

Міністерство освіти і науки України
Чернівецький національний університет
імені Юрія Федьковича

**Програмне забезпечення
інформаційно-вимірювальних систем**
Методичні рекомендації
до лабораторного практикуму

Чернівці 2021

УДК 004.4'232:681.5](076.5)
П784

*Рекомендовано Вченою Радою
Інституту фізико-технічних та комп'ютерних наук
Чернівецького національного університету імені Юрія Федьковича
(Протокол № 9 від 24.09.2021)*

Укладач: Івашко В. В., канд. фіз.-мат. наук, асистент

Івашко В. В. Програмне забезпечення інформаційно-вимірювальних систем: метод. реком. до лабор. практикуму. Чернівці : Чернівецький національний університет імені Юрія Федьковича, 2021. 35 с.

У методичній розробці наведено роботи лабораторного практикуму з предмета: «Програмне забезпечення інформаційно-вимірювальних систем». Зміст робіт охоплює основні розділи курсу.

Для студентів технічних факультетів та інститутів за спеціальністю: «Метрологія та інформаційно-вимірювальна техніка».

УДК 004.4'232:681.5](076.5)
П784

© Чернівецький національний університет
імені Юрія Федьковича, 2021

Зміст

Лабораторна робота № 1

«Елементи вводу/виводу в LabVIEW».....4

Лабораторна робота № 2

«Робота з циклами та графічне виведення даних в LabVIEW».....13

Лабораторна робота № 3

«Робота з структурами Case та Sequence в LabVIEW».....20

Лабораторна робота № 4

«Робота з масивами в LabVIEW».....25

Лабораторна робота № 5

«Рядки та рядкові елементи управління в середовищі LabVIEW».....31

Список рекомендованої літератури.....35

Лабораторна робота № 1

«Елементи вводу/виводу в LabVIEW»

Мета роботи: ознайомлення з програмним пакетом LabVIEW.

Теоретичні відомості

LabVIEW, на відміну від мов програмування таких, як C, Pascal або Python, використовує графічну мову програмування, та призначений для створення програм у формі структурних схем. LabView містить великі бібліотеки функцій та інструментальних засобів, призначених для створення систем збору даних і систем автоматизованого управління. LabView також включає стандартні інструментальні засоби розробки програм.

Програми в LabVIEW називаються віртуальними приладами (ВП або VI, Virtual instrument – англ.), оскільки їх вид і функції імітують реальні вимірювальні прилади. Однак, при цьому віртуальні прилади подібні до функцій в програмах стандартних мов програмування.

Структура віртуального приладу представлена наступними елементами:

- Передня панель або лицьова панель (може містити кнопки, перемикачі, регулятори та інші органи управління, індикатори);
- Блок-діаграма або функціональна панель (являє собою наочне уявлення рішення задачі і містить вихідні коди для віртуального приладу).

LabVIEW дотримується концепції модульного програмування, тобто прикладна програма ділиться на декілька простіших підпрограм, які по суті є окремими віртуальними приладами. Дані підпрограми об'єднують в більш складніший віртуальний прилад на загальній структурній схемі, який в подальшому і буде виконувати основну поставлену задачу. В результаті основний віртуальний прилад верхнього рівня містить сукупність, так званих, підприладів (subVI), які реалізують основні функції прикладної програми. При цьому, багато subVI низького рівня часто здатні виконувати завдання, які по своєму змісту є загальними для декількох прикладних програм вищого рівня. Відповідно, існує можливість розробити окремий спеціалізований набір subVI, який буде добре підходити для подальшого використання в майбутніх прикладних програмах вищого рівня.

При появі елемента управління або індикатора на передній панелі, LabVIEW поміщає відповідний термінал на структурну схему. Піктограми функцій також мають термінали. Дані, які вводяться в елементи управління, надходять з передньої панелі через термінали елементів управління на блок-діаграму. Потім дані надходять у функції. Коли функції завершують свої внутрішні обчислення, вони утворюють нові значення даних на своїх

вихідних терміналах. Дані надходять на термінали індикаторів і повторно потрапляють на передню панель, де вони відображаються.

Вузли – це елементи виконання програми. Вони аналогічні інструкціям, операторам, функціям і підпрограмам в стандартних мовах програмування.

Функція – один з типів вузлів. LabVIEW має велику бібліотеку функцій для математичних обчислень, порівняння, перетворення, введення / виведення і так далі. Інший тип вузлів – структура.

Структури є графічним представленням циклів і операторів вибору традиційних мов програмування, повторюючи блоки інструкцій або виконуючи їх за умовою.

Провідники – це шляхи даних між терміналами джерела і адресата. Не можна підключити термінал-джерело до іншого джерела, потрібно підключати термінал-джерело до іншого термінал-адресата. Можна підключати одне джерело до декількох адресатів. Провідники мають різний вигляд або колір, в залежності від типу даних, які по них передаються.

Принцип, який управляє виконанням програми в LabVIEW, називається потоком даних. Запущений вузол виконується тільки тоді, коли на всіх входах з'являються дані; вузол видає дані на всі вихідні термінали тільки тоді, коли він закінчує виконання; і дані відразу ж надходять від терміналу джерела на термінал адресата. Потік управління регулюється командами. Потік даних – управляється даними або залежить від даних.

Багато з інструментів LabVIEW містяться в меню (або палітра) Tools (рис. 1). При створенні нового віртуального приладу, або завантаженні існуючого, на екрані з'являється вікно передньої панелі.



Рис.1.

Кнопка «вказівний палець» використовується для зміни позицій кнопок і вимикачів, налаштування віртуальних осцилографів, управління значеннями цифрових регуляторів та інше.

Кнопка «стрілка» служить для виділення, переміщення об'єктів, зміни їх розмірів.

Кнопка «котушка» використовується для з'єднання об'єктів на блок-діаграмі.

Кнопка «пензлик» застосовується для розфарбовування фону або об'єктів.

Кнопка «рука» служить для переміщення робочої області панелі у вікні.

Кнопка «піпетка» застосовується для вибору поточного кольору з наявних на панелі.

Кнопка «А» служить для відкриття і редагування текстового вікна;

Кнопка «червоне коло» необхідна для розміщення та зняття точок зупинки виконання програми на блок-діаграмі.

Кнопка «Р» служить для розміщення на блок-діаграмі локальних вікон, що відображають поточні значення даних, що передаються в ході виконання програми.

Перейти до вікна передньої панелі з вікна структурної схеми можна, вибравши в меню Windows >> Show Panel.

Об'єкти на передній панелі створюються при виборі їх з палітри «Controls» (Windows >> Show Controls Palette).

При створенні об'єкти на передній панелі з'являються з у формі прямокутника з міткою (назвою), в який відразу ж можна ввести текст – назва елементу управління або індикатора.

Контекстне меню викликається натисненням правої кнопки миші, коли курсор у вигляді руки або стрілки знаходиться на об'єкті.

Створена назва об'єкта редагується за допомоги відповідного інструменту з меню Tools.

У середовищі LabVIEW провідники даних використовуються для з'єднання численних терміналів даних. Поля введення/виведення повинні бути сумісними з типами даних, що передаються їм через провідники. Наприклад, не можна з'єднувати поле виведення масиву з полем введення даних чисельного типу. Також характер з'єднання повинен бути коректним. Провідники повинні бути приєднані лише до одного джерела даних і до одного поля введення даних. Наприклад, не можна поєднувати 2 елементи відображення.

В LabVIEW використовуються такі типи даних:






- Numeric (чисельний тип) Floating point – число з рухомою комою, відображається у вигляді помаранчевих терміналів. Може бути представлено у вигляді single (32 bit), double (64-bit) або extended (128-bit) precision (з одиночною, подвійною або розширеною точністю);
- Integer - цілочисельний тип, відображається у вигляді блакитних терміналів. Можливі три представлення цілих чисел: 8, 16 і 32 біта.

Один біт може використовуватися для знака числа, якщо це число є знаковим цілим;

- Boolean – логічний тип, відображається у вигляді зелених терміналів;
- String – строковий тип, відображається у вигляді рожевих терміналів. Строковий тип даних містить текст у ASCII форматі;
- Path – шлях до файлу, відображається у вигляді терміналів;
- Array – масиви включають типи даних складових елементів і приймають відповідний їм колір;
- Cluster – кластери включають різні типи даних;
- Waveform – сигнальний тип даних є кластером елементів, що містить дані, початкове значення часу та інтервал часу між вимірами;
- Dynamic – динамічний тип, відображається у вигляді темно-синіх терміналів.

Дані між об'єктами блок-діаграми передаються по з'єднувальних лініях провідникам даних. Провідник даних аналогічний змінним в текстових мовах програмування. Кожен провідник даних має єдине джерело даних, але може передавати їх до багатьох ВП. Провідники даних розрізняються кольором, стилем і товщиною лінії, в залежності від типу даних що передаються. Приклади основних типів провідників даних представлені нижче (табл. 1.).

Табл. 1. Типи провідників

Тип провідника	Одне значення	1D масив	2D масив	Колір
Числовий				Помаранчевий (з рухомою точкою), Блакитний (цілочисельний)
Логічний				Зелений
Рядковий				Рожевий

У середовищі LabVIEW об'єкти з'єднуються провідниками даних після їх утворення на блок-діаграму. В автоматичному режимі LabVIEW підключає ті поля введення/виведення даних, які найбільш сумісні, несумісні поля залишаються роз'єднаними.

Якщо обраний об'єкт поміщається на блок-діаграмі недалеко від іншого об'єкта, LabVIEW показує пунктирні тимчасові провідники даних, вказуючи на області можливого з'єднання. Слід звернути увагу, що при відпуску кнопки миші LabVIEW автоматично підключає провідник даних до поля введення/виведення даних, обраного об'єкта.

Коригування параметрів автоматичного підключення провідників здійснюється через пункти головного меню Tools >> Options >> Block Diagram.

З'єднання об'єктів провідниками даних вручну проводиться за допомогою інструменту «котушка». Після наведення цього інструменту на поле введення або виведення даних на екрані з'являється підказка, яку можна використовувати для уточнення місця підключення провідника.

Хід роботи

1. Створіть новий віртуальний прилад, File >> New VI.
2. На передній панелі (Window>> Show Front Panel) розмістіть чотири керуючих елемента для введення чисел A, B, C, D. Для цього, з палітри Controls >> Numeric виберіть цифровий елемент введення чисел Numeric Control. Додайте елемент на поле передньої панелі, і в написі, що з'явився введіть назву змінних (Tools >> Edit Text).
3. Також, на передній панелі створіть три цифрових індикатора для виведення результатів обчислення. Для цього, з палітри Controls >> Numeric виберіть цифровий елемент виведення чисел Numeric Indicator. Розмістити елемент на поле передньої панелі, і в написі, що з'явився введіть назву (Tools >> Edit Text): $A + B$, $A - B$ та $A * B$.
4. На даний момент передня панель повинна виглядати так, як показано на рис. 2.

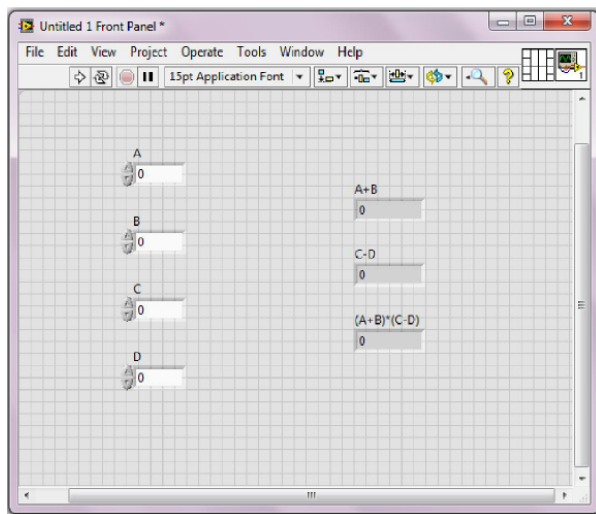


Рис. 2.

5. Перейдіть у вікно блок-діаграми (Windows >> Show Block Diagram). На блок-діаграмі розміщені термінали, відповідні елементам управління та індикаторам передньої панелі. Термінали мають ті ж написи, що і відповідні їм об'єкти передньої панелі.
6. Розташуйте на ній об'єкти управління (Numeric Control) зліва, а індикатори (Numeric Indicator) праворуч аналогічно тому, як вони розташовані на передній панелі.

7. Виберіть функції обчислювального пристрою з палітри Functions >> Programming >> Numeric: додавання (Add), віднімання (Subtract), множення (Multiply) (рис. 3).

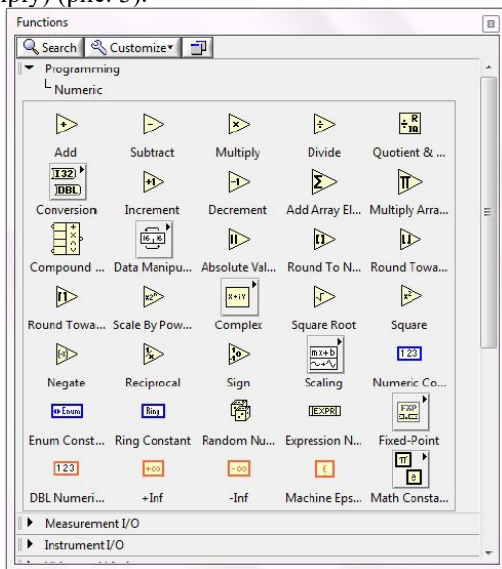


Рис. 3. Блок діаграма простого обчислювального пристрою

8. З палітри Tools виберіть Connect Wire («когушку»). З'єднайте між собою термінали елементів управління, функцій та індикаторів.

9. На цьому монтаж блок-діаграми (структурної схеми) закінчений. На даний момент структурна схема приладу повинна виглядати, так як показано на рис. 4.

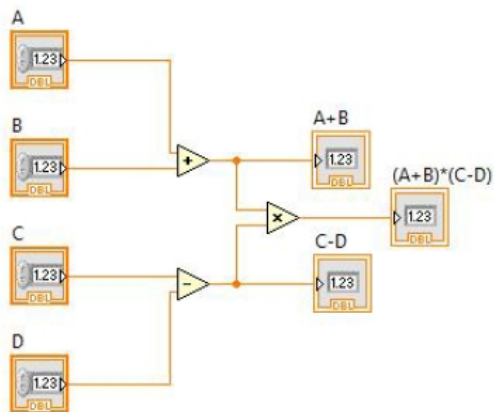


Рис. 4.

10. Перейдіть у вікно передньої панелі та запустіть віртуальний прилад, натиснувши кнопку Run (запуск) в лівому верхньому кутку вікна.

11. Виконаємо ті ж арифметичні обчислення за допомогою функції Formula Node. Перейдіть на блок-діаграму. З палітри Functions >> Programming >> Structures >> виберіть Formula Node. (рис. 5)

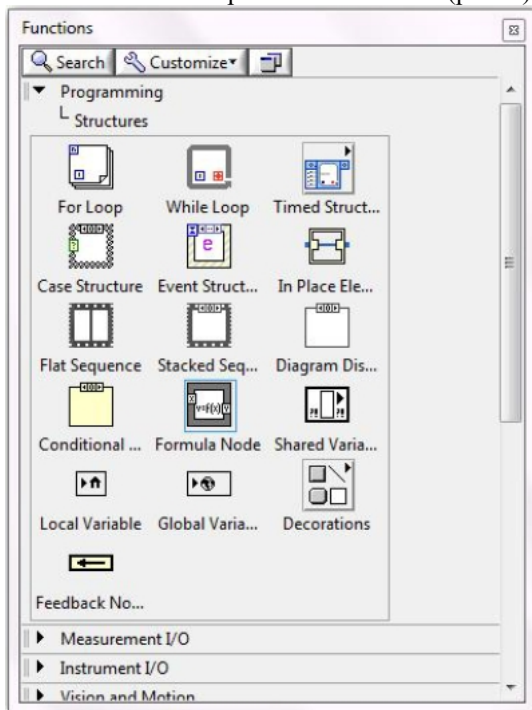


Рис. 5.

12. Створимо прямокутну область Formula Node, на лівій межі якої створіть чотири входи: A, B, C, D (права кнопка миші по лівому краю Formula Node >> контекстне меню >> Add Input), а на правій три виходи: S1, S2, S3 (права кнопка миші по правому краю Formula Node >> контекстне меню >> Add Output).

13. З палітри Controls >> Numeric виберіть цифровий елемент введення чисел Numeric Indicators. Створіть три індикатора з іменами S1, S2, S3.

14. В поле об'єкта Formula Node запишіть арифметичні операції у вигляді формул: $S1 = A + B$; $S2 = C - D$; $S3 = S1 * S2$.

15. З палітри Tools виберіть Connect Wire (котушка). З'єднайте між собою чотири керуючих елементів для введення чисел A, B, C, D і чотири входів (A, B, C, D) області Formula Node, і так само виходи області Formula Node з індикаторами з іменами S1, S2, S3.

16. На цьому монтаж блок-діаграми (структурної схеми) закінчений. На даний момент структурна схема приладу повинна виглядати, так як показано на рис. 6.

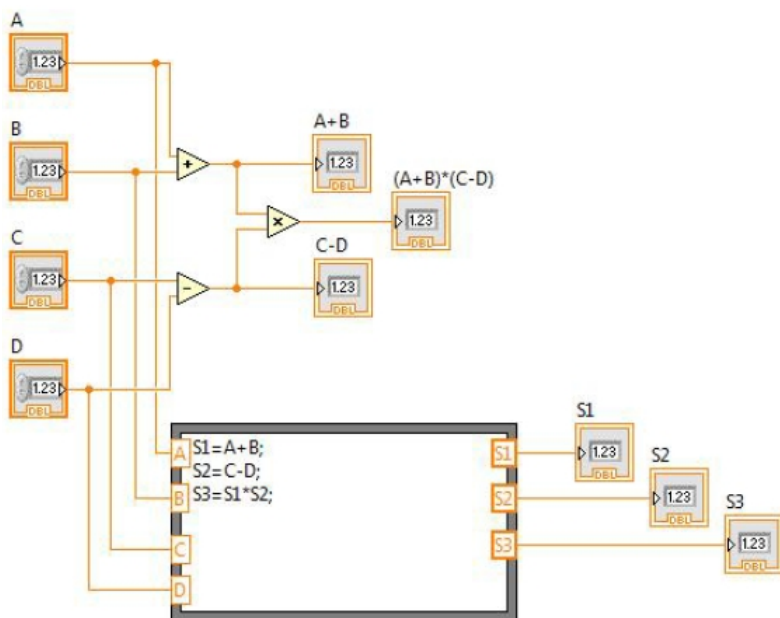


Рис. 6

17. Перейдіть у вікно передньої панелі і запустіть віртуальний прилад, натиснувши кнопку Run в лівому верхньому кутку вікна. Збережіть отриманий результат.

18. Контрольне завдання: створіть віртуальний прилад у відповідності до вашого варіанту завдання.

Табл. № 2

№	Завдання
1	Віртуальний прилад, який перетворює значення температури з градусів Цельсія в температуру за шкалою Кельвіна ($^{\circ}\text{K} = ^{\circ}\text{C} + 273$).
2	Віртуальний прилад, який перетворює значення температури з градусів Цельсія в температуру за шкалою Реомюра ($^{\circ}\text{R} = ^{\circ}\text{C} * 4/5$).
3	Віртуальний прилад, який перетворює значення температури по шкалі Кельвіна в градуси Цельсія ($^{\circ}\text{C} = ^{\circ}\text{K} - 273$).
4	Віртуальний прилад, який перетворює значення температури по шкалі Реомюра в градуси Цельсія ($^{\circ}\text{C} = ^{\circ}\text{R} * 5/4$).
5	Віртуальний прилад, який перетворює значення температури за Фаренгейтом в градуси Цельсія ($^{\circ}\text{F} = 9/5 * ^{\circ}\text{C} + 32$).

6	Віртуальний прилад, який перетворює значення температури по шкалі Реомюра в температуру за Фаренгейтом ($^{\circ}\text{F} = 9/4 * ^{\circ}\text{R} + 32$).
7	Віртуальний прилад, який перетворює значення температури з градусів Цельсія в температуру по Реомюр ($^{\circ}\text{R} = ^{\circ}\text{C} * 4/5$).
8	Віртуальний прилад, який перетворює значення температури по Кельвіну в температуру по Реомюр ($^{\circ}\text{R} = (^{\circ}\text{K} - 273) * 4/5$).
9	Віртуальний прилад, який перетворює згідно із законом Ома значення напруги і опору в силу струму ($I = U / R$).
10	Віртуальний прилад, який перетворює значення напруги (В) і сили струму (А) в потужність (Вт) ($P = I * U$)

Контрольні питання

1. Що таке LabVIEW? З яких основних елементів воно складається? Опишіть їх.
2. Що таке вузли, функції та провідники в LabVIEW?
3. Які типи даних використовує LabVIEW? Опишіть їх.
4. Які типи провідників використовує LabVIEW? Опишіть їх.
5. Для чого служить палітра Controls? Опишіть її.
6. Для чого служить палітра Functions? Опишіть її.

Лабораторна робота № 2

«Робота з циклами та графічне виведення даних у LabVIEW»

Мета роботи: вивчення методів використання терміналу вихідних даних циклу While loop та/або For loop (за умовою) та організації доступу до значень попередніх ітерацій циклу.

Теоретичні відомості

Структури в LabVIEW є графічним представленням операторів циклу і операторів умови (Case або Варіант), які використовуються у звичайних мовах програмування. Вони призначені для виконання повторюваних операцій над потоком даних, операцій в певному порядку і накладання умов на виконання операцій.

Середовище LabVIEW містить п'ять структур: Цикл While (за умовою), Цикл For (з фіксованим числом ітерацій), структура Case (Варіант), структура Sequence (Послідовність), структура Event (Подія), а також Formula Node (вузол Формули).

Блок-діаграма циклу While виконується до тих пір, поки не виконається умова виходу з циклу. За замовчуванням, термінал умови виходу має вигляд червоної кнопки. Це означає, що цикл буде виконуватися до надходження на термінал умови виходу значення True. В цьому випадку термінал умови виходу називається терміналом Stop If True (зупинка якщо істина). Термінал лічильника ітерацій показаний зліва, у формі літери «i», містить значення кількості виконаних ітерацій. Початкове значення терміналу завжди дорівнює нулю.

Передбачена можливість зміни умови виходу та відповідного йому зображення терміналу умови виходу. Шляхом клацанням правої кнопки миші по терміналу умови виходу або по межі циклу необхідно викликати контекстне меню і вибрати пункт Continue If True (продовження якщо істина).

Цикл For (з фіксованим числом ітерацій) виконує повторювані операції над потоком даних певну кількість разів. Він розташований в підпалітрі Structures палітри Functions. Значення, присвоєне терміналу максимального числа ітерацій «N» циклу, показаного ліворуч, визначає максимальну кількість повторень операцій над потоком даних. Термінал лічильника ітерацій, показаний зліва, у формі літери «i», містить значення кількості виконаних ітерацій. Початкове значення лічильника ітерацій завжди дорівнює нулю.

При роботі з циклами часто необхідний доступ до значень попередніх ітерацій циклу. Наприклад, в разі коли ВП, що вимірює температуру і відображає її на графіку, для відображення поточного середнього значення температури, необхідно використовувати значення,

отримані в попередніх ітераціях. Є два шляхи доступу до цих даних: Shift Register (зсувний регістр) і Feedback Node (вузол зворотного зв'язку).

Зсувний регістр виглядає як пара терміналів. Вони розташовані безпосередньо один на проти одного на протилежних вертикальних сторонах межі циклу. Правий термінал містить стрілку «вгору» і зберігає дані по завершенню поточної ітерації. LabVIEW передає дані з цього регістра в наступну ітерацію циклу. Зсувний регістр створюється клацанням правої кнопки миші по межі циклу і вибором з контекстного меню пункту Add Shift Register (додати зсувний регістр).

Щоби форматувати зсувний регістр, необхідно передати на його лівий термінал будь-яке значення ззовні циклу. Якщо не форматувати зсувний регістр, він використовує значення, записане в регістр під час останнього виконання циклу або значення, яке використовується за умовчанням для даного типу даних, якщо цикл ніколи не виконувався.

Вузол зворотного зв'язку, автоматично з'являється в циклах While або For при з'єднанні поля виведення даних підпрограми, функції або групи підпрограм і функцій з полем введення даних тих же самих підпрограм, функцій або їх груп. Як і зсувний регістр, вузол зворотного зв'язку зберігає дані будь-якого типу по завершенню поточної ітерації і передає ці значення в наступну ітерацію. Використання вузлів зворотного зв'язку дозволяє уникнути утворення великої кількості провідників та з'єднань.

Хід роботи

1. Відкрийте новий віртуальний прилад, File >> New VI.
2. На передній панелі (Window >> Show Front Panel) розмістіть два елемента керування для введення чисел A, B. (A – перша цифра варіанту, B – друга цифра варіанту) З палітри Controls >> Express >> Graph Indicator виберіть графік Chart. Додайте два графічних елементи на поле передньої панелі та в написі, що з'явився введіть назви графіків.
3. На даний момент передня панель повинна виглядати так, як показано на рис. 7.

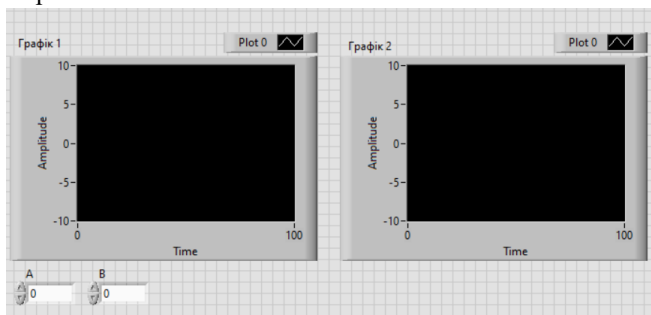


Рис. 7.

4. Перейдіть у вікно блок-діаграми (Windows >> Show Block Diagram). На блок-діаграмі розміщені термінали, відповідні елементи

управління та індикаторами передньої панелі. Термінали мають ті ж мітки (позначення), що і відповідні їм об'єкти передньої панелі.

5. Розмістіть об'єкти управління (Controls) зліва, а індикатори (Graph Indicator – Chart) праворуч.

6. Виберіть і розмістіть на поле блок-діаграми: For Loop і While loop, а також додатково в кожному з циклів Formula Node (рис. 8).

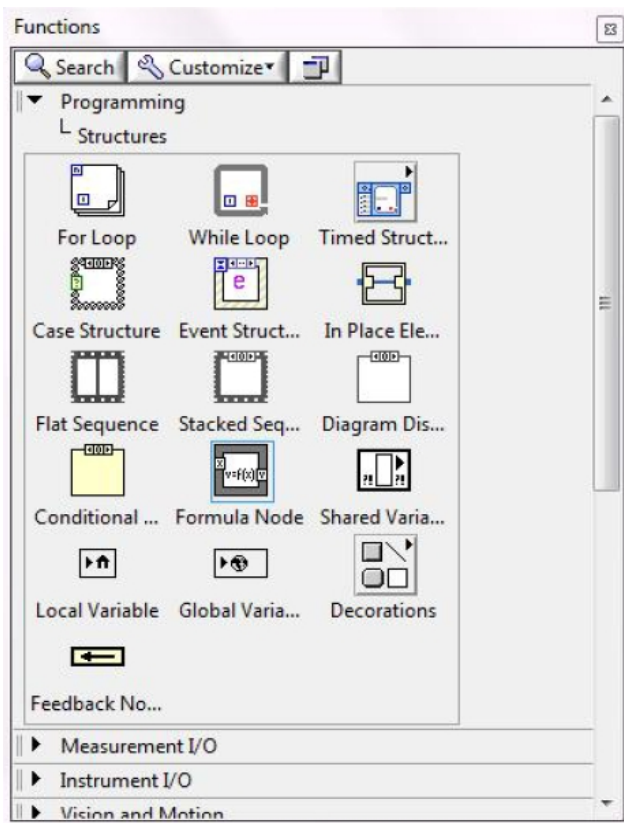


Рис. 8.

7. У кожній із структур Formula Node циклів For і While (вони ідентичні один одному) необхідно створити три входи A, B, x і вихід «у». Formula Node повинен містити арифметичне вираз:

$$y = \sin(A*x) + \sin(B*x);$$

8. Після з'єднання об'єктів управління з відповідними входами Formula Node і відповідні виходи з індикаторами необхідно задати кількість ітерацій $N = 1000$ (правою кнопкою миші на знаку ітерацій «N» Create constant) і використовуючи функцію Equal? підпалітри Comparison палітри Programming задати відповідну умову зупинки циклу While, як показано на рис. 9.

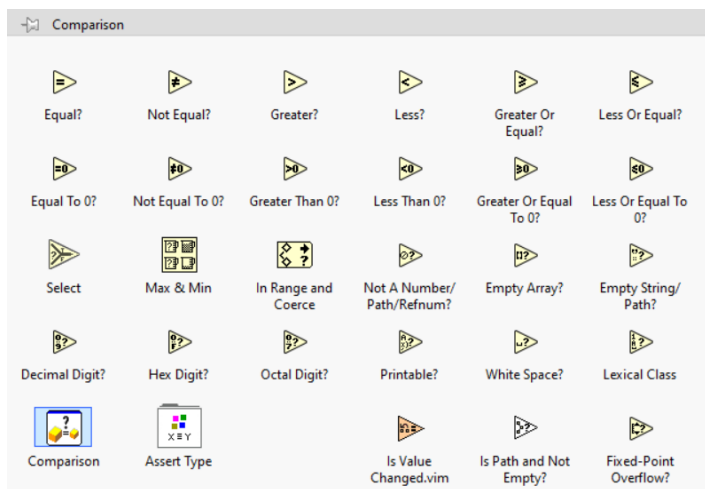


Рис. 9.

9. На даний момент блок-діаграма повинна виглядати так, як показано на рис. 10.

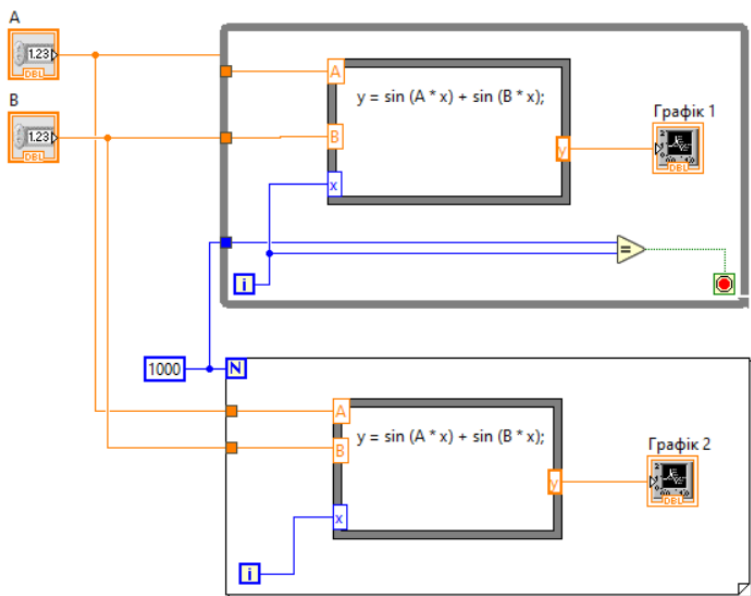


Рис. 10

10. Перейдіть у вікно передньої панелі і запустіть віртуальний прилад, натиснувши кнопку Run в лівому верхньому кутку вікна, попередньо задавши значення для A та B (наприклад, $A = B = 0,25$). Збережіть отриманий результат.

11. Удосконалимо створений вами у попередній роботі (лб. № 1) віртуальний пристрій переведення значення температури зі однієї шкали в іншу, за допомоги циклу While. Створіть новий ВП, File >> New VI.

12. Створіть структурну схему, яка зображена на рис. 11.

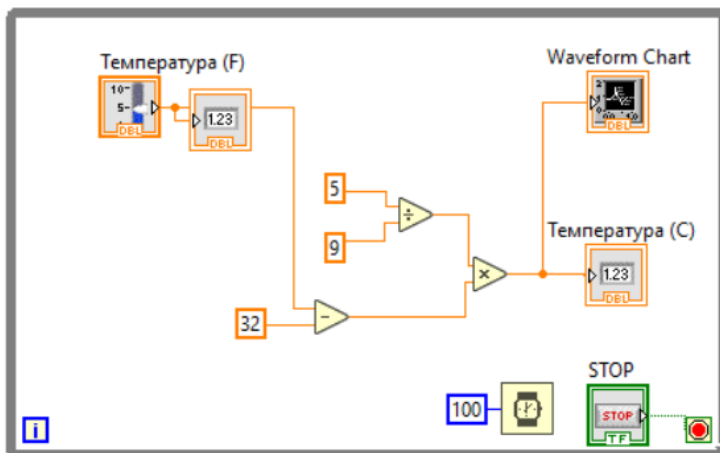


Рис. 11

Для створення константи до елемента схеми, достатньо правою кнопкою миші клацнути правою кнопкою миші по краю необхідного об'єкта та зі контекстного меню вибрати Create Constant. Затримку (Wait (ms)) можна знайти в Functions >> Programming >> Timing.

13. На даний момент передня панель вашого ВП повинна виглядати так, як показано на рис. 12. Збережіть отриманий результат.

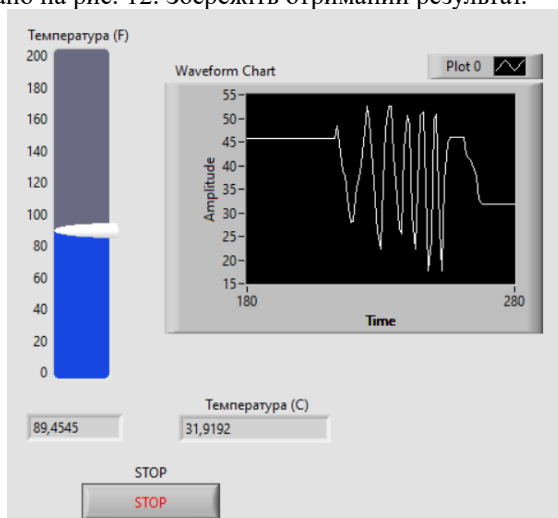


Рис. 12

14. Контрольне завдання: створіть віртуальний прилад, який генерувати періодичний синусоїдальний сигнал з відповідною амплітудою «А», фазою «φ» та частотою «f» згідно зі номером варіанту (табл. 3):

Табл. № 3

№	A	φ , °	f, Гц
1	100	30	50
2	110	45	60
3	120	60	100
4	130	75	120
5	140	90	145
6	150	105	170
7	160	120	195
8	170	135	220
9	180	150	245
10	190	165	270

Структурна схема даного пристрою має вигляд, який зображений на рис. 13. Елемент, який відповідає за генерацію сигналу (Simulate Signal) можна знайти в Functions >> Express >> Input.

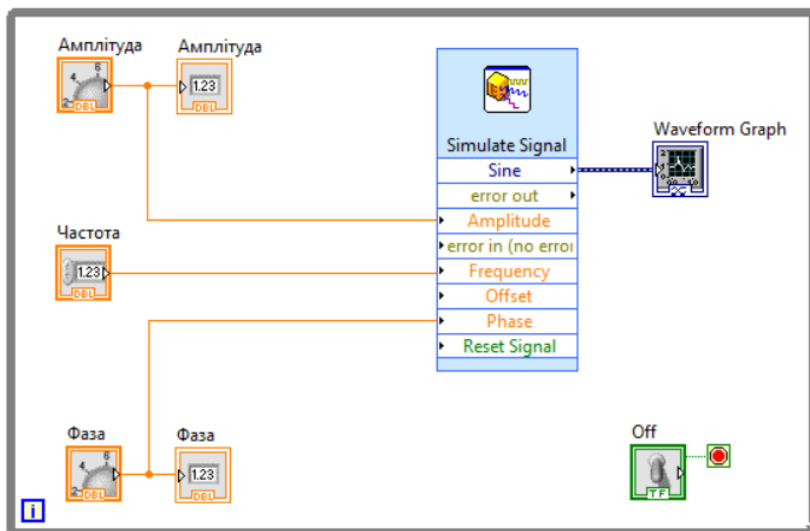


Рис. 13

Контрольні питання

1. Що таке структура? Які бувають структури в LabVIEW?
2. Опишіть принцип роботи циклу While.
3. Опишіть принцип роботи циклу For.
4. Для чого використовуються зсувні регістри та вузол зворотного зв'язку?
5. Для чого потрібна структура Formula Node?

Лабораторна робота № 3

«Робота зі структурами Case та Sequence в LabVIEW»

Мета роботи: вивчення способів використання терміналу вихідних даних структури Case (варіант) та Sequence (послідовність).

Теоретичні відомості

Структура варіанту (Case Structure) є методом виконання команд, що містить умову, тобто аналогом оператора імплікації (if then else). Структура варіанту має дві або більше піддіаграми, або умовних варіанти. Лише одна з них виконується в залежності від логічного, числового чи рядкового значення, яке подається на термінал селектора структури.

Якщо до терміналу селектора структури варіанту підключено логічне значення, то структура матиме два варіанти – Хибне та Істина. Якщо до терміналу селектора підключені числові або рядкові дані, то структура може мати майже необмежену кількість варіантів, починаючи з нульового. Для додавання варіантів необхідно натиснути правою кнопкою миші по межі структури варіанту і в контекстному меню вибрати опції Add Case After (Створити варіант після) або Add Case Before (Створити варіант перед) поточним варіантом. Аналогічно можна видалити будь-який з варіантів.

За замовчуванням можна задати варіант (Default), який буде виконуватися, якщо величина, що подається на термінал селектора структури, не відповідає ніякому іншому варіанту.

Структура послідовності (Sequence Structure) виконує вбудовані в неї блок-схеми послідовно в певному порядку. Кількість вбудованих блок-схем визначається числом фреймів даної структури. Їх кількість додається за допомогою контекстного меню – Add Frame After, а Add Frame Before.

Для передачі значень змінних з фрейма (рамки) в фрейм використовуються локальні змінні структури (контекстне меню – Add Sequence Local variable), створювані на границі фрейма. Дані, пов'язані з такою змінною, доступні у всіх наступних фреймах і не доступні в попередніх.

Хід роботи

1. Запустіть LabVIEW і створіть новий ВП, File >> New VI.
2. Розмістіть на передній панелі (Window >> Show Front Panel) такі елементи: Numeric Control і Numeric Indicator, які знаходяться в палітрі Controls >> Numeric.
3. Перейдіть у вікно блок-діаграми (Window >> Show Block Diagram) та розмістіть на ньому Case Structure (Functions >> Programming >> Structures).

Для виклику палітри Functions клацніть правою кнопкою миші по будь-якій області блок-діаграми.

4. Додайте до схеми об'єкти: Greater or Equal? (Functions >> Programming >> Comparison), та Square Root (Functions >> Programming >> Numeric).

5. Додатково розмістіть об'єкт One Button Dialog (діалогове вікно), який розташований в палітрі Functions >> Programming >> Dialog & User Interface.

6. З'єднайте елементи між собою відповідно до рис. 14.

String Constant (рядкову константу) можна знайти в палітрі Functions >> Programming >> String.

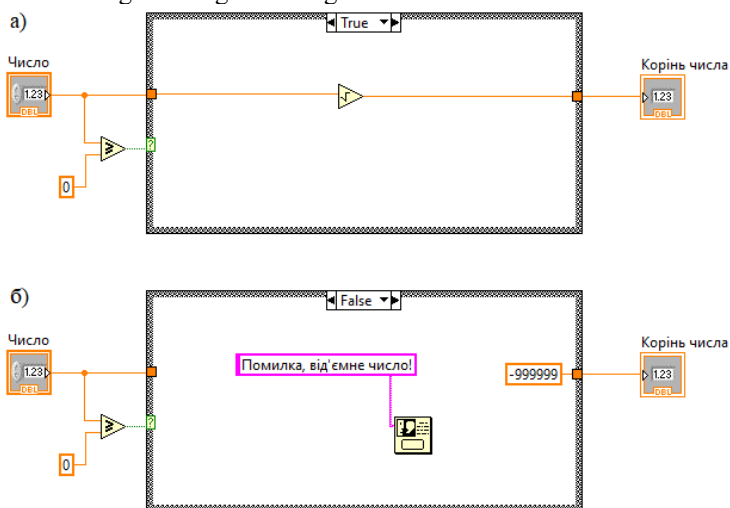


Рис. 14. Блок-діаграма ВП Case Structure: а – варіант True, б – варіант False

7. Перейдіть на передню панель і запустіть ВП (Run). Введіть спочатку додатне, а потім від'ємне число і перевірте роботу пристрою. Збережіть отриманий результат.

8. Розглянемо структуру послідовність (Sequence Structure). створіть новий ВП, File >> New VI.

9. Перейдіть у вікно передньої панелі та розмістіть на ній такі елементи:

- Push Button (Controls >> Express >> Buttons);
- Round LED в кількості 3 шт. (Controls >> Express >> LEDs).

10. Перейдіть у вікно блок-діаграми та розмістіть на ній такі елементи:

- Case Structure (Functions >> Programming >> Structures >> Case Structure);
- Flat Sequence (Functions >> Programming >> Structures >> Flat Sequence);

- Wait в кількості 3 шт. (Functions >> Programming >> Timing >> Wait).

11. Для кожного з LED створіть логічну константу True (права кнопка миші на LED >> Create >> Constant).

Логічну константу можна перевести в значення True або False за допомоги курсора Operate Value (вказівний палець) меню Tools.

12. Для кожного з LED додатково створіть локальну змінну (права кнопка миші на LED >> Create >> Local Variable). Для кожної Local Variable створіть логічну константу (False).

13. Збільшити кількість послідовних структур до трьох (права кнопка миші на Flat Sequence >> Add Frame After або Add Frame Before).

14. З'єднайте елементи між собою відповідно до рис. 15.

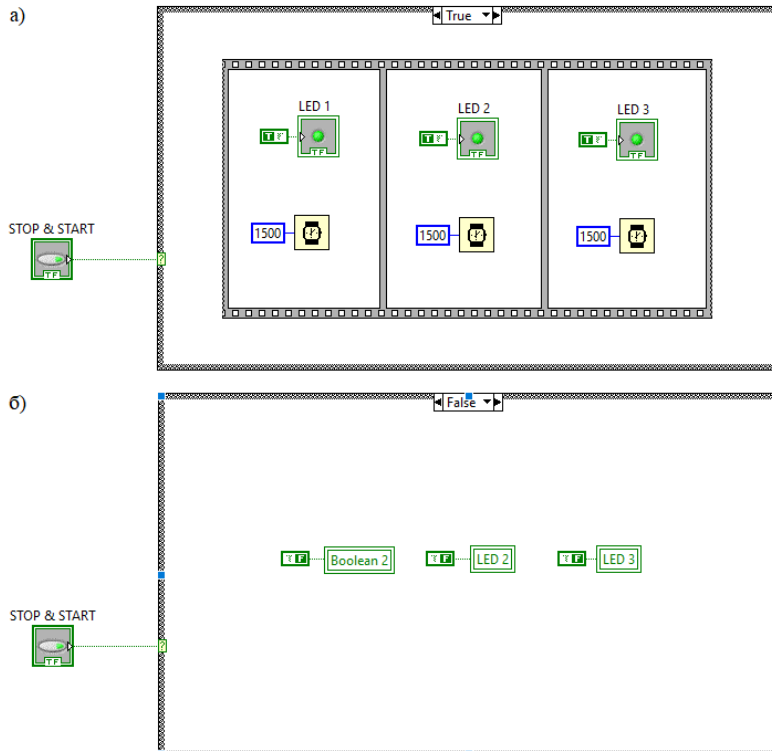


Рис. 15. Блок-діаграма ВП Sequence Structure в поєднанні зі Case Structure:
а – варіант True, б – варіант False

15. Перейдіть на передню панель і переведіть Push Button в режим Switched Until Released (правою кнопкою миші на Push Button >> Mechanical Action >> Switched Until Released).

16. Запустіть ВП в циклічному режимі (Run Continuously) та натисніть Push Button. Збережіть отриманий результат.

17. Контрольне завдання: створіть віртуальний прилад у відповідності до вашого варіанту завдання.

Табл. № 4

№	Завдання
1	Використовуючи структуру Case, створіть віртуальний пристрій, який визначає значення виразу $y = Ax + 14$, де $A - \text{const}$; і якщо $y \geq 0$, то обчислюється значення кореня, інакше видається повідомлення про помилку.
2	Створіть віртуальний пристрій, який за допомогою Formula Node визначає значення виразу $y = \sin(x)$, якщо y – додатне число, то $\text{res} = y + A$, інакше $\text{res} = y - B$, де $A, B - \text{const}$.
3	Створіть віртуальний пристрій, який здійснює почергове включення індикаторів на передній панелі; проміжки між включеннями індикаторів 5, 8 і 12 с, відповідно. Використовуйте послідовність Sequence Structure і функцію Time Delay або Wait, розташовану в палітрі Functions >> Programming >> Timing.
4	Використовуючи структуру Case, створіть віртуальний пристрій, який визначає значення виразу $y = (x + A*20)/2$, де $A - \text{const}$; і якщо $y \geq 0$, то обчислюється значення кореня, інакше видається повідомлення про помилку.
5	Створіть віртуальний пристрій, який за допомогою Formula Node визначає значення виразу $y = \cos(x)$, якщо y – додатне число, то $\text{res} = y * A$, інакше $\text{res} = y / B$, де $A, B - \text{const}$.
6	Створіть віртуальний пристрій, який здійснює почергове включення індикаторів на передній панелі; проміжки між включеннями індикаторів 3, 6, 12 і 15 с, відповідно. Використовуйте послідовність Sequence Structure і функцію Time Delay або Wait, розташовану в палітрі Functions >> Programming >> Timing.
7	Використовуючи структуру Case, створіть віртуальний пристрій, який визначає значення виразу $y = (A*x * 2) + (B*x/2)$, де $A, B - \text{const}$; і якщо $y \geq 0$, то обчислюється значення кореня, інакше видається повідомлення про помилку.
8	Створіть віртуальний пристрій, який за допомогою Formula Node визначає значення виразу $y = \tan(x)$, якщо y – додатне число, то $\text{res} = 2*(y - A)$, інакше $\text{res} = 2*(y + B)$, де $A, B - \text{const}$.
9	Створіть віртуальний пристрій, який здійснює почергове включення індикаторів на передній панелі; проміжки між

	включеннями індикаторів 4, 8 і 10 с, відповідно. Використовуйте послідовність Sequence Structure і функцію Time Delay або Wait, розташовану в палітрі Functions >> Programming >>Timing.
10	Використовуючи структуру Case, створіть віртуальний пристрій, який визначає значення виразу $y = (A+5*x)*(B-5*x)$, де A, B – const; і якщо $y \geq 0$, то обчислюється значення кореня, інакше видається повідомлення про помилку.

Контрольні питання

1. Для чого служить структура Case в LabVIEW?
2. Для чого служить структура Sequence в LabVIEW?
3. Чи можна поєднувати структури Case та Sequence? Якщо так, наведіть приклади.
4. Яким чином можна збільшити число вбудованих блок-схем в структурі Sequence?
5. Для чого служать так звані локальні змінні об'єктів (Local variable)?

Лабораторна робота № 4

«Робота з масивами в LabVIEW»

Мета роботи: ознайомлення з масивами та способами роботи з ними в LabVIEW.

Теоретичні відомості

Масив (array) LabVIEW являє собою набір елементів скалярного типу (скалярна величина являє собою тип даних, які містять єдине значення) так само, як і в традиційних мовах програмування. Масив може мати одну або кілька розмірностей, тобто бути одновимірним або багатовимірним, і включати до 231 елементів на одну розмірність (в залежності від обсягу пам'яті). Елементом масиву може бути будь-який тип даних за винятком масиву, таблиці або графіка.

Доступ до елементів масиву здійснюється за допомогою їх індексів. Індекс кожного елемента знаходиться в діапазоні від 0 до N-1, де N – загальна кількість елементів в масиві. Перший елемент масиву має індекс 0, другий – 1 і т.д. Осцилограми часто зберігаються у вигляді масивів і кожна точка осцилограми містить елемент масиву. Також масиви використовуються для зберігання даних, згенерованих в циклах, де на кожній ітерації циклу створюється один елемент масиву.

Для створення елементів управління і відображення масивів необхідно об'єднати шаблон масиву (array shell) з об'єктом даних, який може бути числовим, логічним, або рядковим. Шаблон масиву знаходиться Controls >> Modern >> Data Containers >> Array. Для створення масиву перемістіть об'єкт даних у вікно відображення елементів.

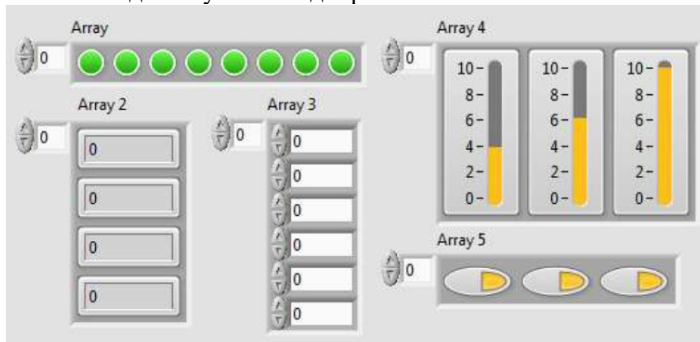


Рис. 16. Масиви в LabVIEW

Вікно відображення елемента зміниться в розмірах, показуючи цим самим, що відбудеться узгодження типу даних. Всі елементи масиву повинні бути одного типу, тобто або елементами управління, або індикаторами, але не їх комбінацією (рис. 16).

Хід роботи

1. Запустіть LabVIEW і створіть новий ВП, File >> New VI.
 2. На передній панелі (Window >> Show Front Panel) створіть 1D масив (Controls >> Data Containers >> Array) та помістіть в нього, один об'єкт типу Numeric Control (Controls >> Numeric). За допомогою курсора змініть розмір області масиву по вертикалі до 10 елементів.
 3. На блок діаграмі (Window >> Show Block Diagram) натисніть праву кнопку миші на масиві, і з контекстного меню виберіть опцію Change to Indicator.
 4. Зберіть схему, яка зображена на рис. 17.
- Елемент Random Number (0-1) розташований на палітрі Functions >> Programming >> Numeric.

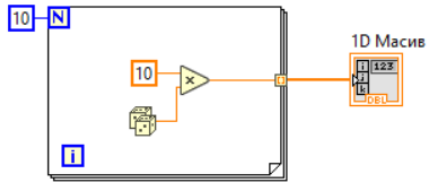


Рис. 17. Блок-діаграма створення 1D масиву

5. Перейдіть на передню панель та запустіть ВП. Кожна ітерація циклу відповідає одному елементу масиву. Проаналізуйте одержаний результат.
6. Зупиніть ВП та змініть блок-діаграму, як показано на рис. 18.

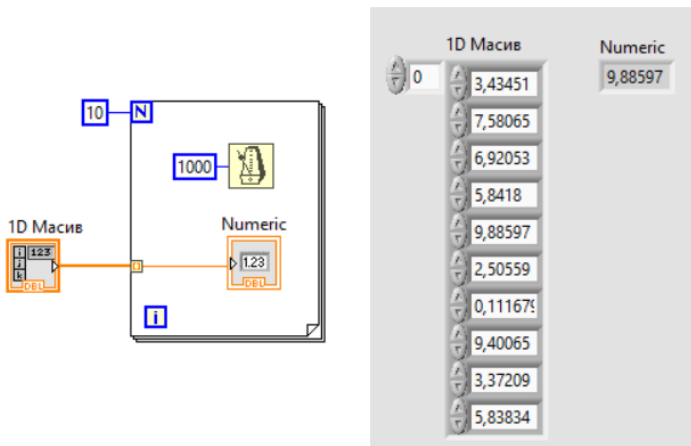


Рис. 18

7. На блок-діаграмі клацніть правою кнопкою миші по вашому масиву і в контекстному меню виберіть опцію Change to Control, тобто змініть тип масиву, тепер ВП не створює масив, а виводить в Numeric Control на кожній ітерації циклу по одному елементу масиву.

8. Перейдіть на передню панель. Запустіть ВП і проаналізуйте його роботу. Збережіть отриманий результат.
9. Створіть новий ВП, File >> New VI.
10. Створіть двовимірний (2D) масив, для цього вставте на передню панель масив (Controls >> Data Containers >> Array).
11. За допомоги курсора, шляхом додавання до масиву 2-х об'єктів типу Numeric Control та збільшення розміру його області, змініть 1D масив на 2D.
12. Перейдіть на блок-діаграму та зберіть схему, як зображено на рис. 19. Обов'язково після з'єднання всіх провідників, задайте тип масиву (Change to Indicator).

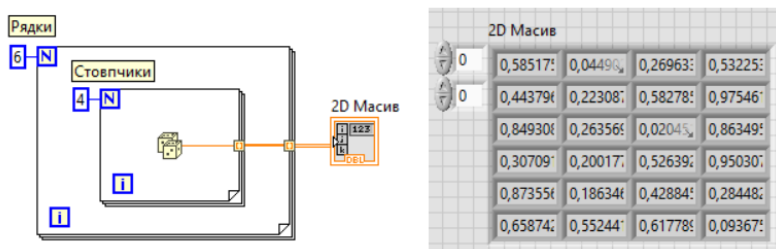


Рис. 19. Блок-діаграма створення 2D масиву

13. Запустіть ВП та проаналізуйте його роботу. Збережіть одержаний результат.
14. Створіть новий ВП, File >> New VI.
15. Помістіть на передню панель Waveform Graph для відображення значень масивів, який розташований на палітрі Controls >> Graph.
16. Додайте ще три елементи Plot до Waveform Graph, тому що на графіку буде відображатися відразу чотири масиви.
17. Для цього наведіть курсор миші на область Plot і при появі синіх точок, що позначають межі елемента, потягніть вгору, наче збільшуючи розмір елемента.

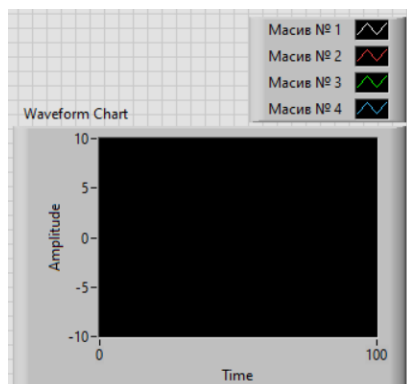


Рис. 20

18. Потім правою кнопкою миші клацніть по Plot і в контекстному меню натисніть на Properties і на вкладці Plots дайте імена всім графікам, як це показано на рис. 20.

19. Створіть чотири 1D масиви, як це зображено на рис. 21.

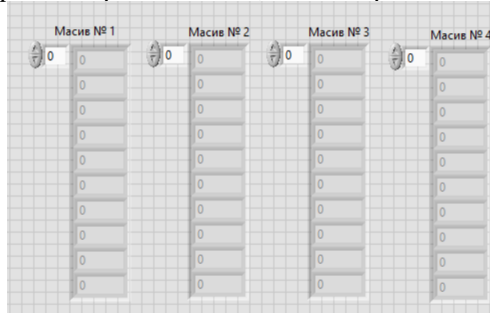


Рис. 21

20. Перейдіть у вікно блок-діаграми і побудуйте структурну схему відповідно до рис. 22.

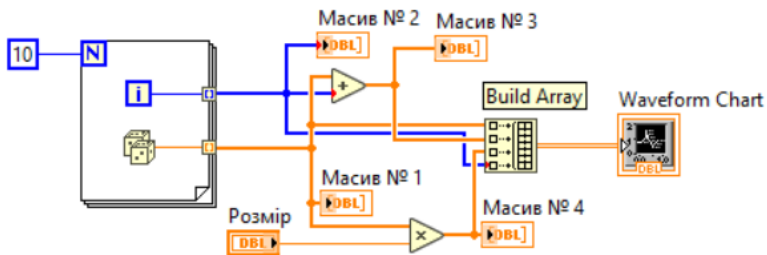


Рис. 22

Об'єкт Build Array можна знайти в палітрі Functions >> Programming >> Array.

21. Запустіть ВП і проаналізуйте його роботу (рис. 23). Збережіть одержаний результат.

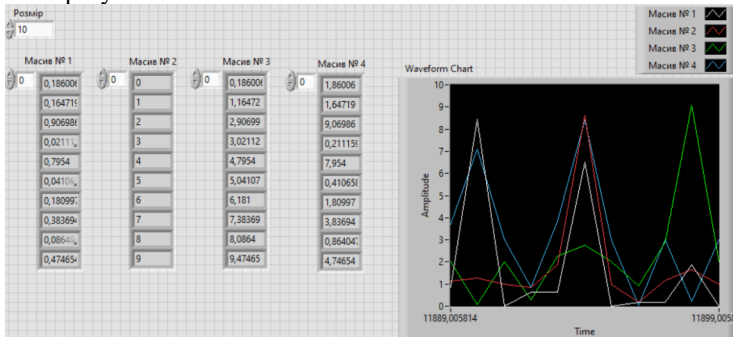


Рис. 23

22. Контрольне завдання: створіть віртуальний прилад у відповідності до вашого варіанту завдання.

Табл. № 5

№	Завдання
1	Створіть віртуальний пристрій, який генерує двовимірний масив випадкових чисел, що містить 3 рядки і 10 стовпців.
2	Створіть віртуальний пристрій, який генерує одновимірний масив, який попарно перемножує елементи, починаючи з елементів з індексами 0 і 1 і т. д., а потім виводить результати в масив елементів відображення даних. Наприклад, вхідний масив має значення [1, 23, 10, 5, 7, 11], а в результаті виходить масив [23, 50, 77]. Використовуйте функцію Decimate 1D Array, розташовану в палітрі Functions >> Array.
3	Створіть віртуальний пристрій, який генерує одновимірний масив випадкових чисел до тих пір, поки не буде натиснута кнопка на передній панелі. На передню панель вивести отриманий масив і його розмірність.
4	Створіть віртуальний пристрій, який генерує одновимірний масив випадкових чисел і виводить мінімальне значення отриманого масиву і його порядковий номер. Використовуйте функцію Max & Min, яка розташована в палітрі Functions >> Array.
5	Створіть віртуальний пристрій, який генерує два одновимірних масиви випадкових чисел і об'єднує ці масиви в двовимірний масив чисел. На передню панель вивести два вихідних масиви випадкових чисел і двовимірний масив, що складається з елементів вихідних масивів.
6	Створіть віртуальний пристрій, який генерує одновимірний масив випадкових чисел і сортує отриманий масив в порядку зростання. На передній панель вивести масив випадкових чисел і відсортований масив. Для сортування елементів в масиві слід використовувати функцію Sort 1D Array, розташовану на палітрі Functions >> Array.
7	Створіть віртуальний пристрій, який генерує двовимірний масив випадкових чисел і здійснює транспонування отриманого масиву. На передню панель вивести масив випадкових чисел і транспонований масив. Для транспонування масиву використовуйте функцію Transpose 2D Array, розташовану на палітрі Functions >> Array.

8	Створіть віртуальний пристрій, який генерує одновимірний масив випадкових чисел і виводить максимальне значення отриманого масиву і його порядковий номер. Використовуйте функцію Max & Min, розташовану на панелі Functions >> Array.
9	Створіть віртуальний пристрій, який генерує одновимірний масив, що містить 30 випадкових чисел, і видає частина масиву, починаючи з індексу 10 до індексу 30. На передню панель вивести масив випадкових чисел і отриманий масив. Використовуйте функцію Array Subset, розташовану на панелі Functions >> Array.
10	Створіть віртуальний пристрій, який генерує одновимірний масив випадкових чисел і сортує елементи масиву із заданого наперед. На передній панель вивести масив випадкових чисел і зворотній масив. Для сортування елементів масиву слід використовувати функцію Reverse 1D Array, розташовану на панелі Functions >> Array.

Контрольні питання

1. Що таке масив?
2. Яким типом даних може бути масив?
3. Які типи даних не припустимі при створенні масиву?
4. Як створити одновимірний (1D) масив?
5. Як створити двовимірний (2D) масив?
6. Як здійснюється доступ до елементів масиву?
7. Для чого служить елемент Random Number (0-1)?
8. Для чого служить елемент Build Array?

Лабораторна робота № 5

«Рядки та рядкові елементи управління в середовищі LabVIEW»

Мета роботи: ознайомлення з можливостями текстового представлення даних в середовищі LabVIEW.

Теоретичні відомості

Рядкові елементи керування (String Control) та індикатори (String Indicator) демонструють текстові дані. Рядки часто містять дані у форматі ASCII (стандартний спосіб зберігання алфавітно-цифрових символів). Рядкові термінали і провідники, по яких проходять рядкові дані, пофарбовані в рожевий колір. Термінали містять літери «abc». Рядкові елементи розташовані в підпалітрі String & Path палітри Controls.

Рядкові елементи керування та індикатори можуть містити числові символи, але не повинні містити числових даних. Над рядковими даними можна здійснювати будь-які числові операції. Якщо необхідно використовувати числову інформацію, що міститься в рядку (для здійснення, наприклад, арифметичних дій), необхідно її перетворити в числовий формат за допомогою необхідних функцій.

Хід роботи

1. Запустіть LabVIEW і створіть новий ВП, File >> New VI.
2. На передню панель (Window >> Show Front Panel) помістіть елемент Listbox, розташований на палітрі Controls >> List, Table & Tree. Даний елемент буде виконувати функцію меню вибору дій над рядками.
3. Введіть в Listbox наступні пункти (Tools >> Edit Text):
 - Об'єднання рядків;
 - Перетворення числа в рядок;
 - Введення пароля.
4. Перейдіть у вікно блок-діаграми (Window >> Show Block Diagram). Підключіть термінал Listbox до конектора структури вибору (Case Structure). Конектор змінить колір з зеленого на синій. Клацніть правою кнопкою миші по вікну вибору (Selector Label) і в контекстному меню виберіть Add Case After, тобто збільште число варіантів до трьох (кількість варіантів в Listbox).
5. Виберіть у вікні Case Structure варіант «0, Default» і побудуйте блок-діаграму відповідно до рис. 24.
 - 5.1. Елементи Concatenate Strings (об'єднання рядків) та String Length (довжина рядка) розташовані Functions >> Programming >> String.
 - 5.2. Для створення елементів String Control (Частина 1-4) клацніть правою кнопкою миші по входу елемента Concatenate Strings і в

контекстному меню виберіть Create Control або помістіть дані елементи на передню панель з палітри Controls >> String & Path >> String Control.

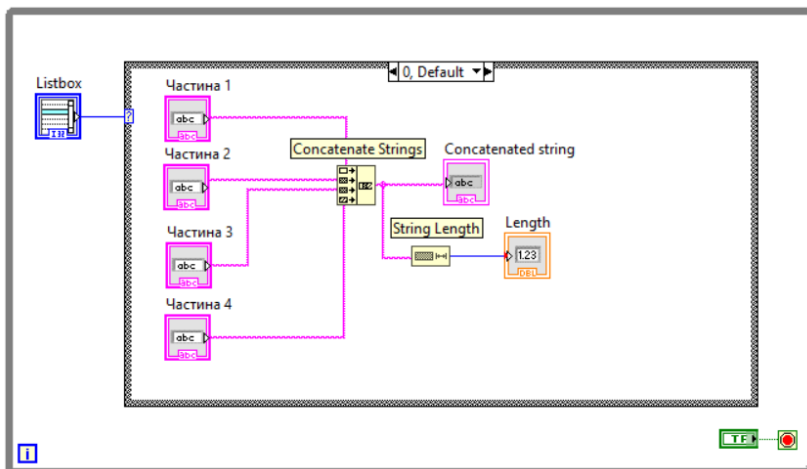


Рис. 24.

6. Виберіть у вікні Case Structure варіант «1» і побудуйте блок-діаграму відповідно до рис. 25.

6.1. Елементи Format Into String (Перетворення в рядок) і Space Constant (пропуск) розташовані Functions >> Programming >> String.

6.2. Формат «%.4f» означає, що числове значення перетвориться в рядкове значення з чотирма знаками після коми.

Для створення Error in та Error out об'єктів, клацніть правою кнопкою миші на відповідному елементі об'єкта Format Into String («!») і в контекстному меню виберіть елемент Create Control.

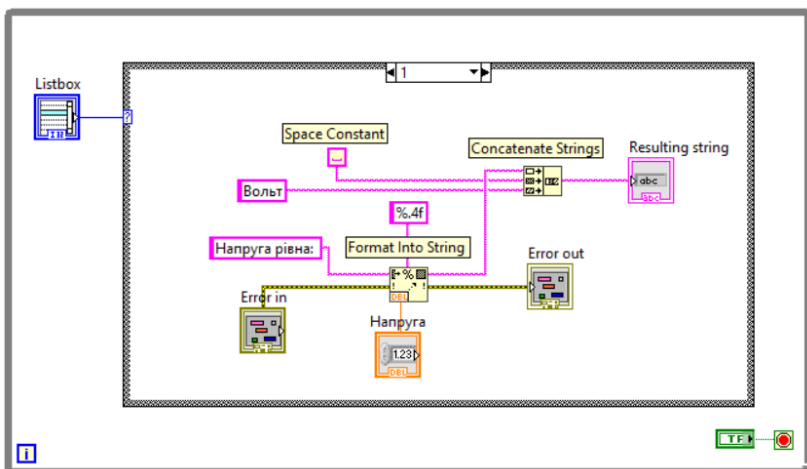


Рис. 25

7. Виберіть у вікні Case Structure варіант «2» і побудуйте блок-діаграму відповідно до рис. 26.

7.1. Елемент Match Pattern (шаблон рядка) розташований Functions >> Programming >> String.

7.2. Функція Empty String/Path? (порожній рядок/шлях?) розташована Functions >> Programming >> Comparison. Ця функція повертає логічне значення True (істина), якщо знаходить порожній рядок в шаблоні підрядка виведення в функції Match Pattern.

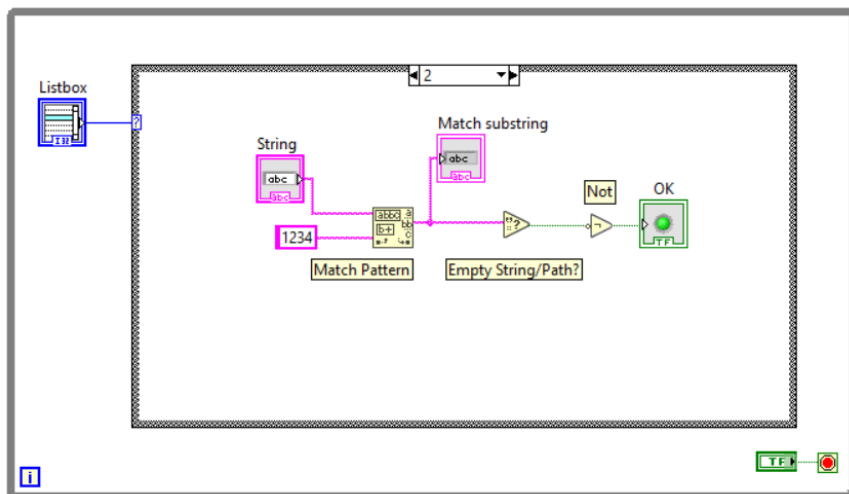


Рис. 26

8. Передня панель ВП представлена на рис. 27. Запустіть ВП і проаналізуйте його роботу. Збережіть отриманий результат.

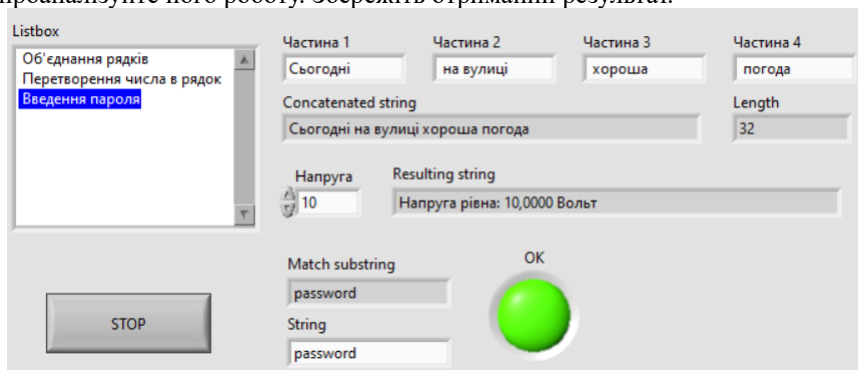


Рис. 27

Контрольні питання

1. Для чого потрібні рядкові елементи керування та індикатори?
2. Чи можна проводити арифметичні операції над числами в рядкових елементах?
3. Навіщо потрібен елемент Listbox?
4. Для чого призначений елемент Concatenate Strings?
5. Для чого потрібен елемент String Length?
6. Як перетворити число в рядок?
7. Для чого призначений елемент Match Pattern?

Список рекомендованой литературы

1. Кузьменко М. Ю., Егоров О. П., Шибакінський В. І. Методичні вказівки до виконання лабораторних робіт з дисципліни «Теорія автоматичного управління» Частина І. Дніпро: НМетАУ, 2018. 74 с.
2. Шагиахметов Д. Р., Абдулина. З. В. Основы моделирования приборов в LabVIEW. Методические указания по выполнению лабораторных работ для студентов специальности 5В071600 – Приборостроение. Алматы: АУЭС, 2013. 31 с.
3. Трэвис Дж., Кринг Дж. LabVIEW для всех. 4-е издание, переработанное и дополненное. Москва : ДМК Пресс, 2011. 904 с.
4. Евдокимов, Ю. К. Labview для радиоинженера: от виртуальной модели до реального прибора: учебное пособие для вузов. Москва : ДМК Пресс, 2007. 400 с
5. Лупов С. Ю., Муякшин С. И., Шарков В. В. LabVIEW в примерах и задачах. Учебно-методические материалы по программе повышения квалификации «Обучение технологиям National Instruments». Нижний Новгород : НГУ им. Н.И. Лобачевского, 2007. 101 с.
6. Дивин А. Г., Жилкин В. М., Свириденко А. Д. Автоматизация измерений, контроля и испытаний. В 2 ч. : лабораторные работы. Ч. 1 : Основы работы в программной среде LabVIEW. Тамбов : Изд-во Тамб. гос. техн. унта, 2005. 44 с.