який виконує базове опрацювання даних вимірювання напруги. Розроблений модуль може використовуватися як базовий віртуальний прилад для роботи з вимірювальними даними.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Glowinski S., Pecolt S., Blazejewski A., Sobieraj M. Design of a Low-Cost Measurement Module for the Acquisition of Analogue Voltage Signals. *Electronics*. 2023. Vol. 12, No. 3. Article 610. DOI: <https://doi.org/10.3390/electronics12030610>. URL: <https://www.mdpi.com/2079-9292/12/3/610> (дата звернення: 10.03.2026).
2. Гуменюк Д. В. Установка для автоматизованого вимірювання вольт-амперних характеристик. *ElectronAcoustEng*. 2019. Vol. 2, No. 4. С. 39–44. URL: <https://ela.kpi.ua/bitstreams/e53451fe-e89e-4489-b631-778a7e4db871/download> (дата звернення: 10.03.2026).
3. Гуменюк Д. В. Автоматизована система визначення параметрів моделі сонячних елементів : дипломний проєкт бакалавра. Київ : КПІ ім. Ігоря Сікорського, 2019. 101 с. URL: https://ela.kpi.ua/bitstream/123456789/28083/1/Humeniuk_bakalavr.pdf (дата звернення: 10.03.2026).
4. Al Mamun M. A., Rahman M. A., Sattar S., Begum M. Microcontroller Based Multichannel Data Acquisition System Using LabVIEW. *International Journal of Recent Technology and Engineering*. 2019. Vol. 8, Issue 2. P. 6538–6543. DOI: <https://doi.org/10.35940/ijrte.B3173.078219>. URL: https://www.researchgate.net/publication/364118388_Microcontroller_Based_Multichannel_Data_Acquisition_System_using_LabVIEW (дата звернення: 10.04.2026).
5. Teng J.-H., Chan S.-Y., Lee J.-C., Lee R. A LabVIEW Based Virtual Instrument for Power Analyzers. 2000 International Conference on Power System Technology Proceedings. 2000. Vol. 1. P. 179–184. URL: https://kucuker.sakarya.edu.tr/sites/kucuker.sakarya.edu.tr/file/A_LabVIEW_based_virtual_instrument_for_power_analyzers.pdf (дата звернення: 10.04.2026).
6. Abdel Mageed H. M., El-Rifaie A. M. Electrical Metrology Applications of LabVIEW Software. *Journal of Software Engineering and Applications*. 2013. Vol. 6, No. 3. P. 113–120. DOI: <https://doi.org/10.4236/jsea.2013.63015>. URL: <https://www.scirp.org/journal/paperinformation?paperid=29508> (дата звернення: 10.04.2026).

7. Watjanatepin N. Application of LabVIEW and Compact Data Acquisition to Study the Behavior of the Photovoltaic Stand-Alone Energy System. Australian Journal of Basic and Applied Sciences. 2015. Vol. 9, No. 32. P. 59–65. URL: <https://www.ajbasweb.com/old/ajbas/2015/Special%20IPN%20Oct/59-65.pdf> (дата звернення: 10.04.2026).

8. Makan G., Mingesz R., Gingl Z. How accurate is an Arduino Ohmmeter? Physics Education. 2019. Vol. 54, No. 3. Article 033001. DOI: <https://doi.org/10.1088/1361-6552/ab0910>. URL: <https://arxiv.org/abs/1901.03811> (дата звернення: 10.04.2026).

9. Simić M. Microcontroller-based Readout of Resistive Sensors. IFAC-PapersOnLine. 2022. Vol. 55, No. 4. P. 242–247. DOI: <https://doi.org/10.1016/j.ifacol.2022.06.040>. URL: <https://zenodo.org/records/10446295> (дата звернення: 10.04.2026).

10. Mingesz R., Gingl Z., Mellar J., Vadai G. Compact USB-based Instruments for Education and Remote Laboratory Projects. Proceeding Book of the Joint International Conference MPTL'16 – HSCI 2011. Ljubljana : University of Ljubljana, 2011. P. 144–150. URL: <https://arxiv.org/abs/1108.2155> (дата звернення: 10.04.2026).

11. Mingesz R., Gingl Z. Measurement and Data Acquisition : English Supplement. 2014. URL: <https://eta.bibl.u-szeged.hu/899/> (дата звернення: 10.06.2026).

12. Greene H. A LabVIEW-Based Open-Source Data Acquisition Program for Acoustic Emission Sensing : Mechanical Engineering Undergraduate Honors Thesis. Fayetteville : University of Arkansas, 2025. URL: <https://scholarworks.uark.edu/meeguht/138> (дата звернення: 10.04.2026).

13. Harris T., Pierson S., Pandey H., Hu H. Data Acquisition Using LabVIEW and MATLAB for Mechanical Engineering Laboratories. 2023 ASEE Midwest Section Conference. 2024. DOI: <https://doi.org/10.18260/1-2-1139-46375>. URL: <https://peer.asee.org/data-acquisition-using-labview-and-matlab-for-mechanical-engineering-laboratories> (дата звернення: 10.04.2026).

14. Павловський О. М., Цибульник С. О. Системи автоматизованого збору інформації. Комп'ютерний практикум : навч. посіб. Київ : КПІ ім. Ігоря

Сікорського, 2021. 70 с. URL: https://ela.kpi.ua/bitstream/123456789/45932/1/Systemy_avtomatyzovanoho_zboru_informatsii_KP.pdf (дата звернення: 10.05.2026).

15. Комп'ютерне проектування систем авіоніки: лабораторний практикум : навч. посіб. / уклад.: Ю. В. Бобков, А. А. Сердюк. Київ : КПІ ім. Ігоря Сікорського, 2022. 97 с. URL: https://ela.kpi.ua/bitstream/123456789/49823/1/Kompiuterne_proiektuvannia.pdf (дата звернення: 10.05.2026).

16. Гніденко В. Я. Віртуальний інструментарій для дослідження роботи відбивного клістроу : кваліфікаційна робота. Черкаси : Черкаський державний технологічний університет, 2023. URL: <https://er.chdtu.edu.ua/handle/ChSTU/5838> (дата звернення: 10.05.2026).

17. National Instruments. LabVIEW User Manual. Austin : National Instruments Corporation, 1998. URL: <https://download.ni.com/support/manuals/320999b.pdf> (дата звернення: 10.06.2026).

18. National Instruments. LabVIEW Function and VI Reference Manual. Austin : National Instruments Corporation, 1998. URL: <https://download.ni.com/support/manuals/321526a.pdf> (дата звернення: 10.05.2026).

19. National Instruments. LabVIEW Fundamentals. Austin : National Instruments Corporation, 2005. 165 р. URL: https://neurophysics.ucsd.edu/Manuals/National%20Instruments/LV_Fundamentals.pdf (дата звернення: 10.05.2026).

20. National Instruments. Getting Started with LabVIEW. Austin : National Instruments Corporation, 2003. URL: <https://download.ni.com/support/manuals/321527e.pdf> (дата звернення: 10.05.2026).

21. National Instruments. Taking an NI-DAQmx Measurement in NI LabVIEW SignalExpress. Austin : National Instruments Corporation, 2008. URL: <https://documentation.help/NI-LabView-NI-DAQmx-Measurement/firstpage.html> (дата звернення: 10.05.2026).

22. National Instruments. NI-DAQmx Help. Austin : National Instruments Corporation, 2008. URL: <https://documentation.help/NI-DAQmx/documentation.pdf> (дата звернення: 10.05.2026).

23. National Instruments. NI myDAQ User Guide and Specifications. Austin : National Instruments Corporation. URL: <https://www.ni.com/docs/en-US/bundle/mydaq-features/resource/373060g.pdf> (дата звернення: 10.05.2026).

24. National Instruments. Introduction to NI LabVIEW and Computer-Based Measurements. Austin : National Instruments Corporation, 2013. URL: https://indico.cern.ch/event/209985/session/20/attachments/327000/456071/Introduction_to_LabVIEW_ISOTDAQ_2013_.pdf (дата звернення: 10.05.2026).

25. National Instruments. Multisim User Guide. Austin : National Instruments Corporation. URL: <https://download.ni.com/support/manuals/374483a.pdf> (дата звернення: 10.05.2026).

26. Keithley Instruments. Data Acquisition Primer: An Introduction to Multi-Channel Measurement Systems. Cleveland : Keithley Instruments, 2020. URL: https://www.datatec.eu/media/26/6c/7b/1682501916/Keithley-DAQ6510-APP-1_DE.pdf (дата звернення: 10.05.2026).

27. MathWorks. Simulink Documentation. URL: <https://www.mathworks.com/help/simulink/index.html> (дата звернення: 10.05.2026).

28. Node-RED. User Guide. URL: <https://nodered.org/docs/user-guide/> (дата звернення: 10.06.2026).

29. Node-RED. Getting Started. URL: <https://nodered.org/docs/getting-started/> (дата звернення: 10.06.2026).

30. Simulink Basics Tutorial. University of Michigan. URL: https://ctms.engin.umich.edu/CTMS/index.php?aux=Basics_Simulink (дата звернення: 10.05.2026).

31. National Instruments. LabVIEW Tutorial. URL: <https://learn.ni.com/learn/article/labview-tutorial> (дата звернення: 10.05.2026).

32. National Instruments. NI-DAQmx User Manual. URL: <https://www.ni.com/docs/en-US/bundle/ni-daqmx/page/user-manual-welcome.html> (дата звернення: 10.05.2026).

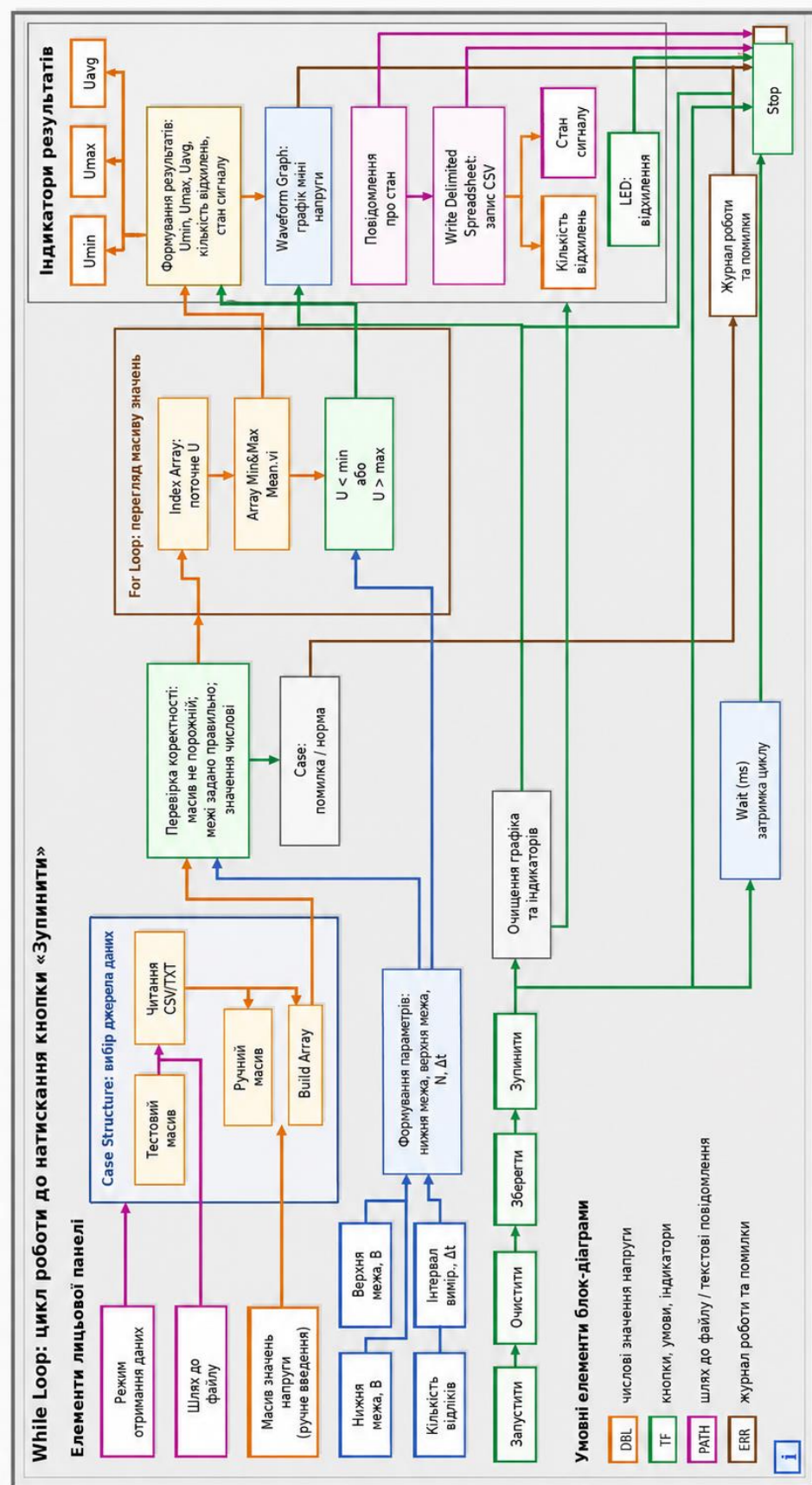
33. Kohout S., Roos J., Keller H. Automated Operation of a Home Made Torque Magnetometer Using LabVIEW. 2005. URL: <https://arxiv.org/abs/physics/0503005> (дата звернення: 10.05.2026).

34. Гуменюк Іван Ігорович, Програмний модуль у середовищі labview для опрацювання даних вимірювання напруги, XI Міжнародна наукова конференція «Традиційні та інноваційні підходи до наукових досліджень» (м. Луцьк, Україна).

35. Комар М.П., Саченко А.О., Васильків Н.М., Гладій Г.М., Коваль В.С., Ліп'яніна-Гончаренко Х.В. Методичні рекомендації до виконання кваліфікаційної роботи з освітньо-професійної програми «Комп'ютерні науки» спеціальності 122 «Комп'ютерні науки» за першим (бакалаврським) рівнем вищої освіти. Тернопіль: ЗУНУ, 2024. 52 с.

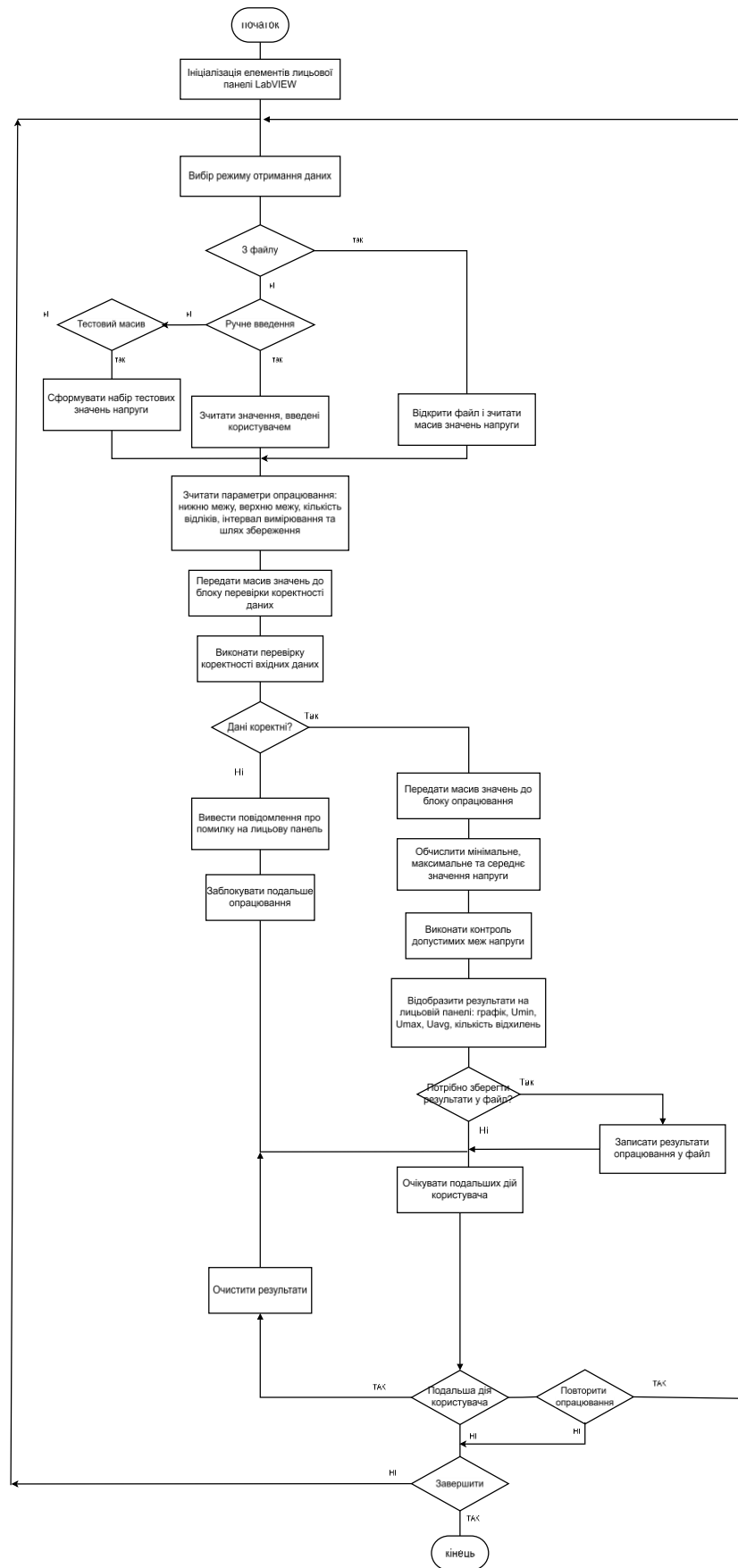
Додаток А

Блок - діаграма вимірювального модуля



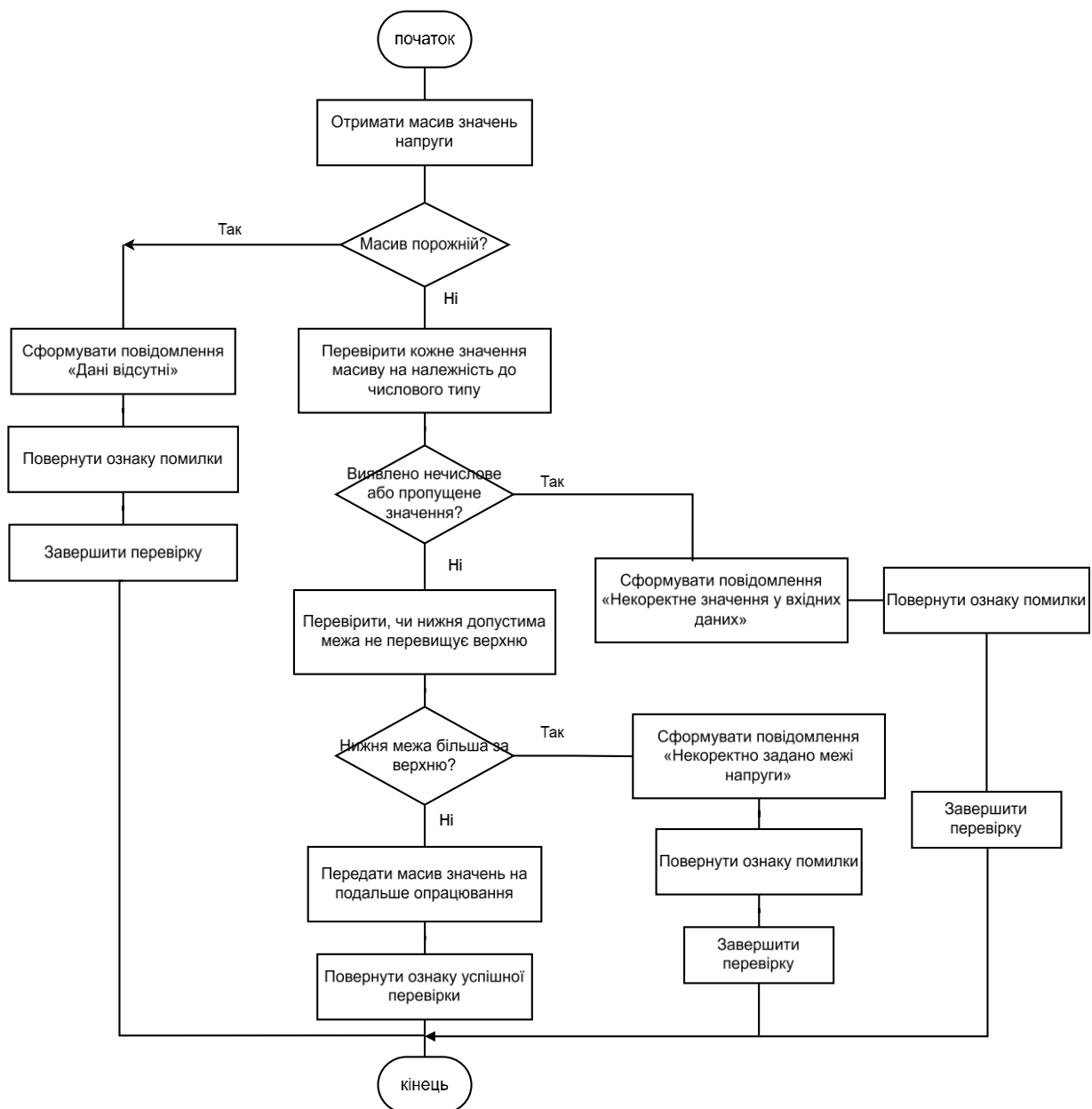
Додаток Б

Алгоритм функціонування модуля



Додаток В

Алгоритм опрацюванням масиву значень напруги



Додаток Д
Апробація отриманих результатів

ДОВІДКА

ПРО ПРИЙНЯТТЯ НАУКОВОЇ РОБОТИ ДО ПУБЛІКАЦІЇ

11.06.2026

Шановний(і) авторе(и):

Гуменюк Іван Ігорович ,

Організаційний комітет з радістю повідомляє, що наукову роботу ПРОГРАМНИЙ МОДУЛЬ У СЕРЕДОВИЩІ LABVIEW ДЛЯ ОПРАЦЮВАННЯ ДАНИХ ВИМІРЮВАННЯ НАПРУГИ» прийнято до публікації в збірнику за матеріалами XI Міжнародної наукової конференції «Традиційні та інноваційні підходи до наукових досліджень» (19.06.2026; м. Луцьк, Україна).

Опублікована робота буде доступна з 19.06.2026 за посиланням:

<https://archives.mcnd.org.ua/index.php/conference-proceeding/issue/view/19.06.2026>

.....

Електронні сертифікати учасників конференції будуть доступні з 19 червня.

Конференцію зареєстровано в базі даних науково-технічних заходів УкрІНТЕІ (Посвідчення № 176 від 26.01.2026). Матеріали конференції знаходитимуться в відкритому доступі (Open Access) на умовах ліцензії CC BY-SA 4.0 International.

З повагою,

Віце-президент МЦНД
Голова оргкомітету конференції
НАСТАСІЯ РАБЕЙ



ПРОГРАМНИЙ МОДУЛЬ У СЕРЕДОВИЩІ LABVIEW ДЛЯ ОПРАЦЮВАННЯ ДАНИХ ВИМІРЮВАННЯ НАПРУГИ

Гуменюк Іван Ігорович

Бакалавр спеціальності 122 – Комп’ютерні науки

Західноукраїнський національний університет, Україна

Науковий керівник: Кочан Володимир Володимирович

ORCID ID: 0000-0001-8376-4660

к.т.н., професор кафедри інформаційно-обчислювальних систем і управління

Західноукраїнський національний університет

Україна

Вступ

В сучасних комп’ютеризованих вимірювальних системах важливо не лише отримати окреме значення електричної величини, але також забезпечити його подальше опрацювання, візуалізацію та збереження. Це стосується вимірювання напруги, оскільки за її значенням можна оцінювати стан джерел живлення, електронних схем, сенсорних вузлів та інших технічних об’єктів. Якщо сигнал змінюється в часі, ручне зчитування показників не дає достатнього уявлення про поведінку системи, тому необхідно працювати з послідовністю виміряних значень.

Для таких задач доцільно застосовувати середовища візуального програмування, у яких логіка роботи подається через функціональні блоки та зв’язки між ними. LabVIEW є одним із поширених засобів створення віртуальних приладів, оскільки поєднує лицьову панель, блок-діаграму, засоби роботи з масивами, графіками та файлами [1–3]. Використання такого середовища дозволяє швидко побудувати модуль, який приймає дані напруги, перевіряє їх, обчислює основні показники та відображає результати у зручному для користувача вигляді.

Структура програмного модуля

Програмний модуль реалізовано як віртуальний прилад LabVIEW, що складається з лицьової панелі та блок-діаграми. Лицьова панель забезпечує взаємодію з користувачем через кнопки, числові поля, графіки й індикатори, а блок-діаграма відповідає за алгоритмічну частину опрацювання даних.

У структурі модуля виділено блоки введення даних, налаштування параметрів, перевірки коректності, опрацювання значень, контролю меж, візуалізації, збереження результатів і керування.

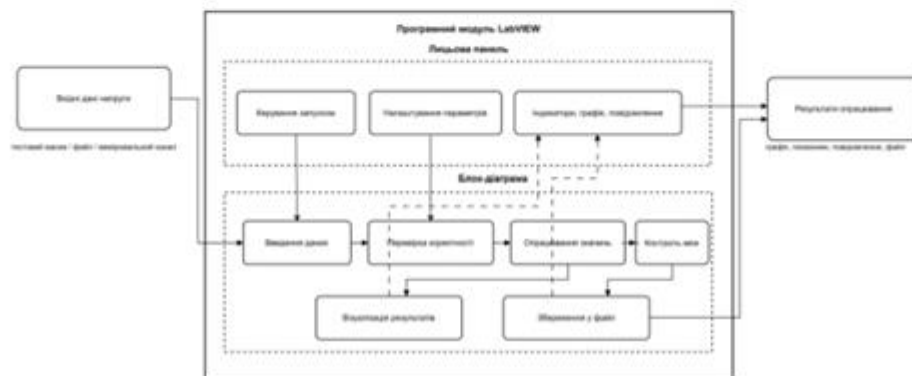


Рис. 1. Загальна структура програмного модуля опрацювання даних вимірювання напруги

Алгоритм роботи передбачає послідовне проходження даних від введення до збереження. Спочатку програма перевіряє наявність масиву значень, коректність числових даних і правильність заданих меж. В разі помилки подальша обробка блокується, а користувач отримує відповідне повідомлення. Якщо перевірка успішна, модуль обчислює мінімальне, максимальне та середнє значення напруги.

Додатково кожне значення порівнюється з нижньою та верхньою межею. Значення поза допустимим діапазоном фіксуються як відхилення, а їх кількість відображається на лицьовій панелі. Це дає змогу швидко оцінити стан сигналу. Результати опрацювання можуть бути збережені у CSV-файл для подальшого аналізу в табличних редакторах.

Інтерфейс користувача та тестування

Інтерфейс модуля побудовано так, щоб користувач міг виконати основні дії без редагування блок-діаграми. Лицьова панель поділена на зони введення даних, налаштування параметрів, керування, графічного відображення, числових результатів і службових повідомлень. В зоні введення обирається джерело даних: тестовий масив, файл або ручне введення. У зоні параметрів задаються допустимі межі напруги, кількість відліків та інтервал вимірювання. Блок керування містить кнопки запуску, зупинки, очищення і збереження результатів.

Основним елементом візуалізації є графік зміни напруги, який дає змогу оцінити характер сигналу, наявність спадів, стрибків або виходу за допустимі межі. Поруч із графіком розміщуються числові індикатори: мінімальне, максимальне та середнє значення напруги, а також кількість відхилень. Додатковий індикатор стану повідомляє, чи обробку виконано успішно, чи виявлено помилку або значення поза межами.



Рис. 2. Тестування контролю допустимих меж напруги

Тестування виконувалося для кількох сценаріїв роботи. В нормальному режимі модуль приймав масив значень, перевіряв його, обчислював основні параметри та будував графік. Правильність обчислень перевірялася на контрольних наборах даних із заздалегідь відомими значеннями.

Також перевірено реакцію на порожній масив, неправильно задані межі та відсутній потік даних або масив. В таких випадках модуль зупиняв обробку і показував повідомлення про помилку.

Висновки. Розроблено програмний модуль у середовищі LabVIEW для опрацювання даних вимірювання напруги. Модуль підтримує введення даних, налаштування параметрів, перевірку коректності, обчислення основних показників, контроль меж, графічне відображення та збереження результатів. Завдяки такій структурі модуль може використовуватися як базовий віртуальний прилад для роботи з вимірювальними даними.

Список використаних джерел:

1. Glowinski S., Pecolt S., Blazejewski A., Sobieraj M. Design of a Low-Cost Measurement Module for the Acquisition of Analogue Voltage Signals. *Electronics*. 2023. Vol. 12, No. 3. Article 610. DOI: <https://doi.org/10.3390/electronics12030610>.
2. Al Mamun M. A., Rahman M. A., Sattar S., Begum M. Microcontroller Based Multichannel Data Acquisition System Using LabVIEW. *International Journal of Recent Technology and Engineering*. 2019. Vol. 8, Issue 2. P. 6538-6543.
3. Teng, J. H., Chan, S. Y., Lee, J. C., & Lee, R. (2000, December). A LabVIEW based virtual instrument for power analyzers. In *PowerCon 2000. 2000 International Conference on Power System Technology. Proceedings (Cat. No. 00EX409)* (Vol. 1, pp. 179-184). IEEE.
4. National Instruments. LabVIEW User Manual. URL: <https://www.ni.com/docs/en-US/bundle/labview/page/user-manual-welcome.html> (дата звернення: 10.04.2026).
5. National Instruments. LabVIEW Fundamentals. Austin : National Instruments Corporation, 2005. 165 p. URL: https://www.testforce.com/media/sparsh/product_attachment/LabVIEW_Fundamentals_compressed.pdf (дата звернення: 10.04.2026).