

Лабораторна робота № 3. Впорядкування послідовності виконання окремих блоків програми та використання масивів.

Мета роботи: Ознайомлення студентів з методами упорядкування послідовності виконання операцій в програмі за допомогою використання “Sequence”-структур. Набуття навичок формування окремих масивів даних та їх обробки.

1. Впорядкування послідовності виконання окремих блоків програми.

При створенні VI в середовищі LabVIEW, послідовність виконання двох незалежних блоків програми не визначена, що в певних випадках призводить до неоднозначних результатів. Впорядкувати порядок виконання окремих блоків програми можна за допомогою структури “Sequence”. В якості приклада використання даної структури, розглянемо програму, що буде розраховувати час виконання певного циклу.

Для реалізації поставленої задачі, необхідно провести наступні дії:



Після запуску нового VI, у вікні редагування діаграм формуємо структуру “Stacked Sequence Structure” (рис. 3.1, а). Для роботи даної структури необхідно задати кількість необхідних нам кадрів. Підводимо вказівник миші на границю структури, натискаємо праву кнопку миші та у вікні, що з’явилося, обираємо пункт “Add Frame After” (рис. 3.1, б). Для вирішення поставленої задачі нам вистачить 3 кадри (0...2).

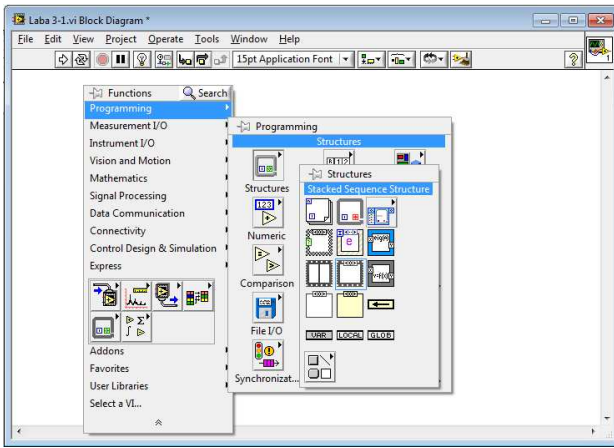


Рис. 3.1,а – Вибір структури “Stacked Sequence Structure”

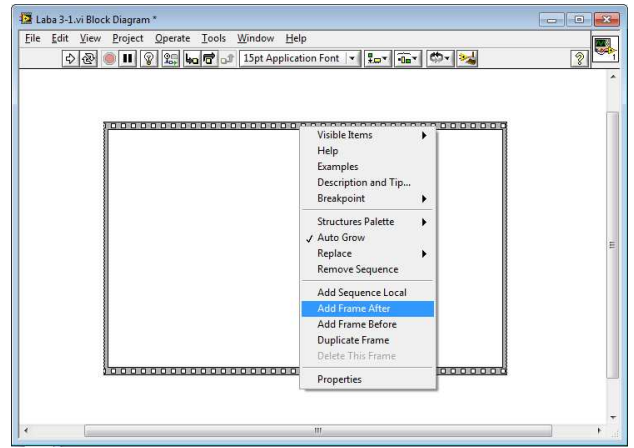


Рис. 3.1,б – Додавання кадрів в структуру “Stacked Sequence Structure”

Наступною задачею буде створення окремої блок-діаграми для кожного кадру, в яких будуть використані окремо проініціалізовані локальні змінні. Ініціалізація локальних змінних в кадрах реалізується аналогічно з заданням кількості кадрів структури. Після появи меню, обираємо вкладку “Add Sequence Local” (рис. 3.2, а). В першому кадрі структури помістимо компонент “Tick Count” (рис. 3.2, б), що дозволяє зчитувати поточне значення системного часу та повертає результат в мілісекундах (рис. 3.3, а). В другому кадрі реалізовано безпосередньо алгоритм програми, що генерує послідовність випадкових чисел та відображає їх на графіку (рис. 3.3, б). В третьому кадрі зчитуються поточні значення системного часу та порівнюються з початковими (рис. 3.4, а). В цьому ж кадрі отриманий результат ділиться на 1000 для переведення розрахованого часу до секунд. Результат обчислень виводиться на зовнішню панель програми (рис. 3.4, б)

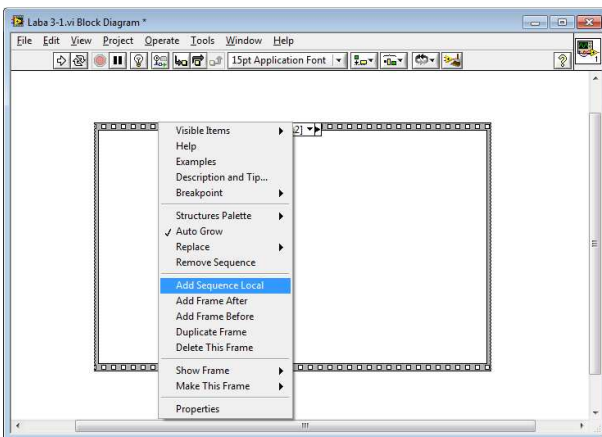


Рис. 3.2, а – Додавання локальної змінної в структуру “Stacked Sequence Structure”

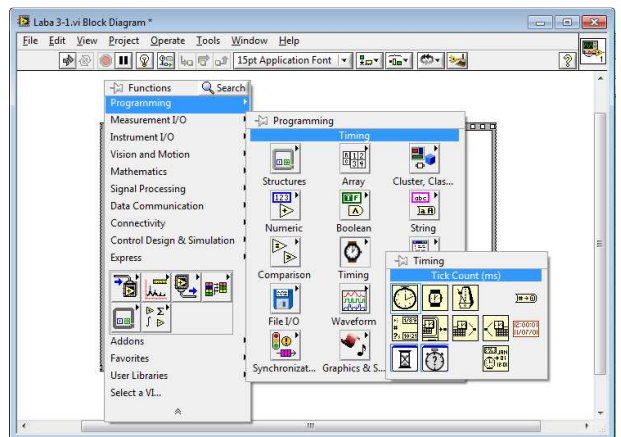


Рис. 3.2, б – Додавання “Tick Count” для зчитування поточного значення системного часу

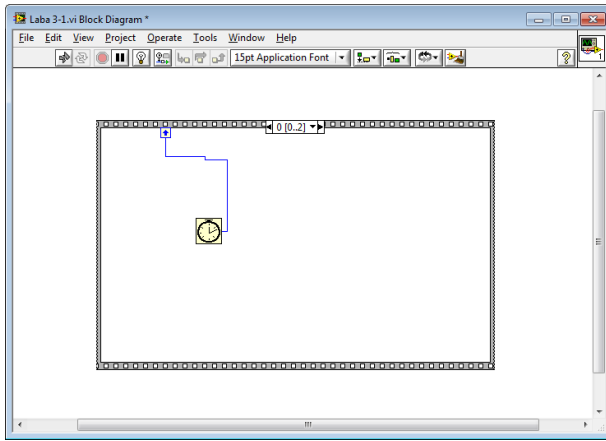


Рис. 3.3, а – Перший кадр структури “Stacked Sequence Structure”

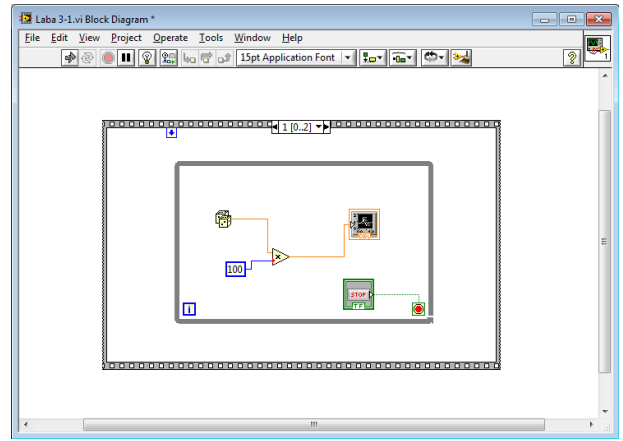


Рис. 3.3, б – Другий кадр структури “Stacked Sequence Structure”

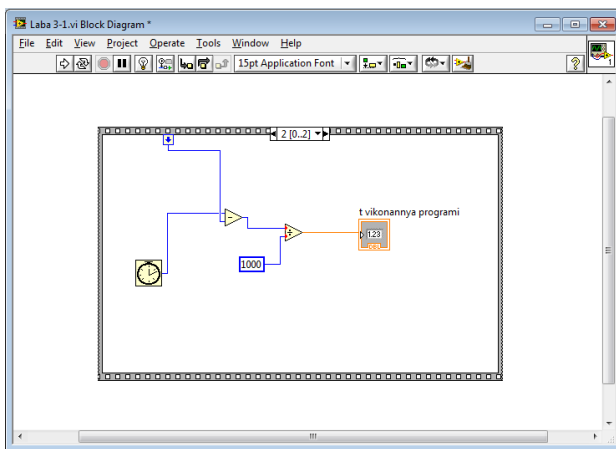


Рис. 3.4, а – Третій кадр структури “Stacked Sequence Structure”

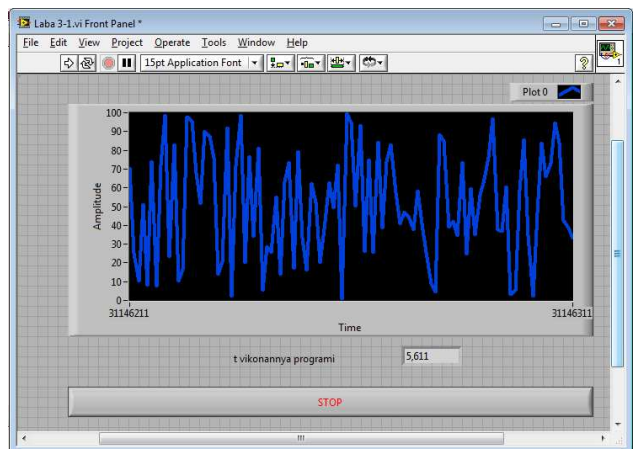


Рис. 3.4, б – Зовнішній вигляд програми для розрахунку часу виконання циклу

2. Робота з масивами даних в середовищі LabView.

Одним з важливих елементів при автоматизації та програмуванні задач збору даних та керування є масиви – набір однотипних компонентів, доступ до яких реалізується по індексу. Створення масиву як елемента керування чи індикації в середовищі LabView реалізується комбінуванням оболонки масива та об’єкту даних, що може бути цифровим, булевим (логічним: наприклад, «так» - «ні»), строковим чи комбінованим.

Для створення нового масиву оберіть пусту оболонку масиву (“Array”) та перемістіть її на інтерфейсну панель (рис. 3.5, а). Після чого оберіть елемент керування або індикації, який ви будете використовувати в даному масиві та помістіть його в створену оболонку (рис. 3.5, б). Регулювання кількості рядків масиву забезпечується розтяганням вікна сформованого масиву (рис. 3.6, а). При цьому елементи керування/індикації та їх властивості будуть автоматично дублюватись, але номер елемента буде змінюватись в залежності від його розташування в масиві. Важливо пам’ятати, що нумерація строк та рядків масиву починається з “0”. Тобто, при зверненні

до першої клітинки масива, її адреса буде наступною: “0, 0”. Для додавання додаткових строк в масив, натисніть правою кнопкою миші на панелі вибору строки масиву, оберіть опцію “Add Dimention” стільки раз, скільки строк вам необхідно у масиві (рис. 3.6, б) та розтягніть масив на вказану кількість рядків.

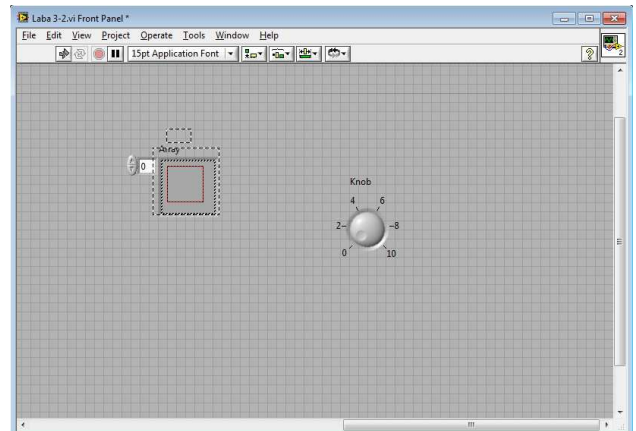
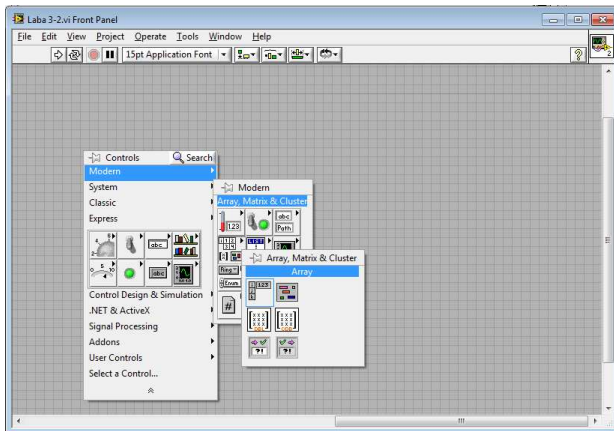


Рис. 3.5, а – Створення оболонки масиву

Рис. 3.5, б – Формування масиву з елементів керування

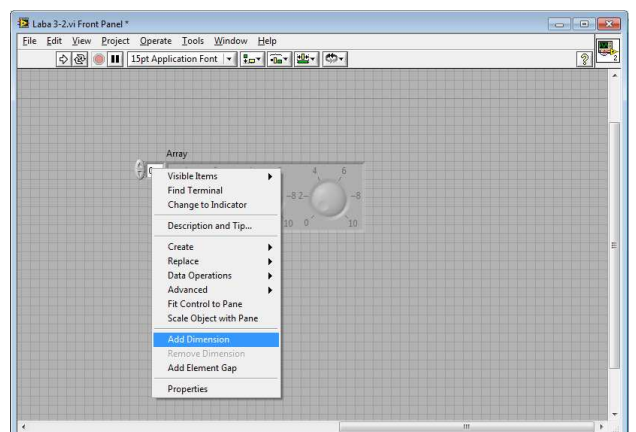
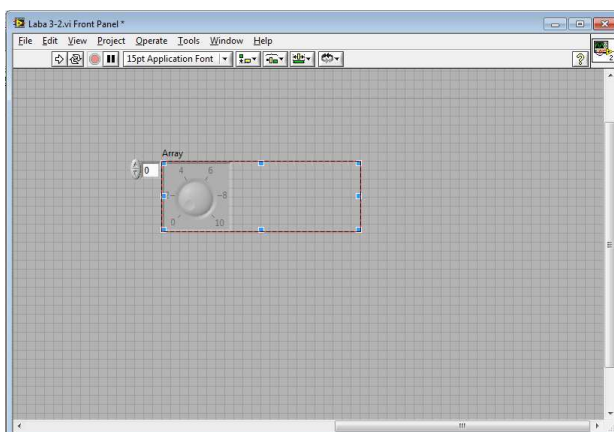

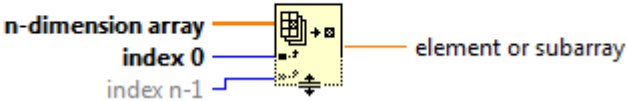
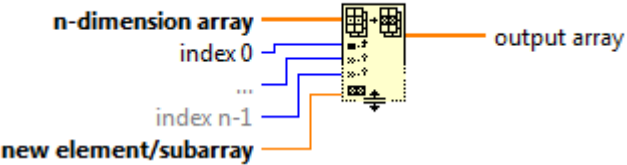
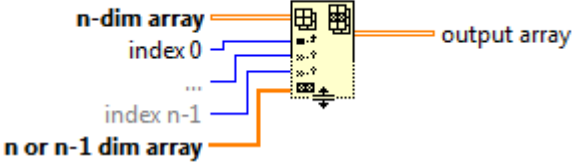
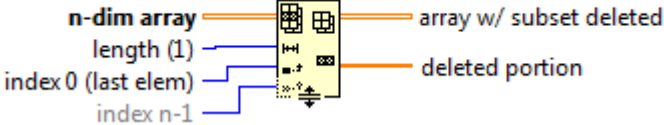
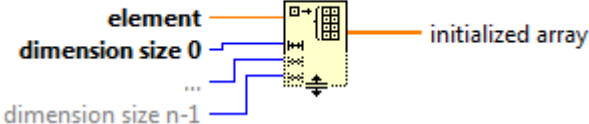
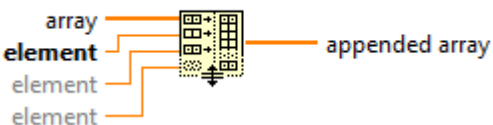
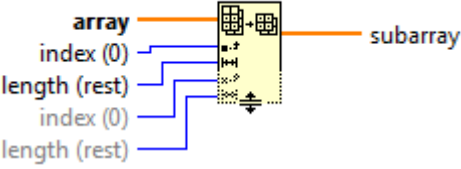
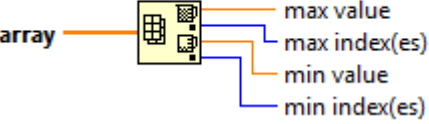
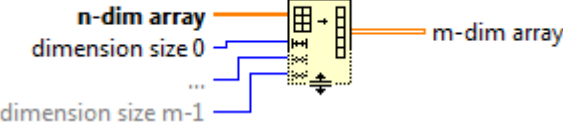
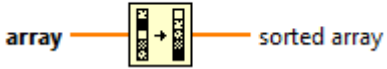


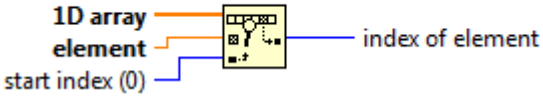


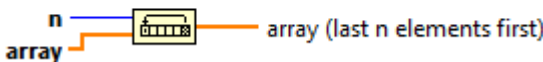
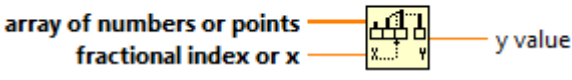
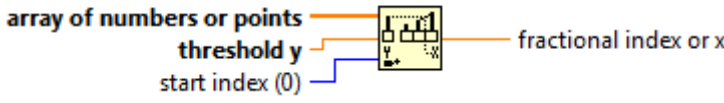
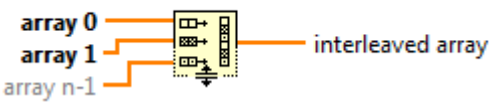
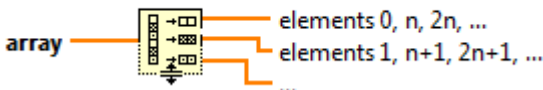
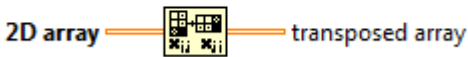



Рис. 3.6, а – Додавання рядків у масив

Рис. 3.6, б – Додавання строк у масив

Середовище LabView пропонує велику кількість інструментів для роботи з масивами: зміна їх розмірів, заміна значень заданих елементів, транспонування даних, їх упорядкування і т.п. Всі ці функції для роботи з масивами виведено на панелі інструментів “Programming-Array”, опис даних функцій приведено в таблиці 3.1.

| | |
|---|---|
| <p style="text-align: center;">Array Size</p>  | <p>Повертає значення кількості елементів в кожному напрямі масиву</p> |
| <p style="text-align: center;">Index Array</p>  | <p>Повертає значення елемента масиву за його індексом</p> |

| | |
|--|--|
| <p style="text-align: center;">Replace Array Subset</p>  <p>The diagram shows a block with an input 'n-dimension array' on the left and an output 'output array' on the right. Three input lines labeled 'index 0', '...', and 'index n-1' point to a specific region within the array icon. A fourth input line labeled 'new element/subarray' points to the same region.</p> | <p>Змінює значення елементу масиву за його індексом на нове.</p> |
| <p style="text-align: center;">Insert Into Array</p>  <p>The diagram shows a block with an input 'n-dim array' on the left and an output 'output array' on the right. Three input lines labeled 'index 0', '...', and 'index n-1' point to a region. A fourth input line labeled 'n or n-1 dim array' points to the same region.</p> | <p>Вводить в масив додатковий елемент у відповідності до його індексу</p> |
| <p style="text-align: center;">Delete From Array</p>  <p>The diagram shows a block with an input 'n-dim array' on the left and two outputs on the right: 'array w/ subset deleted' and 'deleted portion'. Three input lines labeled 'length (1)', 'index 0 (last elem)', and 'index n-1' point to a region.</p> | <p>Видаляє з масиву елемент у відповідності до вказаного індексу</p> |
| <p style="text-align: center;">Initialize Array</p>  <p>The diagram shows a block with an input 'element' on the left and an output 'initialized array' on the right. Three input lines labeled 'dimension size 0', '...', and 'dimension size n-1' point to the array icon.</p> | <p>Створює масив розміром n, в якому елементи приймають вказане значення</p> |
| <p style="text-align: center;">Build Array</p>  <p>The diagram shows a block with an input 'array' on the left and an output 'appended array' on the right. Two input lines labeled 'element' and 'element' point to the array icon.</p> | <p>Створює масив розміру n з n елементів або масивів</p> |
| <p style="text-align: center;">Array Subset</p>  <p>The diagram shows a block with an input 'array' on the left and an output 'subarray' on the right. Four input lines labeled 'index (0)', 'length (rest)', 'index (0)', and 'length (rest)' point to the array icon.</p> | <p>Створює масив певної довжини (розміру), що починається з елемента n.</p> |
| <p style="text-align: center;">Array Max & Min</p>  <p>The diagram shows a block with an input 'array' on the left and four outputs on the right: 'max value', 'max index(es)', 'min value', and 'min index(es)'.</p> | <p>Повертає мінімальне та максимальне значення масиву</p> |
| <p style="text-align: center;">Reshape Array</p>  <p>The diagram shows a block with an input 'n-dim array' on the left and an output 'm-dim array' on the right. Three input lines labeled 'dimension size 0', '...', and 'dimension size m-1' point to the array icon.</p> | <p>Змінює розмірність масиву</p> |
| <p style="text-align: center;">Sort 1D Array</p>  <p>The diagram shows a block with an input 'array' on the left and an output 'sorted array' on the right. The array icon shows a plus sign and a sort symbol.</p> | <p>Повертає значення одномірного масиву, в якому елементи розташовані в порядку зростання їх значень</p> |

| | |
|--|---|
| <p style="text-align: center;">Search 1D Array</p>  | <p>Повертає значення індексу елемента, що приймає значення, яке шукається</p> |
| <p style="text-align: center;">Split 1D Array</p>  | <p>Розділяє масив на два – до вказаного індексу елемента та після нього</p> |
| <p style="text-align: center;">Reverse 1D Array</p>  | <p>«Перевертає» масив даних (відповідно, останнє значення масиву стане першим).</p> |
| <p style="text-align: center;">Rotate 1D Array</p>  | <p>«Перевертає» масив даних починаючи зі вказаного елемента.</p> |
| <p style="text-align: center;">Interpolate 1D Array</p>  | <p>Знаходить інтерпольоване значення з масиву даних у відповідності до вказаного індексу елемента у десятковому вигляді</p> |
| <p style="text-align: center;">Threshold 1D Array</p>  | <p>Знаходить значення індексу елемента у десятковому вигляді з масиву даних у відповідності до вказаного значення у.</p> |
| <p style="text-align: center;">Interleave 1D Arrays</p>  | <p>Формування масиву з n - масивів</p> |
| <p style="text-align: center;">Decimate 1D Array</p>  | <p>Формування n – масивів з одного масиву</p> |
| <p style="text-align: center;">Transpose 2D Array</p>  | <p>Проводить операцію «транспонування» масиву</p> |
| <p style="text-align: center;">Array Constant</p>  | <p>Формує в блок-діаграмі масив з константних значень</p> |
| <p style="text-align: center;">Array To Cluster</p>  | <p>Перетворює масив даних у кластер</p> |
| <p style="text-align: center;">Cluster To Array</p>  | <p>Перетворює кластер в масив даних</p> |



| | |
|--|----------------------------------|
| Array To Matrix.vi  | Перетворює масив даних в матрицю |
| Matrix To Array.vi  | Перетворює матрицю у масив даних |

Табл. 3. Опис основних функції панелі інструментів “Programming-Array”

В якості прикладу, розгляньте програму, в якій згенеровані випадкові значення формуються у вигляді масиву даних розмірністю 3x3 (рис. 3.7 а, б). В даній програмі реалізовано функцію упорядкування одномірного масиву “Sort 1-D Array”, зміна розміру масиву “Reshape Array” та заміна елемента масиву “Replace Array Subset”. В результаті, які б значення регулятора елемента “1-2” не вводились, відповідний індикатор буде показувати “7”.

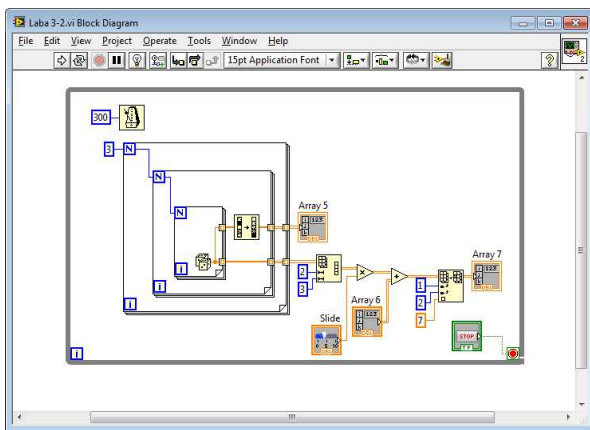


Рис. 3.7, а – Блок-схема програми для генерації та відображення масивів даних

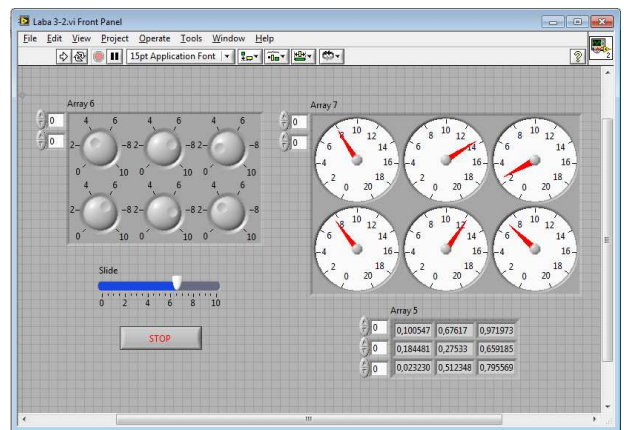


Рис. 3.7, б – Зовнішній вигляд програми для генерації та відображення масивів даних

Індивідуальні завдання:

1. Розробити програму типу «еквалайзер», що забезпечить регулювання користувачем значень, що приймають відповідні елементи індикації.
2. Розробити програму на основі завдання з лабораторної роботи №1. При цьому результати роботи програми додатково мають відображатись у вигляді масиву даних (кількість елементів масиву, що буде відображатись - не менше 5). По завершенню роботи програми, користувачу має бути запропоновано записати отримані дані на жорсткий диск. Окремий індикатор має відображати значення елемента масиву у відповідності до варіанту. По завершенню виконання програми, в окремому індикаторі має виводитись значення часу, на протязі якого використовувалась програма у секундах.