

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ

КАФЕДРА АВТОМАТИЗАЦІЇ ВИРОБНИЧИХ ПРОЦЕСІВ

Методичні вказівки
з виконання самостійної роботи

з курсу: «Мехатронні системи» для студентів спеціальності 151
«Автоматизація та комп'ютерно-інтегровані технології»

м. Кропивницький 2021

**Міністерство освіти і науки України
Центральноукраїнський національний технічний
університет**

**Факультет автоматики і енергетики
Кафедра автоматизації виробничих процесів**

**Методичні вказівки
з виконання самостійної роботи**

з курсу: «Мехатронні системи» для студентів спеціальності 151
«Автоматизація та комп'ютерно-інтегровані технології»

Затверджена
на засіданні кафедри АВП
протокол № 5 від 01.11.2021 р.

м. Кропивницький 2021

Методичні вказівки з виконання самостійної роботи для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / Укл. : В.А. Зозуля.- Кропивницький: ЦНТУ,2021. – 53с.

Укладач: В.А. Зозуля – доцент кафедри АВП.

Рецензент: С.І. Осадчий– проф., д.т.н.

Центральноукраїнський національний технічний університет, 2021р.

ЗМІСТ

	Стор.
ВСТУП	2
Заняття 1. ВИВЧЕННЯ ОСНОВНИХ ПОНЯТЬ ПРОГРАМНОГО СЕРЕДОВИЩА LabVIEW І ВІРТУАЛЬНОГО ПРИЛАДУ	4
Заняття 2. СТВОРЕННЯ, РЕДАГУВАННЯ ТА ОТЛАДКА ВІРТУАЛЬНОГО ПРИЛАДУ	12
Заняття 3. СТВОРЕННЯ ПІДПРОГРАМ ВІРТУАЛЬНОГО ПРИЛАДУ	17
Заняття 4. БАГАТОРАЗОВІ ПОВТОРЕННЯ І ЦИКЛИ ПРИ СТВОРЕННІ ВІРТУАЛЬНОГО ПРИЛАДУ (ВП) В SEREDOVISHCHІ LabVIEW	25
Заняття 5. Робота з масивами В SEREDOVISHCHІ LabVIEW	31
Заняття 6. СТВОРЕННЯ КЛАСТЕРІВ З ЕЛЕМЕНТІВ УПРАВЛІННЯ ТА ВІДОБРАЖЕННЯ ДАНИХ. РОБОТА З КЛАСТЕРАМИ.	38
Заняття 7. ВИВЧЕННЯ СТРОКОВИХ РЕГУЛЯТОРІВ ТА ІНДИКАТОРІВ.	44
СПИСОК ЛІТЕРАТУРИ	53

ВСТУП

В даний час до підготовки інженерів пред'являються високі вимоги в області автоматизації вимірювань, контролю та випробувань. Одним з обов'язкових елементів освоєння даної предметної області є придбання навичок практичного використання систем автоматичного проектування різних комп'ютерних контрольно-вимірювальних систем. Мова **LabVIEW (Laboratory Virtual Instrument Engineering Workbench)** розроблений фірмою **NATIONAL INSTRUMENT** і призначений для розробки програмно - апаратних систем збору даних, обробки даних і управління різними об'єктами і процесами. LabVIEW - представляє собою мову графічного програмування, на відміну від інших мов використовуються текстові мови. Фірмою випускаються різноманітні інтерфейсні пристрої, що вбудовуються в комп'ютер або підключаються до його портів, пристрою генерації та обробки реальних електричних сигналів, датчики, які реєструють різні фізичні процеси, і т.п.

Мова LabVIEW не схожий на інші мови програмування. З її допомогою створюється не програми, як ми звикли її уявляти, а віртуальний інструмент, призначений не тільки для моделювання тих чи інших процесів, але й для керування апаратними засобами і дослідження реальних фізичних об'єктів. Простота образних графічних конструкцій, наочність і читаність готових програм змушує віддати перевагу мові LabVIEW перед іншими. Він подібний до таких систем, як C ++ або Basic, проте на відміну від них оперує не кодовими рядками, а блоками діаграм, що робить програмування більш простим і зрозумілим.

Переваги цієї мови виявляються при виконанні наступних завдань:

- дослідження можливості контролю вимірювальною системою параметрів об'єкта вимірювань;
- аналіз та оптимізація структури комп'ютерних контрольно-вимірювальних систем;
- розрахунок та аналіз статистичних та динамічних метрологічних характеристик (МХ) комп'ютерних контрольно-вимірювальних систем та її компонентів.

Побудовані з використанням даного пакета імітаційні моделі наочно відображають суть реальних фізичних процесів, що протікають в об'єктах дослідження. LabVIEW включає великий вибір бібліотек програм, що дозволяють:

- управляти доступом до зовнішніх даних за допомогою стандартних інтерфейсів (КОП, VXI, RS232, USB, тощо);
- аналізувати дані вимірювань;

- здійснювати архівацію і зберігання даних.

Завданням цих занять є освоєння студентами основ побудови моделей засобів вимірювань, проведення імітаційного моделювання процесів вимірювання та контролю, а також прищеплення студентів навичок роботи з об'єктно-орієнтованим програмним забезпеченням.

У ході проведення занять студенти набувають практичний досвід автоматизації процесу вимірювань, навик роботи з віртуальними приладами, вміння формування з окремих засобів вимірювань просторово-розподіленої комп'ютерно контрольовано-вимірювальних систем на базі різних стандартних інтерфейсів.

Складність виконання завдань за варіантами зростає від заняття до заняття. Лабораторний практикум побудований таким чином, щоб по його закінченні учні мали достатній рівень знань і навичок роботи з інтерфейсом програмного середовища LabVIEW.

ЗАНЯТТЯ 1

ВИВЧЕННЯ ОСНОВНИХ ПОНЯТЬ ПРОГРАМНОГО СЕРЕДОВИЩА LabVIEW I ВІРТУАЛЬНОГО ПРИЛАДУ (ВП)

- МЕТА РОБОТИ** :
- розглянути побудову ВП;
 - розглянути палітру Елементів (**Controls Palette**);
 - розглянути палітру **Tools Palette**;
 - розглянути палітру **Functions**;
 - набути практичних навичок створення ВП.

Теоретичний матеріал

Програма, написана в середовищі LabVIEW, називається **віртуальним приладом**(ВП).

ВП симулюють реальні фізичні прилади. LabVIEW містить повний набір інструментів для збору, аналізу, представлення і зберігання даних.

Запуск середовища програмування LabVIEW здійснюється або подвійним кліком миші на ярлику LabVIEW, який знаходиться на робочому столі, або з розділу Пуск – Програми -National Instruments LabVIEW. При вході в головне меню LabVIEW користувачеві пропонується створення нового віртуального інструменту (**New VI**) або відкриття вже існуючого (**Open VI**).

ВП складається з чотирьох основних компонентів - *лицьової панелі, блок-діаграми, іконки і сполучної панелі*.

Розробка VI (ВП) здійснюється на двох панелях, що знаходяться в двох вікнах, - передній (лицьова панель) і функціональній (блок-діаграма). Передня панель - інтерфейс користувача створюється з використанням **палітри Елементів (Controls Palette)**. Ці елементи можуть бути або засобами введення даних - *елементи управління*, або засобами відображення даних - *елементи відображення*. *Елементи управління* - кнопки, перемикачі, повзунки і інші елементи введення. *Елементи відображення* - графіки, цифрові табло, світлодіоди і так далі

Після цього на блок-діаграмі ВП здійснюється програмування з використанням **палітри Функцій (Functions Palette)**, яка включає графічне представлення функцій для управління об'єктами на передній панелі.



Рис. 1. Керуючі кнопки панелей LabVIEW.

Структура панелей однакова. Основним елементом кожної панелі є робоча область, забезпечена горизонтальним і вертикальним скролінгами, в якій і розміщуються елементи. Також на панелях є верхнє меню і набір керуючих кнопок (рис.1):

- кнопка "**стрілка**" - пуск виконання програми; якщо в програмі є помилки, то ця кнопка розколота на дві частини;
- кнопка "**Стрілки в циклі**" - запуск програми в циклічному режимі;
- кнопка "**Червоний кружечок**" - зупинка виконання програми;
- кнопка "**Дві вертикальні риси**" - пауза у виконанні програми.

Для обох панелей доступна палітра **Tools Palette** (рис.2), що включає набір керуючих кнопок для зміни режиму редагування. Перерахуємо деякі з них:

- кнопка "**вказівний палець**" – інструмент «УПРАВЛІННЯ», слугує для зміни позицій вимикачів і кнопок, управління значеннями цифрових регуляторів, налаштування віртуальних осцилографів та ін.;
- кнопка "**стрілка**" – інструмент «ПЕРЕМІЩЕННЯ», слугує для виділення, переміщення об'єктів, зміна їх розміру;
- кнопка "**А**" - відкриття і редагування текстового вікна;
- кнопка "**котушка**" - інструмент «З'ЄДНАННЯ», слугує для з'єднання об'єктів на функціональній панелі;
- кнопка "**пензель**" - розфарбовування об'єктів або фону;
- кнопка "**рука**" - переміщення робочої області панелі у вікні;
- кнопка "**піпетка**" - вибір поточного кольору з наявних на панелі;
- кнопка "**червоний круг**" - для розміщення і зняття точок зупинки виконання програми на функціональній панелі;
- кнопка "**Р**" - для розміщення на функціональній панелі локальних вікон для відображення поточних значень даних, що передаються в ході виконання програми.



Рис.2 Палітра Tools Palette.

При активній передній панелі стає доступною палітра **Controls** (рис. 3), вона викликається або натисненням правої кнопки миші в робочому просторі лицьової панелі, або необхідно вибрати в пункті головного меню **View >> Controls Palette**. З її допомогою здійснюється візуальне розміщення *елементів управління і елементів відображення* на передній панелі ВП. У панелі Controls вони розподілені по окремих групах за деякими ознаками - числові, логічні, строкові, масиви, діалогові, ActivX, Internet та ін



Рис. 3 Палітра Controls

Розглянемо основні категорії палітри **Controls** :

- **Numeric** (числові значення). Складається з елементів управління і елементів відображення для числових даних;
- **Boolean** (Булеві значення). Складається з елементів управління і елементів відображення для булевих величин;
- **String&Table** (строкові значення і таблиці). Складається з елементів управління і елементів відображення для ASCII рядків і таблиць.
- **List & Ring** (списки і закріплені списки). Складається з елементів управління і елементів відображення для меню виконаних у формі списків і закріпчених списків;
- **Array & Cluster** (масиви і кластери). Складається з елементів управління і елементів відображення для угруповання наборів типів даних;
- **Graph** (віртуальні осцилографи). Складається з елементів відображення для побудови графіків даних в графах або діаграмах в реальному масштабі часу;
- **Path & Refnum** (шляхи і посилання). Складається з елементів управління і елементів відображення для шляхів і посилань;
- **Decorations** (оформлення). Складається з елементів управління і елементів відображення графічних об'єктів для налаштування дисплеїв лицьової панелі;

- **Select Control** (вибір регулятора). Відображає діалогове вікно для завантаження саморобних елементів управління;
- **User Controls** (засоби управління користувача). Складається із спеціальних засобів управління, які формує сам користувач;
- **ActiveX** (об'єкти ActiveX). Складається із засобів управління, які дозволяють впровадити об'єкти ActiveX на лицьову панель;
- **Dialog** (діалогова панель). Складається із стандартних об'єктів для формування діалогу з користувачем;
- **IMAQ Vision** (обробка зображень). Складається із засобів обробки і аналізу зображень;
- **Internet Toolkit** (робота з Internet). Складається із засобів управління, що розташовуються на передній панелі що дозволяють організувати роботу віртуальних інструментів в мережі Internet (ftp, електронна пошта, telnet, CGI і інші).

Елементи, які входять до цих категорій розташовані на тематичних закладках, які видно зразу після розгорнення палітри Елементів. Повний перелік цих закладок палітри Елементів: Modern, System, Classic, Express, Control Design & Simulation, NET & ActiveX, Signal Processing, Addons, User Controls, Select a Control. Елементи відображення та управління можуть дублюватися в кількох закладках, тому шлях до них вказується наступним чином. Наприклад: елемент “Termometr”, розташований в категорії Modern > Numeric.

Після розташування елементів управління або відображення даних на лицьову панель вони отримують своє графічне відображення (у вигляді терміналу даних) на блок-діаграмі. Символи на терміналі відповідають типу даних терміналу. Наприклад, **DBL** - термінал представляє дані у вигляді дійсних чисел з подвійною точністю, **TF** - логічний термінал, **I16** - термінал 16-бітових цілих та ін.

При активуванні функціональної панелі стає доступною палітра **Functions** (рис. 4), яка аналогічно панелі **Controls** включає систематизовані набори стандартних елементів у вигляді окремих піктограм з яких здійснюється складання блок-схеми ВП. Палітра Functions викликається або клацанням правої кнопки миші в робочому просторі блок-схеми, або шляхом вибору в пункті головного меню **View >>Function Palette**.

Розглянемо основні категорії компонентів палітри **Functions** :

- **Structures** (структури). Складається із структур програми, що управляють, таких як цикли For Loop, While Loop та ін.;
- **Numeric** (числові функції). Складається з тригонометричних, логарифмічних і інших функцій;

- **Boolean** (Булеві функції). Складається з логічних і булевих функцій;
- **String** (строкові функції). Складається з функцій для роботи із строковими величинами;
- **Array** (масиви). Складається з функцій для обробки масивів;
- **Cluster** (кластери). Складається з функцій для обробки кластерів;
- **Comparison** (порівняння). Складається з функцій для порівняння змінних;
- **Time & Dialog** (час і діалог). Складається з функцій для діалогових вікон, синхронізації і обробки помилок;
- **File I/O** (введення/виведення файлу). Складається з функцій для здійснення операцій по вво-ду/виведенню файлів;
- **Instrument I/O** (інструменти введення/виводу). Складається з ВП для зв'язку і управління приладами різної архітектури;
- **Instrument Drivers** (драйвери приладів). Складається з ВП, здатних управляти зовнішніми приладами. осцилоскопами, генераторами і так далі, через послідовний порт або інтерфейс GPIB;
- **Data Acquisition** (збір даних). Складається з ВП для використання плат збору даних;
- **Signal Processing** (обробка сигналів). Складається з ВП для генерації і обробки сигналів;
- **Mathematics** (математичні). Складається з оптимізаційних, алгебраїчних, інтегральних, диференціальних і інших функцій;
- **Graphics & Sound** (графіка і звук). Складається з ВП для роботи з Зхвимірною графікою, зображеннями і звуком;
- **Communication** (зв'язки). Складається з віртуальних приладів для роботи з мережами TCP DDE та ін.;
- **Application Control** (управління застосуванням). Складається з ВП, управляючих віртуальними приладами;
- **Advanced** (розширена). Складається з різних функцій типу функції бібліотечного запиту, маніпуляції цих та ін.;
- **Report Generation** (генерація звіту). Складається з ВП використовуваних для підготовки звітних документів;
- **Tutorial** (повчальні програми). Складається з VI, використовуваних в повчальній програмі LabVIEW.
- **User Libraries** (призначені для користувача бібліотеки). За допомогою її організовується швидкий доступ до потрібному ві;

- **Select VI** (вибір ВП). Складається з діалогового вікна для впровадження підпрограм в поточний ВП;
- **IMAQ Vision** (обробка зображень). Складається з ВП використовуваних для обробки і аналізу зображень;

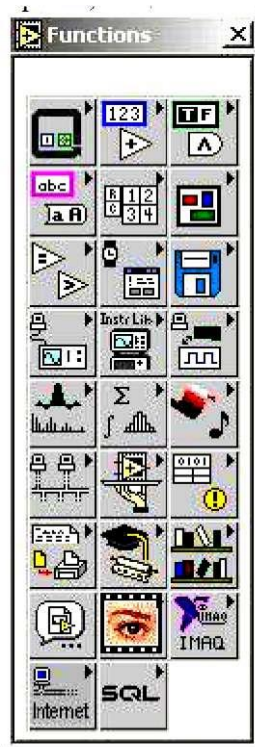


Рис. 4 Палітра Functions

- **Image Acquisition** (отримання зображення). Складається з ВП, використовуваних для отримання і обробки зображень;
- **Internet Toolkit** (робота з Internet). Складається з ВП, використовуваних для роботи в мережі Internet (ftp, електронна пошта, telnet, CGI та ін.);
- **SQL** (SQL запити). Складається з ВП, використовуваних для організації зв'язку з SQL сервером і обробки запитів.

Палітра **Functions** має наступні закладки: Programming, Measurement I/O, Instrument I/O, Vision & Motion, Mathematics, Signal Processing, Data Communication, Connectivity, Control Design & Simulation, Express, Addons, Favorites, User Libraries, Select a VI. Адресація до того чи іншого компонента цієї палітри буде аналогічною до адресації в палітрі елементів. Наприклад: функція Decrement з категорії Mathematics > Numeric.

Об'єкти блок-діаграми включають графічне відображення елементів передньої панелі, операторів, функцій, підпрограм ВП, констант, структур і **провідників даних**, по яким виконується обмін даними між об'єктами блок-діаграми.

Провідники даних між терміналами аналогічні змінним на звичайних мовах. Дані йдуть в тільки одному напрямі, з початкового терміналу на один або більше за термінали адресата. Дроти мають різну товщину і колір. Синій колір відповідає цілим числам помаранчевий - дійсним числам, зелений - логічним, ліловий - строковим даним і так далі.

При натисненні правої кнопки миші на регуляторі/індикаторі (як на передній, так і на функціональній панелі) з'являється **контекстне меню**, за допомогою якого можливо здійснити :


- заміну елементу управління (регулятора) на елемент відображення (індикатор) і навпаки (Change to Control, Change to Indicator);
- швидкий пошук терміналу на функціональній панелі (Find Terminal) і регулятора/індикатора на передній панелі (Find Control, Find Indicator);
- налаштування параметрів регулятора/індикатора (Data Operations);
- заміну на інший регулятор/індикатор (Replace);
- отримання довідки по використовуваній функції (Online Help);
- створення для функцій відповідних констант, індикаторів і регуляторів (Create Constant, Create Indicator, Create Control) та ін.

Завдання 1.1. ВП Частотний аналіз

1 Запустіть LabVIEW Пуск >"Програми" > National Instruments > " LabVIEW 2010" > LabVIEW. З'явиться діалогове вікно LabVIEW.

2 Виберіть "Help" > "Find Examples". На екрані з'явиться діалогове вікно пошуку прикладів ВП, розбитих по категоріях.

3 Перейдіть на закладку «Browse» (Огляд). Відмітьте пункт «Directory Structure». Виберіть Apps, Freqresp.llb, і двічі клацніть на «Frequency Response.VI» («Частотна характеристика»). З'явиться Передня панель ВП «Частотний аналіз» (рис. 5):

4 За допомогою інструменту «УПРАВЛІННЯ»  змініть значення установки амплітуди Amplitude. Змінити значення можна або перемістивши покажчик кнопки в потрібне положення, або використовуючи стрілки зміни значень елементу управління, або ввівши число безпосередньо в дисплей елементу.

Кнопка « Enter» . Якщо число введене безпосередньо в дисплей елементу то необхідно натиснути кнопку Enter, яка з'явиться на інструментальній панелі. Інакше число не буде введено.

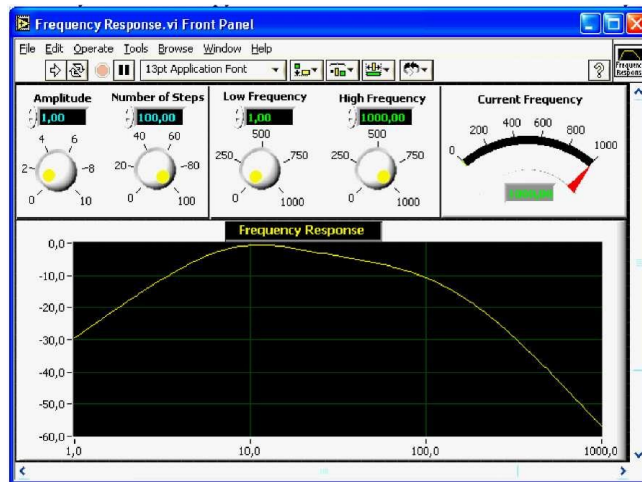



Рис. 5. ВП «Частотний аналіз»


5 Натиснути кнопку Run і запустити ВП. Змінюючи значення інших засобів управління, що знаходяться на панелі, дослідити роботу ВП.

Блок-діаграма


6 Перейдіть на блок-діаграму. Для цього виберіть в головному меню Window > Show Block Diagram, або введіть <Ctrl - E> з клавіатури.

7. Відкрийте вікно контекстної довідки, вибравши в пункті головного меню Help > Show Context Help. Отримати інформацію про об'єкт у вікні контекстної довідки Context Help можна, навівши на них курсор.


а) Розташуйте інструмент «ПЕРЕМІЩЕННЯ»  над функцією Logarithm Base 10, яка розташована під міткою Bode Plot. У вікні контекстної довідки Context Help з'явиться опис функції.

б) У вікні контекстної довідки Context Help натисніть кнопку Detailed Help  для переходу у відповідний розділ LabVIEW Help (Вбудованої Допомоги). Можна також клацнути по посиланню «Detailed Help» вікна контекстної довідки Context Help.

LabVIEW Help (Вбудована Допомога) містить детальний опис палітр, меню, інструментів, ВП і функцій. Тут можна отримати детальний опис і інших функцій.

в) Наведіть інструмент «З'ЄДНАННЯ»  на поля введення/виведення даних функції Logarithm Base 10. Відповідні поля у вікні контекстної довідки Context Help почнуть блимати.

г) Перемістіть інструмент «З'ЄДНАННЯ» на провідник даних. У вікні контекстної довідки Context Help з'явиться опис типу даних в провіднику.

8 Щоб зафіксувати поточне вікно Context Help (контекстної довідки), необхідно або натиснути кнопку , або вибрати в пункті головного меню Help > Lock Context Help.

Коли поточне вікно Context Help (контекстної довідки) зафіксоване, то його вміст не міняється після наведення курсору на інший об'єкт. Для відміни фіксації слід натиснути кнопку другий раз.

Контрольні питання


1. З яких основних компонентів складається ВП?
2. Що розуміється під інтерфейсом користувача ВП?
3. Які палітри доступні для передньої панелі?
4. Які палітри доступні для блок-діаграми?
5. Що таке Передня панель?
6. Яке призначення блок-діаграми?
7. На які категорії поділяються компоненти, які входять до палітри Controls (Елементів)?
8. На які категорії поділяються компоненти, які входять до палітри Functions (Функцій)?
9. На яких панелях здійснюється розробка ВП?
10. Назвіть призначення керуючих кнопок на блок-діаграмі.
11. Назвіть призначення керуючих кнопок на передній панелі.
12. Що таке елемент управління і елемент відображення?
13. Назвіть основні типи даних.
14. Що таке провідник даних?
15. Яким чином здійснюється виклик контекстної довідки?
16. Як можна зафіксувати поточне вікно контекстної довідки?
17. Назвіть призначення контекстного меню.

ЗАНЯТТЯ 2

СТВОРЕННЯ, РЕДАГУВАННЯ ТА ОТЛАДКА ВІРТУАЛЬНОГО ПРИЛАДУ (ВП)

- МЕТА РОБОТИ** :
- розглянути компоненти ВП;
 - створити ВП (перетворювач °C в °F);
 - розглянути типи даних и провідники даних;
 - відредагувати ВП;
 - набути практичних навичок налагодження ВП.

Теоретичний матеріал

 Об'єкти лицьової панелі на блок-діаграмі відображаються у вигляді терміналів даних (графічне зображення прямокутної форми з буквено-чисельними позначеннями). Термінали даних забезпечують обмін даними між лицьовою панеллю і блок-діаграмою; вони подібні змінним і константам текстових мов програмування. Розрізняють термінали цих наступних типів - термінали елементів управління і відображення даних, термінали вузлів.

▷ Вузли - це об'єкти на блок-діаграмі, які мають одне або більше за одне поля введення/виведення даних і виконують алгоритмічні операції ВП. Вони аналогічні операторам, функціям і підпрограмам текстових мов програмування. Вузли включають функції підпрограми ВП і структури. *Підпрограма ВП* - віртуальний прилад, який можна використовувати на блок-діаграмі іншого ВП в якості підпрограми. *Структури* - це елементи управління процесом, такі як структура Case (Варіанти), цикл While (цикл за умовою), і так далі. Вузли Add (Складання) і Subtract (Віднімання), - *вузли функцій*.

Типи і провідники даних. У середовищі LabVIEW провідники даних використовуються для з'єднання численних терміналів даних. Поля введення/виводу мають бути сумісними з типами даних, що передаються їм по провідниках. Наприклад не можна сполучати поле виведення масиву з полем введення даних чисельного типу. Крім того, характер з'єднання має бути коректним. Провідники мають бути приєднані лише до одного джерела даних і, принаймні, до одного поля введення даних. Наприклад не можна сполучати два елементи відображення. Компонентами, що визначають сумісність з'єднання, являються: тип даних елементу управління і/або відображення і тип даних поля введення/виводу.

Типи даних. У цьому курсі використовуються наступні типи даних :

- **Numeric** (чисельний тип);
- **Floating point** - число з плаваючою комою, відображається у вигляді помаранчевих терміналів. Може бути представлено у вигляді single (32 bit), double (64 - bit) або extended (128 - bit) precision (з поодиноким подвійним або розширеною точністю). Число з плаваючою комою може бути комплексним;
- **Integer** - цілочисельний тип, відображається у вигляді блакитних терміналів. Можливі три представлення цілих чисел: 8, 16 і 32 біта. Один біт може використовуватися для знаку числа якщо це число є знаковим цілим;
- **Boolean** - логічний тип, відображається у вигляді зелених терміналів. Логічний тип може набувати тільки два значення: 0 (FALSE) або 1 (TRUE);

- **String** - строковий тип, відображається у вигляді рожевих терміналів. Строковий тип даних містить текст в ASCII форматі;
- **Path** - шлях до файлу, відображається у вигляді терміналів. Шлях до файлу близький строковому типу, проте, LabVIEW форматує його, використовуючи стандартний синтаксис для використовуваної платформи;
- **Array** - масиви включають типи даних, що складають елементи і приймають колір, що їм відповідає;
- **Cluster** - кластери включають різні типи даних. Кластерний тип даних відображається коричневим кольором, якщо усі його елементи чисельні, якщо ж елементи кластера є даними різних типів, він відображається рожевим;
- **Waveform** - сигнальний тип даних є кластером елементів, що містить дані, початкове значення часу і інтервал часу між вимірами;
- **Dynamic** - динамічний тип, відображається у вигляді темно-синіх терміналів. Окрім даних сигналу, динамічний тип містить додаткову інформацію, наприклад, назву сигналу або дату і час його отримання. Більшість експрес - ВП приймають і/або повертають дані динамічного типу. Дані динамічного типу можна направляти до того елементу 01 поразки або поля введення, що приймає дані чисельного, логічного або сигнального типу.

Провідники даних. Дані між об'єктами блок-діаграми передаються по сполучних лініях - провідниках даних. Провідник даних аналогічний змінним в текстових мовах програмування. Кожен провідник даних має єдине джерело даних але може передавати їх багатьом ВП і функціям. Провідники даних розрізняються кольором, стилем і товщиною лінії, залежно від типу даних що передаються.

Автоматичне з'єднання об'єктів провідниками даних. У середовищі LabVIEW об'єкти з'єднуються провідниками даних після їх приміщення на блок-діаграму. У автоматичному режимі середовище LabVIEW підключає ті поля введення/виведення даних, які найбільш сумісні, несумісні поля залишаються несполученими.

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

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

Завдання 2.1. Перетворення °C в °F

Нижче приведена послідовність дій для створення ВП, який перетворюватиме значення температури з градусів Цельсія в температуру за Фаренгейтом.

Передня панель

- 1 Виберіть пункт головного меню File > "New VI", щоб відкрити нову лицьову панель.
- 2 Розташуйте цифровий елемент управління на передній панелі. У полі власної мітки елемента управління надрукуйте "Град °C".
- 3 Перемістіть елемент відображення даних на лицьову панель. Він буде використаний для відображення значень температури в °F. У полі власної мітки елемента управління надрукуйте "Град °F" і клацніть мишею у вільному просторі лицьової панелі або натисніть кнопку Enter.



Рис. 1. Елементи керування та відображення.

На блок-діаграмі LabVIEW створить термінали даних, що відповідають елементам управління і відображення. Термінали даних представляють тип цих відповідних елементів. Наприклад, термінал даних DBL представляє тип числових даних подвійної точності з плаваючою комою.

Увага! Термінали даних, що відповідають елементам управління, мають ширший обвідною контур в порівнянні з терміналами даних, що відповідають елементам відображення.

Блок-діаграма

- 4 Перейдіть на блок-діаграму, вибравши пункти головного меню Window > Show Block Diagram.
- 5 Виберіть функцію Multiply (Множення) з палітри Функцій в розділі Programming - Numeric (Арифметичні функції). Помістіть її на блок-діаграму.
- 6 Виберіть функцію Add (Складання) з палітри Функцій в розділі Programming - Numeric (Арифметичні функції). Помістіть її на блок-діаграму.
- 7 Виберіть числову константу (Numeric Constant) з палітри Функцій в розділі Programming - Numeric (Арифметичні функції). Помістіть дві числові константи на блок-діаграму. Після розміщення числової константи на блок-діаграмі поле введення її значень підсвічує і готово для редагування. Одній константі присвойте значення 1,8, інший 32,0.

- 8 З'єднаєте об'єкти блок-діаграми за допомогою інструменту «З'ЄДНАННЯ».
- 9 Перейдіть на лицьову панель, вибравши в головному меню пункт Window - Show Front Panel.
- 10 Збережіть ВП, він використовуватиметься пізніше.

Запуск ВП

- 11 Введіть число в елемент управління і запустите ВП:
 - а) для введення числа в елемент управління слід використовувати інструмент «УПРАВЛІННЯ» або інструмент «ВВЕДЕННЯ ТЕКСТУ»;
 - б) натисніть кнопку Run, щоб запустити ВП;
 - в) введіть декілька різних значень температури і запустіть ВП.
- 12 Закрийте ВП, вибравши пункт головного меню File > Close.

Завдання 2.2. Створення ВП, згідно з Вашим варіантом

№ вар.	Завдання
1	ВП перетворює значення температури з градусів Цельсія в температуру за шкалою Кельвіна ($^{\circ}\text{K} = ^{\circ}\text{C} + 273$)
2	ВП перетворює значення температури з градусів Цельсія в температуру за шкалою Реомюра ($^{\circ}\text{R} = ^{\circ}\text{C} \cdot 4/5$)
3	ВП перетворює значення температури за шкалою Кельвіна в градуси Цельсія ($^{\circ}\text{C} = ^{\circ}\text{K} - 273$)
4	ВП перетворює значення температури за шкалою Реомюра в градуси Цельсія ($^{\circ}\text{C} = ^{\circ}\text{R} \cdot 5/4$)
5	ВП перетворює значення температури за Фаренгейтом в градуси Цельсія ($^{\circ}\text{F} = 9/5 \cdot ^{\circ}\text{C} + 32$)
6	ВП перетворює значення температури за шкалою Реомюра в температуру за Фаренгейтом ($^{\circ}\text{F} = 9/4 \cdot ^{\circ}\text{R} + 32$)
7	ВП перетворює значення температури з градусів Цельсія в температуру за Реомюром ($^{\circ}\text{R} = ^{\circ}\text{C} \cdot 4/5$)
8	ВП перетворює значення температури за шкалою Кельвіна в температуру за Реомюром ($^{\circ}\text{R} = (^{\circ}\text{K} - 273) \cdot 4/5$)
9	ВП перетворює по закону Ома значення напруги та опору в силу струму ($I = U / R$)
10	ВП перетворює значення динамічної в'язкості ρ в кінематичну в'язкість ν ($\nu = -\mu/\rho$)
11	ВП перетворює значення напруги (мВ) та сили струму (мкА) у вати (Вт) ($P = IU$)
12	ВП перетворює значення напруги (В) та сили струму (А) в потужність (Вт) ($P=IU$)

Контрольні питання

- 1 З яких основних компонентів складається Ваш ВП?
- 2 Які вузли на блок-діаграмі ВП Ви знаєте?
- 3 Як відображаються термінали даних і які функції вони виконують?
- 4 Що таке провідник даних?
- 5 Які провідники даних Ви знаєте?
- 6 Яких типів даних Ви знаєте?
- 7 Як з'єднуються об'єкти провідниками даних?
- 8 На які категорії поділяються компоненти, які входять до палітри Controls (Елементів)?
- 9 На які категорії поділяються компоненти, які входять до палітри Functions (Функцій)?
- 10 Як здійснюється пуск розробленого ВП?
- 11 Назвіть призначення кнопок, що управляють, на блок-діаграмі.
- 12 Назвіть призначення кнопок, що управляють, на передній панелі.
- 13 Що таке елемент управління і елемент відображення?
- 14 Назвіть основні типи даних.
- 15 Як ввести число в елемент управління?

ЗАНЯТТЯ 3

СТВОРЕННЯ ПІДПРОГРАМ ВІРТУАЛЬНОГО ПРИЛАДУ (ВП)


МЕТА РОБОТИ : розглянути підпрограму ВП;
- створити іконку та налаштувати сполучну панель для можливості використання ВП в якості підпрограми ВП

Теоретичний матеріал

Наступний крок після створення блок-діаграми і формування лицьової панелі ВП - створення іконки ВП і налаштування сполучної панелі для використання віртуального приладу в якості *підпрограми ВП*. Підпрограма ВП відповідає підпрограмі в текстових мовах програмування. Використання підпрограм ВП допомагає швидко управляти змінами і відладкою блок-діаграм.

Будь-який VI може бути використаний як підпрограма при створенні в подальшому інших віртуальних інструментів. Для об'єднання декількох функціональних блоків блок-діаграми, що розробляється, в підпрограму досить виділити їх мишкою на діаграмі

утримуючи клавішу Shift, і потім вибрати у верхньому меню пункт Edit - Create SubVI. При цьому вони об'єднуються в нову підпрограму з новим значком (іконкою) на функціональній панелі. Подвійний клік на цьому значку дозволить викликати створену підпрограму настроїти її належним чином і зберегти із заданим ім'ям. У подальшому цей модуль може бути багаторазово використаний в різних VI.

Кожен віртуальний прилад в правому верхньому кутку лицьової панелі і у вікні блок-діаграми відображає іконку . Іконка - графічне представлення приладу. Вона може містити текст, малюнок або і те і інше одночасно. Якщо ВП використовується в якості підпрограми те іконка ідентифікує його на блок-діаграмі іншого ВП.

Для редагування іконки створюваних vi і підпрограм досить клікнути правою кнопкою миші на піктограмі vi в правому верхньому кутку і вибрати пункт Edit Icon ... (рис. 1). За допомогою простих функцій графічного редактора можна створити власний варіант іконки.

Налаштування входів/виходів (терміналів) підпрограм здійснюється таким чином. Необхідно натиснути правою кнопкою миші на піктограмі vi в правому верхньому кутку і вибрати пункт Show Connector. (рис. 1). При цьому піктограма розділиться на декілька прямокутників, загальний набір і вид яких можна редагувати за допомогою спливаючого меню піктограми (добавити/видалити термінал - Add Terminal/Remove Terminal, поворот на 90 градусів - Rotate 90 Degrees, інший вид - Patterns. та ін.). Для того, що б зіставити кожен термінал з певними даними необхідно лівою кнопкою миші клікнути на потрібному терміналі а потім - на тому елементі управління або відображення на передній панелі, якою він відповідатиме. При цьому термінал забарвиться в колір, що відповідає типу даних вказаного елементу управління або відображення. В результаті усі термінали будуть пов'язані з певними вхідними або вихідними даними.

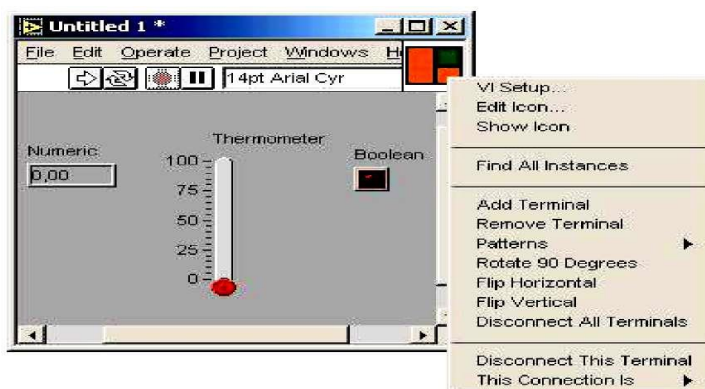


Рис. 1. Редагування іконки та таблиці з'єднань

Використання підпрограм ВП. Після створення ВП, оформлення його іконки і налаштування панелі з'єднань ВП може використовуватися в якості підпрограми. Щоб

помістити підпрограму ВП на блок-діаграму, слід вибрати на палітрі **Functions** (Функцій) підрозділ **Select a VI** (Вибір ВП). Вказати ВП і перенести його на блок-діаграму.

Відкритий ВП можна помістити на блок-діаграму іншого ВП, перемістивши на неї іконку цього ВП за допомогою інструменту «ПЕРЕМІЩЕННЯ».

Завдання 3.1. ВП Перетворення °C в °F

У цій роботі представлена послідовність дій із створення іконки і налаштування сполучної панелі для створеного ВП, який переводить значення вимірної температури з градусів Цельсія в градуси за Фаренгейтом.

Передня панель


1 Відкрийте файл з раніше створеним ВП («Перетворювач °C в °F».vi.)

Іконка і сполучна панель

2 Клацніть правою кнопкою миші по іконці ВП і в контекстному меню виберіть пункт Edit Icon (Редагування іконки). З'явиться діалогове вікно редактора іконки Icon Editor.

3 Двічі клацніть правою кнопкою миші по інструменту «ВИБІР» .

4 Натиснувши кнопку <Delete> очистіть область редагування іконки.

5 Двічі клацніть по інструменту «ПРЯМОКУТНИК»  щоб обвести область редагування рамочкою вибраного кольору.

6 Створіть наступну іконку:

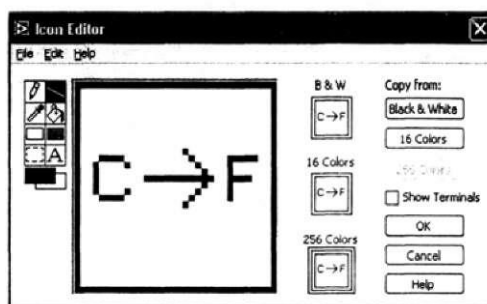




Рис. 2. Створення іконки.

- a) введіть текст інструментом «ВВЕДЕННЯ ТЕКСТУ» .
- б) надрукуйте "C" і "F";
- в) для вибору розміру шрифту двічі клацніть лівою кнопкою миші по інструменту «ВВЕДЕННЯ ТЕКСТУ»;
- г) щоб намалювати стрілку, скористайтеся інструментом «ОЛІВЕЦЬ» .

Увага! Для малювання вертикальних горизонтальних і діагональних ліній потрібно під час малювання натиснути і утримувати клавішу <Shift>.

- д) для пересування тексту і стрілки по полю редагування іконки використовуйте інструмент «ВИБІР» і стрілки на клавіатурі;
- е) в розділі Copy from (Копіювати з) виберіть B & W (чорно-білу) іконку і 256 Colors (256 - кольоровий режим) для створення чорно-білої іконки, яку LabVIEW використовує у разі відсутності кольорового принтера;
- ж) в розділі Copy from (Копіювати з) виберіть 16 Colors і 256 Colors;
- з) після завершення редагування іконки натисніть кнопку ОКІ і закрийте Icon Editor. Нова іконка з'явиться в правому верхньому кутку обох панелей.

7 Перейдіть на лицьову панель клацніть правою кнопкою миші на іконці і виберіть пункт Show Connector (Показати поля введення/виведення даних) з контекстного меню. Кількість відображених LabVIEW полів введення/виведення даних відповідає кількості елементів на передній панелі. Наприклад: Передня панель цього ВП має два елементи Град °C і Град °F і LabVIEW виводить в сполучній панелі два поля.

8 Елементом управління і відображення даних призначте відповідно поля введення і виведення даних :

- а) в пункті головного меню Help (Допомога) виберіть Show Context Help (показати контекстну довідку) і виведіть на екран вікно Context Help (контекстної довідки) для перегляду з'єднань;
- б) клацніть лівою кнопкою мишки на лівому полі сполучної панелі. Інструмент «УПРАВЛІННЯ» автоматично поміняється на інструмент «З'ЄДНАННЯ», а вибране поле забарвиться в чорний колір;
- в) клацніть лівою кнопкою миші по елементу Град °C. Ліве поле стане помаранчевим і виділиться маркером;
- г) клацніть курсором по вільному простору. Маркер зникне і поле забарвиться в колір даних типу відповідного елементу управління;
- д) клацніть лівою кнопкою миші по правому полю сполучної панелі і елементу Град °F. Праве поле стане помаранчевим;
- е) клацніть курсором по вільному простору. Обидва поля залишаться помаранчевими;
- ж) наведіть курсор на область полів введення/виведення даних. Вікно Context Help (контекстної довідки) покаже, що обидва поля відповідають типу даних подвійної точності з плаваючою комою.

9. Виберіть пункт головного меню File-Save і збережіть розроблений ВП під ім'ям *Перетворення С в F. vi*.

10. Виберіть пункт головного меню File - Close. Закрийте ВП.

Завдання 3.2. ВП Термометр

Нижче приведена послідовність дій для створення ВП, який вимірює температуру і відображає значення температури в градусах Цельсія або температуру за Фаренгейтом.

Передня панель

1 Створіть елемент відображення даних температури, як показано нижче (на рис.3.):

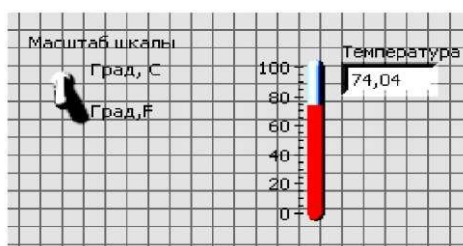


Рис.3. Створення елементів відображення температури.

- а) виберіть елемент відображення даних «термометр», розташований на палітрі Controls в категорії **Modern > Numeric** (Числові елементи) ;
- б) надрукуйте «Температура» усередині власної мітки і натисніть кнопку Enter на інструментальній панелі ;
- в) клацніть правою кнопкою миші по елементу і виберіть пункт контекстного меню **Visible Items** (Елементи, що відображаються), **Digital Display** (Цифровий індикатор) .

2 Створіть елемент управління у вигляді вертикального перемикача.

- а) виберіть вертикальний перемикач, який розташовано в палітрі Controls категорія **Modern > Boolean** (Логічні елементи) ;
- б) введіть ім'я власної мітки перемикача «Масштаб шкали» і натисніть кнопку Enter на інструментальній панелі;
- в) використовуючи інструмент ВВЕДЕННЯ ТЕКСТУ, створіть на передній панелі вільну мітку °C як показано вище;
- г) за допомогою інструменту ВВЕДЕННЯ ТЕКСТУ створіть на передній панелі вільну мітку °F, як показано вище.

3 Створіть опис ВП, який з'являється у вікні контекстної довідки **Context Help** після наведення курсору на іконку ВП:

- а) виберіть пункт головного меню File > VI Properties;
 - б) виберіть пункт Documentation (Опис) в розділі Category (Категорія) із спадаючого меню;
 - в) в полі введення тексту надрукуйте наступне: «Цей ВП вимірює температуру, використовуючи ВП Demo Read Voltage VI».
- 4 Створіть опис елементів управління і відображення даних, яке з'являється у вікні контекстної довідки **Context Help** після наведення на них курсору :
- а) клацніть правою кнопкою миші по елементу відображення і виберіть пункт контекстного меню Description and Tip (Опис і попередження);
 - б) в полі введення тексту надрукуйте наступне:
 - в) «Виводить на екран значення вимірної температури»
 - г) введіть в поле Tip значення Температура;
 - д) натисніть кнопку ОК;
 - е) клацніть правою кнопкою миші по елементу управління і виберіть пункт контекстного меню Description and Tip (Опис і попередження);
 - ж) в полі введення тексту надрукуйте наступне: «Визначає шкалу (за Фаренгейтом або Цельсієм), використовувану для виміру температури»;
 - з) введіть в поле Tip значення шкала - °C або °F і натисніть кнопку ОК.
- 5 Відобразите вікно контекстної довідки **Context Help**, яке доступне з пункту головного меню Help > Show Context Help.
- 6 Наведіть курсор на один з об'єктів для перегляду опису їх роботи у вікні **Context Help**.

Блок-діаграма

- 7 Перейдіть на блок-діаграму, вибравши View > Show Block Diagram.
- 8 Створіть блок-діаграму, показану нижче (на рис.4.).

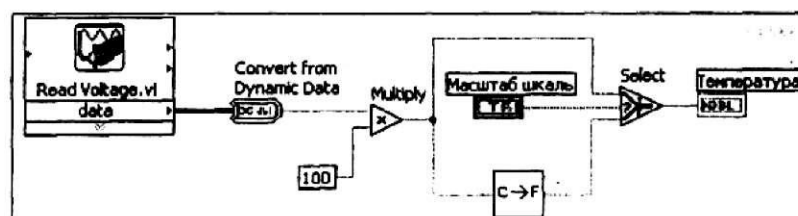






Рис.4. Блок-діаграма.


Розташуйте на блок-діаграмі ВП Demo Read Voltage VI, що знаходиться в каталозі c:\Program Files \National Instruments\LabVIEW 2010\for_LabVIEW, і слугує для імітації зчитування напруги, пропорційне температурі. Наприклад, якщо температура складає 20 °C, то напруга на виході датчика дорівнюватиме 20 В.

 Розташуйте на блок-діаграмі ВП Convert from Dynamic Data (перетворення динамічних даних), який розташовано в палітрі Functions, категорія Express > Signal Manipulation. Цей ВП перетворить динамічний тип даних. У конфігураційному діалоговому вікні виберіть пункт Single Scalar in зі списку Resulting data type.

Виберіть  функцію Multiply (Множення), розташовану в палітрі Functions категорія Programming > Numeric. Ця функція множить прочитану ВП "Read Voltage VI" напругу на "100.0" для представлення температури в градусах Цельсія;

 Клацніть правою кнопкою миші по полю введення даних у функції Multiply (Множення) і в контекстному меню виберіть пункт Create > Constant (Створити константу). Константі присвойте значення "100" і натисніть клавішу <Enter>.

 У палітрі Functions (Функцій) в розділі «Select a VI» (Вибір ВП) виберіть ВП «Перетворення С в F», створений в завданні 3.1. Розташуйте його на блок-діаграмі. Цей ВП переведе градуси Цельсія в градуси Фаренгейта.

Далі  виберіть функцію Select (Вибір) розташовану в палітрі Functions категорія Programming > Comparison. Ця функція видає значення °C або °F залежно від стану перемикача "Масштаб шкали".

Передня панель

- 9 Перейдіть на лицьову панель.
- 10 Натисніть на кнопку безперервного запуску, показану ліворуч.
- 11 Натисніть на кнопку безперервного запуску ще раз для зупинки ВП.

Іконка і сполучна панель

- 12 Створіть іконку, показану нижче, щоб використовувати ВП як підпрограми.

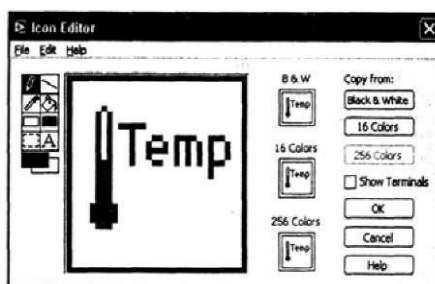


Рис.5. Створення іконки

13 Елементам управління і відображення даних поставте у відповідність поля введення і виведення даних, клацнувши правою кнопкою миші по іконці і вибравши пункт контекстного меню Show Connector (Показати поля введення/виведення даних).

- 14 Збережете ВП під ім'ям «Термометр», він використовуватиметься пізніше.

15 Закрийте ВП, вибравши пункт головного меню File > Close.

Завдання 3.3.

Аналогічно завданню 3.2, створити ВП, використовуючи як підпрограму прилад, створений в завданні 2.2.

№ вар.	Завдання
1	ВП вимірює температуру і відображає значення температури за шкалою Кельвіна або в градусах Цельсія
2	ВП вимірює температуру і відображає значення температури за шкалою Реомюра або в градусах Цельсія
3	ВП вимірює температуру і відображає значення температури в градусах Цельсія або за шкалою Кельвіна
4	ВП вимірює температуру і відображає значення температури в градусах Цельсія або за шкалою Реомюра
5	ВП вимірює температуру і відображає значення температури в градусах Цельсія або за шкалою Фаренгейта
6	ВП вимірює температуру і відображає значення температури за Реомюром або за шкалою Фаренгейта
7	ВП вимірює температуру і відображає значення температури в градусах Цельсія або за шкалою Реомюра
8	ВП вимірює температуру і відображає значення температури за Кельвіном або за шкалою Реомюра
9	ВП вимірює напругу; на виході - значення напруги або сили струму (Обчисленої за законом Ома) в залежності від стану перемикача
10	ВП вимірює в'язкість і відображає значення динамічної в'язкості D або кінематичної в'язкості V
11	ВП вимірює напругу і відображає напругу(мВ) або потужність (Вт)
12	ВП вимірює напругу і відображає напругу(В) або потужність (Вт)

Примітки:

1) для імітації зчитування напруги, пропорційної температурі, у варіантах 1 - 8 слід використовувати ВП Demo Read Voltage VI, розташований в каталозі C:\Program Files \National Instruments\LabVIEW 2010\for_LabVIEW;

2) у варіантах 9, 11, 12 для імітації зчитування напруги - ВП Demo Read Voltage VI;

3) у варіанті 10 для імітації зчитування значень в'язкості - функцію Random Number (0-1), розташовану в палітрі Functions категорія Mathematics > Numeric. Ця функція генеруватиме випадкове число в межах від 0 до 1.

КОНТРОЛЬНІ ПИТАННЯ

- 1 З яких основних компонентів складається Ваш ВП?
- 2 Що називається іконкою ВП?
- 3 Як створюється іконка ВП?
- 4 Як відображаються іконки ВП і які функції вони виконують?
- 5 Як редагується іконка ВП?
- 6 Як вивести на екран значення вимірної температури?
- 7 Навіщо потрібна функцію Select?
- 8 Як здійснюється безперервний пуск розробленого ВП?

ЗАНЯТТЯ 4


БАГАТОРАЗОВІ ПОВТОРЕННЯ І ЦИКЛИ ПРИ СТВОРЕННІ ВІРТУАЛЬНОГО ПРИЛАДУ (ВП) В СЕРЕДОВИЩІ LabVIEW


- МЕТА РОБОТИ :** - вивчення прийомів використання терміналу вихідних даних циклу While (по умові);
- вимір температури разів в секунду впродовж однієї хвилини;
 - використання сдвигових регістрів і вузлів зворотного зв'язку для організації доступу до значень на попередніх ітераціях циклу For (з фіксованим числом ітерацій);
 - вивчення організації доступу до значень попередніх ітерацій циклу.


Теоретичний матеріал

Структури є графічним представленням операторів циклу і операторів Case (Варіанту), використовуваних в текстових мовах програмування. Структури на блок-діаграмі використовуються для виконання операцій, що повторюються, над потоком даних операцій в певному порядку і застосування умов на виконання операцій. Середовище Lab VIEW 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).


Цикл While (по умові). Цикл While (за умовою) працює до того часу, поки не виконається логічна умова виходу з циклу.


 Блок-діаграма циклу While виконується до того часу, доки не виконається умова виходу з циклу. За умовчанням, термінал умови виходу має вигляд, показаний ліворуч. Це означає, що цикл виконуватиметься до вступу на термінал умови виходу значення TRUE. В цьому випадку термінал умови виходу називається терміналом Stop If True (Зупинка якщо Істина). Вона розташована на закладці Programming > Structures палітри Functions.

 Термінал лічильника ітерацій, показаний ліворуч, містить значення кількості виконаних ітерацій. Початкове значення терміналу завжди дорівнює нулю.

 Передбачена можливість зміни умови виходу і зображення терміналу умови виходу, що відповідає йому. Клацанням правої кнопки миші по терміналу умови виходу або по грані-це циклу необхідно викликати контекстне меню і вибрати пункт **Continue If True** (Продовження якщо Істина).


Цикл For (з фіксованим числом ітерацій). Цикл For Loop (з фіксованим числом ітерацій) виконує операції, що повторюються, над потоком даних певна кількість разів.

 Цикл For розташований в палітрі Функцій в розділі Functions >. Programming > Structures. Значення присвоєне терміналу максимального числа ітерацій N циклу, показаного ліворуч, визначає максимальна кількість повторень операцій над потоком даних.


 Термінал лічильника ітерацій, показаний ліворуч, містить значення кількості виконаних ітерацій. Початкове значення лічильника ітерацій завжди дорівнює 0.

Організація доступу до значень попередніх ітерацій циклу. При роботі з циклами потрібний доступ до значень попередніх ітерацій циклу. Наприклад, у разі ВП що вимірює температуру і відображає її на графіці, для відображення поточного середнього значення температури, необхідно використовувати значення, отримані в попередніх ітераціях. Є два шляхи доступу до цих даних: Shift Register (регістр зсуву) і Feedback Node Node (вузол зворотного зв'язку).

Регістри зсуву. Регістри зсуву використовуються при роботі з циклами для передачі значень від поточної ітерації циклу до наступної. Регістри зсуву аналогічні статичним змінним в текстових мовах програмування.

 Регістр зсуву виглядає як пара терміналів, показаних ліворуч. Вони розташовані безпосередньо один проти одного на протилежних вертикальних сторонах межі циклу. Правий термінал містить стрілку "вгору" і зберігає дані після завершення поточної ітерації. LabVIEW передає дані з цього регістра в наступну ітерацію циклу. Регістр зсуву створюється клацанням правої кнопки миші по межі циклу і вибором з контекстного меню пункту Add Shift Register (Додати Регістр зсуву).

Щоб ініціалізувати реєстр зсуву необхідно передати на його лівий термінал будь-яке значення ззовні циклу. Якщо не ініціалізувати Реєстр зсуву, він використовує значення, записане в реєстр під час останнього виконання циклу або значення, використовуване за умовчанням для цього типу даних, якщо цикл ніколи не виконувався.

Вузли зворотного зв'язку.  Вузол зворотного зв'язку, показаний ліворуч, автоматично з'являється в циклах **While** або **For** при з'єднанні поля виведення даних підпрограми ВП, функції або групи підпрограм ВП і функцій з полем введення даних тих же самих підпрограм ВП, функцій або їх груп. Як і Реєстр зсуву, вузол зворотного зв'язку зберігає дані будь-якого типу після завершення поточної ітерації і передає ці значення в наступну ітерацію. Використання вузлів зворотного зв'язку дозволяє уникнути великої кількості провідників даних і з'єднань.

Можна розмістити вузол зворотного зв'язку усередині циклу **While** або **For**, вибравши **Feedback Node (Вузол зворотного зв'язку)** в палітрі **Structures (Структури)**. При приміщенні вузла зворотного зв'язку на провідник даних до відгалуження, передавального дані на вихідний термінал циклу вузол зворотного зв'язку передає усі значення на вихідний термінал циклу. При приміщенні вузла зворотного зв'язку на провідник після відгалуження, передавального дані на вихідний термінал циклу, вузол зворотного зв'язку передасть усі значення назад на поле введення даних ВП або функції а потім передасть останнє значення на вихідний термінал циклу. Наступна вправа містить приклад роботи вузла зворотного зв'язку.

Структура вибору Case. У структурі вибір **Case** є дві або більше вбудованих блок-схеми. Вибір однієї з них, яка буде виконана, визначається залежно від значення, поданого на вхід цієї структури. Структура **Case** включає:

- Термінал вибору. Значення що подається на нього, може бути цілим, логічним або строковим
- Перемикач блок-схем (**True \ False \ і так далі**). Дозволяє переходити від однієї блок-схеми до іншої. Містить за умовчанням два вікна **True** і **False**. При необхідності кількість блок-схем вибору може бути збільшена. Окрім **True** і **False** в якості значень перемикача можуть використовуватися цілі числа або строкові значення.

Формульний блок Formula Node. Формульний блок **Formula Node** дозволяє вводити формули у звичайному вигляді прямо в блок-схему. Особливо це зручно, коли вираз має багато змінних і складний вид. Формули вводяться як простий текст. При цьому створюються термінали на кордоні блоку (контекстне меню **Add Input** або **Add Output**), в які вписуються імена змінних. Кожен вираз закінчується роздільником **";"** ».

Структура послідовності Sequence Structure. 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, Add Frame Before. Для передачі значень змінних з фрейму в кадр використовуються локальні змінні структури (контекстне меню - Add Sequence Local variable), створювані на кордоні фрейму. Дані, пов'язані з такою змінною, доступні у всіх наступних фреймах і не доступні в попередніх.

Завдання 4.1. ВП Вимірювання температури в часі

Нижче наведена послідовність дій для створення СП, який використовує ВП термометр для вимірювання температури раз в секунду протягом однієї хвилини.

Передня панель

1 Відкрийте новий ВП і створіть лицьову панель, як показано нижче на рисунку 1.

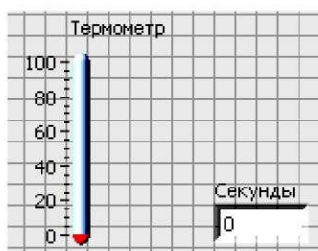


Рис.1. Передня панель

Розташуйте Термометр, розташований на палітрі Controls > Modern > Numeric, на блок-діаграму для відображення на екрані вимірювань температури.

Розташуйте на передній панелі цифрової елемент відображення даних, розташований на панелі Controls > Modern > Numeric. Назвіть його “Секунды”.

Блок-діаграма

2 Створіть блок-діаграму, показану нижче (рис. 2).

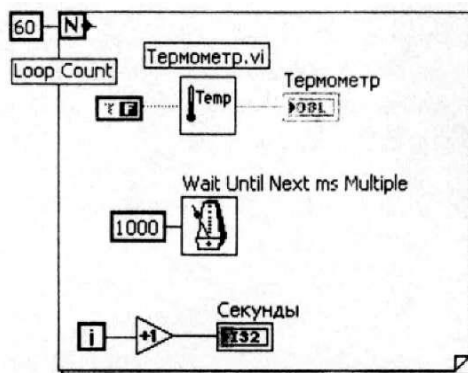


Рис.2. Блок-діаграма



Розташуйте ВП Термометр на блок-діаграму. Для цього виберіть Functions »Select a VI і вкажіть папку, в якій знаходиться прилад, створений в завданні 3.2.

Клацніть правою кнопкою миші по полю введення даних Temp Scale (Шкала температур), в контекстному меню виберіть пункт Create »Constant. Константі надайте значення FALSE - для градусів Фаренгейта і TRUE - для градусів Цельсія.

Розташуйте на блок-діаграму функцію Wait Until Next ms Multiple, що знаходиться на панелі Functions > Programming > Timing. Функція Wait Until Next ms Multiple (чекати кратного значення) забезпечує інтервал між ітераціями, рівний інтервалу часу, необхідного для того, щоб мілісекундний лічильник досяг значення, кратного введеному користувачем. Клацніть правою кнопкою миші по полю введення даних і виберіть пункт Create > Constant. Створеній константі надайте значення 1000. Тепер кожна ітерація циклу виконується з інтервалом часу 1000 мс (раз на секунду).

Розташуйте на блок-діаграму функцію Increment (приріст), що знаходиться на панелі Functions > Mathematics > Numeric. Ця функція додає 1 до значення лічильника ітерацій після завершення виконання циклу.

- 3 Збережіть ВП під ім'ям Вимірювання температури в часі.
- 4 Запустіть ВП.
- 5 Закрийте ВП.

Завдання 4.2

№ вар.	Завдання
1	Створіть ВП, що генерує випадкові числа в циклі While. Організуйте вихід з циклу після натискання кнопки на передній панелі ВП
2	Порахуйте значення виразу $Y = X^2 + Z^3 - XZ + 10$ за допомогою блоку Formula Node та ВП Formula Express, розташованого на панелі Functions »Arith / Compare (арифметика / порівняння)
3	Створіть ВП, що генерує 70 випадкових чисел у циклі FOR
4	Створіть ВП, який за допомогою Formula Node вважає значення виразу $Y = x + \cos(x) - 10$ і якщо $Y \geq 0$, то $Z = \sqrt{Y}$
5	Використовуючи структуру CASE, створіть ВП, який рахує різницю двох чисел і якщо отримане число > 0 , то обчислюється значення кореня, інакше видається повідомлення про помилку
6	ВП здійснює почергове включення індикаторів на передній панелі; інтервали між включеннями індикаторів 2, 3 та 7с., відповідно. Використовуйте послідовність Sequence Structure і функцію Time Delay, розташовану на панелі Functions - Time

	and Dialog
7	Створіть ВП, який вимірює температуру протягом хвилини за допомогою термометра, створеного в завданні 3.2 і рахує середнє значення температури. Підрахунок середнього значення здійснити за допомогою регістру зсуву.
8	Порахуйте значення виразу $Y = X^5 + \cos^2(Z) - XZ + 10$ за допомогою блоку Formula Node та ВП Formula Express, розташованого на панелі Functions - Arith / Compare
9	Використовуючи структуру CASE, створіть ВП, який рахує значення виразу $y = ax + 14$, де $a = \text{const}$; і якщо $y > 0$, то обчислюється значення кореня, інакше видається повідомлення про помилку
10	Створіть ВП, який генерує 70 випадкових чисел і рахує середнє значення. Підрахунок середнього значення здійснити за допомогою регістру зсуву
11	Створіть ВП, який за допомогою Formula Node рахує значення виразу $y = \sin(x)$, якщо y - позитивне число, то $z = y + A$, інакше $z = y - A$
12	ВП здійснює почергове включення індикаторів на передній панелі; інтервали між включеннями індикаторів 5, 8 і 12с., відповідно. Використовуйте послідовність Sequence Structure і функцію Time Delay, розташовану на панелі Functions - Time and Dialog

Контрольні питання

- 1 З яких основних компонентів складається Ваш ВП?
- 2 Які прийоми використання циклу While Ви знаєте?
- 3 Як виміряти температуру з інтервалом раз на секунду протягом однієї хвилини?
- 4 Як і навіщо використовуються зсувні регістри в ВП?
- 5 Навіщо потрібні вузли зворотного зв'язку в ВП?
- 6 Як додати 1 до значення лічильника ітерацій після завершення виконання циклу?
- 7 На які категорії поділяються компоненти, які входять до палітри Controls (Елементів)?
- 8 На які категорії поділяються компоненти, які входять до палітри Functions (функцій)?
- 9 Назвіть призначення кнопок на блок-діаграмі.
- 10 Назвіть основні типи даних.

ЗАНЯТТЯ 5

РОБОТА З МАСИВАМИ В СЕРЕДОВИЩІ LABVIEW

- МЕТА РОБОТИ:**
- вивчити типові прийоми створення масиву, елементів керування та відображення;
 - вивчити типові прийоми створення масиву констант;
 - вивчити функції роботи з масивами, поліморфні функції;
 - освоїти прийоми роботи з масивами.

Теоретичний матеріал

Масиви об'єднують елементи одного типу даних. Масив - це набір елементів певної розмірності. Елементами масиву називають групу складових його об'єктів. Розмірність масиву - це сукупність стовпців (довжина) і рядків (висота), а також глибина масиву. Масив може мати одну і більше за розмірність і до 231-1 елементу в кожному напрямі, наскільки дозволяє оперативна пам'ять.

Дані, що становлять масив, можуть бути будь-якого типу: цілочисельного, логічного або строкового. Масив також може містити елементи графічного представлення даних і кластери. Використати масиви зручно при роботі з групами даних одного типу і при накопиченні даних після обчислень, що повторюються. Масиви ідеально підходять для зберігання даних, отриманих з графіків, або накопичених під час роботи циклів, причому одна ітерація циклу створює один елемент масиву.

Усі елементи масиву впорядковані. Щоб до них було легко звертатися, кожному елементу привласнений індекс. Нумерація елементів масиву завжди починається з 0. Таким чином, індекси масиву знаходяться в діапазоні від 0 до $(n - 1)$, де n - число елементів в масиві.

Створення масиву елементів управління і відображення. Для створення масиву елементів управління або відображення даних, як показано в прикладі, необхідно вибрати шаблон масиву з палітри **Controls – Array & Cluster** і помістити його на лицьову панель. Потім помістити в шаблон масиву елемент керування або відображення даних.

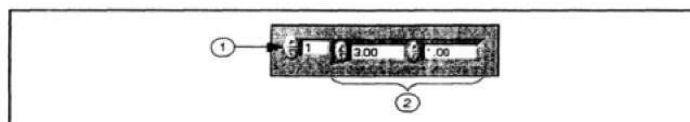


Рис.1. Елемент індексу масиву 2. Елементи значень масиву

Створення масиву констант. Створити масив констант на блок-діаграмі можна, вибравши на панелі Functions »Array шаблон Array Constant і помістивши в нього числову

константу. Масив констант зручно використовувати для передачі даних у підпрограми ВП.

Двовимірні масиви. У двовимірному (2D) масиві елементи зберігаються у вигляді матриці. Таким чином, для розміщення елемента потрібно вказівку індексу стовпця і рядка. На ілюстрації двовимірний масив, що складається з 6 стовпчиків (довжина) і 4 рядків (висота). Кількість елементів у масиві одно 24 ($6 \times 4 = 24$).



Рис.2.

Для збільшення розмірності масиву необхідно клацнути правою кнопкою миші по елементу індексу і вибрати з контекстного меню пункт **Add Dimension (Додати розмір)**. З цією метою також можна використовувати інструмент ПЕРЕМІЩЕННЯ. Для цього треба просто змінити розмір елемента індексу.

Автоматична індексація. Цикл For і цикл While можуть автоматично накопичувати масиви і проводити їх індексацію на своїх кордонах. Це властивість називається *овщай* дексацією. Після з'єднання терміналу масиву з терміналом виходу з циклу кожна ітерація циклу створює новий елемент масиву. На ілюстрації видно, що провідник даних, сполучний термінал даних масиву з терміналом виходу з циклу став *овщай*, а сам термінал виходу з циклу забарвлений в колір терміналу даних масиву.

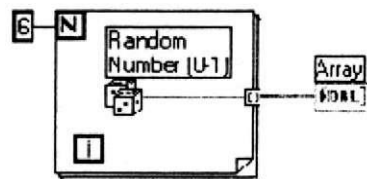


Рис.3.

Автоіндексація відключається клацанням правої кнопки миші по терміналу входу/виходу з циклу і вибором пункту контекстного меню **Disable Indexing** (заборонити автоіндексацію). Автоіндексацію слід відключати, наприклад, у разі, коли треба знати тільки останнє значення.

З огляду на те, що цикл For часто використовується при роботі з циклами, для нього в LabVIEW автоіндексація включена за умовчанням. Для циклу While автоіндексація за умовчанням відключена. Для того, щоб включити автоіндексацію, необхідно клацнути правою кнопкою миші по терміналу входу/виходу з циклу і вибрати в контекстному меню пункт **Enable Indexing** (дозволити автоіндексацію).

Створення двовимірних (2D) масивів. Для створення двовимірних масивів необхідно використати два цикли For, один усередині іншого. Як показано на ілюстрації, зовнішній цикл створює елементи масиву в рядку, а внутрішній цикл створює елементи масиву в стовпці.

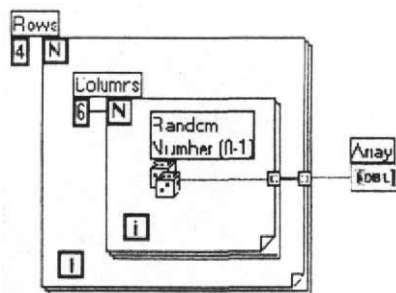


Рис.4.

Функції роботи з масивами. Для створення і управління масивами використовуються функції, розташовані в палітрі **Functions** » **Array**. Найбільш часто використовувані функції роботи з масивами включають:



Array Size (Розмір масиву) - показує кількість елементів масиву кожної розмірності. Якщо масив n-мірний, на виході функції **Array Size** буде масив з n елементів. Наприклад, для наведеного нижче масиву функція **Array Size** видасть значення 3.

7	4	2
---	---	---

Рис.5.



Initialize Array (задати масив) - створює n-мірний масив, в якому кожен елемент іні-зованих значенням поля введення даних **element**. Для збільшення розмірності масиву досить додати поля введення даних, розтягнувши вузол функції. Наприклад, якщо для функції **Initialize Array** задані такі значення параметрів: на полі **element** подається значення 4, а на полі **dimension size** (якщо воно одне) - значення 3, то на виході вийде масив, показаний нижче.

4	4	4
---	---	---

Рис. 6.



Build Array (створити масив) - об'єднує кілька масивів або додає елемент у n-мірний масив. Зміна розміру функції збільшує кількість полів вводу даних, що дозволяє збільшити кількість додаються елементів. Наприклад, якщо об'єднати два попередніх масиву, то функція **Build Array** видасть на виході наступний масив.

7	4	2
4	4	4

Рис. 7.

Для об'єднання вхідних даних в більш довгий масив тієї ж розмірності, як показано нижче, досить клацнути правою кнопкою миші на функції і вибрати з контекстного меню пункт **Concatenate Inputs** (об'єднання вхідних даних).

7	4	2	4	4	4
---	---	---	---	---	---

Рис. 8.



Array Subset (підмножина масиву) - видає частина масиву, починаючи з індексу, що надійшов на полі **index**, і довжиною, зазначеної в полі **length (довжина)**. Наприклад, якщо подати попередній масив на полі введення функції **Array Subset**, значення 2 - на полі **index** і 3 - на полі **Підмножина**, то

2	4	4
---	---	---

Рис.9.



Index Array (індекс масиву) - видає елемент, відповідний індексу, значення якого подається на поле введення **index**. Наприклад, при використанні попереднього масиву, функція **Index Array** видасть значення 2, якщо на полі введення даних **index** подати значення 0.

Функцію **Index Array** можна використовувати для виділення рядка або стовпця з двовимірною масиву і подальшого відображення у вигляді півмасиву. Для цього двовимірний масив треба подати в полі введення даних функції. Функція **Index Array** повинна мати два поля **index**. Верхнє поле **index** вказує рядок, а нижнє - стовець. Можна задіяти обидва поля **index** для вибору окремого елемента або тільки одне, для вибору рядка або стовпця. Наприклад, в полі введення даних функції подається масив, показаний нижче.

7	4	2
4	4	4

Рис. 10.

Функція **Index Array** в поле виводу даних видасть наступний масив у разі, якщо на полі **index** (рядок) подається значення 0.

7	4	2
---	---	---

Рис. 11.

Завдання 5.1. ВП Робота з масивами

Виконайте наступні кроки для створення ВП, який формує масив випадкових чисел, масштабує отриманий масив і виділяє з нього підмножину.

Передня панель

1 Відкрийте новий ВП і створіть лицьову панель, як показано нижче:

- а) у палітрі Controls »Array & Cluster виберіть шаблон масиву;
- б) створеному масиву надайте ім'я Масив випадкових чисел;
- в) Розташуйте всередину шаблону масиву цифровий елемент відображення, розташований на панелі Controls »Numeric;
- г) за допомогою інструменту ПЕРЕМІЩЕННЯ змініть розмір масиву таким чином, щоб він містив 10 пунктів;
- д) натисніть і утримуйте клавішу <Ctrl> і, переміщуючи елемент Масив випадкових чисел, створіть дві його копії;
- е) копіям надайте імена Кінцевий Масив і Підмножина Масиву;
- ж) створіть три цифрових елемента управління і надайте їм імена Масштабний коефіцієнт, Старт підмножини;
- з) клацніть правою кнопкою миші за елементами Старт підмножини і Кількість елементів підмножини, в контекстному меню виберіть пункт Representation, потім пункт 132;
- е) значення елементів управління даних поки не змінюйте.

БЛОК-ДІАГРАМА

2 Побудуйте блок-діаграму, як показано нижче.



 Виберіть функцію **Random Number (0 - 1)**, розташовану на панелі **Functions** »**Numeric**.

Ця функція буде генерувати випадкове число в межах від 0 до 1.



Виберіть цикл **For**, розташований на панелі **Functions** »**Structures**. Цей цикл на терміналі виходу накопичує масив з 10 випадкових чисел. Терміналу кількості ітерацій надайте значення 10.



Виберіть функцію **Array Subset**, розташовану на панелі **Functions** »**Array**. Ця функція видає підмножина масиву, починаючи зі значення, введеного в елементі Старт підмножини, і буде містити кількість елементів, вказане в елементі Кількість елементів підмножини. Збережіть ВП під ім'ям Робота з масивами.

ЗАПУСК ВП

4 Перейдіть на лицьову панель, змініть значення елементів управління та запустіть ВП.

Цикл For здійснить 10 ітерацій. Кожна ітерація створить випадкове число і збереже його в терміналі виходу з циклу. В елементі Масив випадкових чисел відобразиться масив з 10 випадкових чисел. ВП примножить кожне значення цього масиву на число, введене в елемент управління Масштабний коефіцієнт, для створення масиву, відображуваного в індикаторі Кінцевий масив. ВП виділить підмножина з отриманого масиву, починаючи зі значення в елементі Старт підмножини довжиною, зазначеної в елементі Кількість елементів підмножини, і відобразить це підмножина в індикаторі Підмножина масиву.

5 Закрийте ВП.

Завдання 5.2

№ вар.	Завдання
1	Створіть ВП, який повністю змінює порядок елементів у масиві, що містить 10 випадкових чисел. Наприклад, елемент масиву з індексом 0 стає елементом масиву з індексом 9, а елемент масиву з індексом 1 стає елементом масиву з індексом 8, і так далі. Для зміни порядку даних в масиві слід використовувати функцію Reverse ID Array, розташовану на панелі Functions »Array
2	Створіть ВП, який генерує двовимірний масив випадкових чисел, що містить 3 рядки і 10 стовпців
3	Створіть ВП, який генерує одновимірний масив, у якому 80 випадкових чисел, і видає частина масиву, починаючи з індексу 15 до індексу 60. На лицьову панель вивести масив випадкових чисел і отриманий масив
4	Створіть ВП, який генерує одновимірний масив випадкових чисел до того часу,

	поки не натиснута кнопка на передній панелі. На лицьову панель вивести отриманий масив і його розмірність
5	Створіть ВП, який генерує одновимірний масив і потім 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 на панелі Functions »Array
6	Створіть ВП, який генерує одновимірний масив випадкових чисел і сортує отриманий масив в порядку зростання. На лицьову панель вивести масив випадкових чисел і відсортований масив. Для сортування елементів у масиві слід використовувати функцію Sort 1D Array, розташовану на панелі Functions »Array
7	Створіть ВП, який генерує одновимірний масив випадкових чисел і виводить максимальне значення отриманого масиву і його порядковий номер. Використовувати функцію Array Max & Min, розташовану на панелі Functions »Array
8	Створіть ВП, який генерує одновимірний масив випадкових чисел і виводить мінімальне значення отриманого масиву і його порядковий номер. Використовувати функцію Array Max & Min, розташовану на панелі Functions »Array
9	Створіть ВП, який генерує двовимірний масив випадкових чисел, що містить 4 рядки і 5 стовпців
10	Створіть ВП, який генерує два одномірних масиву випадкових чисел і об'єднує ці масиви в двовимірний масив чисел. На лицьову панель вивести два вихідних масиву випадкових чисел і двовимірний масив, що складається з елементів вихідних масивів
11	Створіть ВП, який генерує двовимірний масив випадкових чисел розмірністю 5 x 6 і видає частина цього масиву розмірністю 4 x 5. На лицьову панель вивести вихідний масив випадкових чисел і отриманий масив
12	Створіть ВП, який генерує двовимірний масив випадкових чисел і здійснює транспонування отриманого масиву. На лицьову панель вивести масив випадкових чисел і транспонований масив. Для транспонування масиву використовуйте функцію Transpose 2D Array, розташовану на панелі Functions »Array

Контрольні питання

1 З яких основних компонентів складається Ваш ВП?

- 2 Які типові прийоми створення масиву констант Ви знаєте?
- 3 Які функції роботи з масивами Ви знаєте?
- 4 Що таке поліморфні функції?
- 5 Які прийоми роботи з масивами доступні для лицьової панелі?
- 6 Які палітри доступні для блок-діаграми?
- 7 Що Ви знаєте про цикл For?
- 8 Як створюються багатомірні масиви?
- 9 Яке призначення елемента Старт підмножини в підпрограмі ВП?
- 10 Як вибирається кількість елементів підмножини в підпрограмі ВП?
- 11 Яка функція генерує випадкове число в межах від 0 до 1?
- 12 Які функції створення масивів Ви знаєте?

ЗАНЯТТЯ 6

СТВОРЕННЯ КЛАСТЕРІВ З ЕЛЕМЕНТІВ УПРАВЛІННЯ ТА ВІДОБРАЖЕННЯ ДАНИХ. РОБОТА З КЛАСТЕРАМИ.

Масштабування КЛАСТЕРУ

- МЕТА РОБОТИ:** - вивчення типових прийомів створення кластерів і функцій відображення кластерів;
- створення кластерів на передній панелі;
 - збірка і демонтування кластерів з використанням функцій обробки кластерів;
 - створення ВП, що використовує поліморфізм у кластерах;
 - вивчення кластерів помилок.

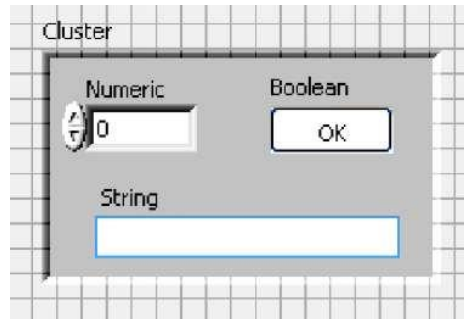
Теоретичний матеріал

Як і масив, кластер є структурою, що групує дані. Однак на відміну від масиву кластер може групувати дані різних типів (логічні, числові і т.д.). Об'єднання декількох груп даних у кластер усуває безлад на блок-діаграмі і зменшує кількість полів вводу/виводу даних, необхідних підпрограмі ВП. Максимально можлива кількість полів вводу/виводу даних ВП одно 28. Якщо Передня панель містить більше 28 елементів, які необхідно використовувати у ВП, можна деякі з них об'єднати в кластер і зв'язати кластер з полем вводу / виводу даних. Як і масив, кластер може бути елементом управління або відображення даних, однак кластер не може одночасно містити елементи управління і відображення даних.

Створення кластерів з елементів управління і відображення даних. Для створення кластерів з елементів управління і відображення даних слід вибрати шаблон на

палітрі **Controls** » **Array & Cluster** і помістити його на лицьову панель. Після цього шаблон кластера слід заповнити елементами. Змінити розмір кластера можна за допомогою курсору.

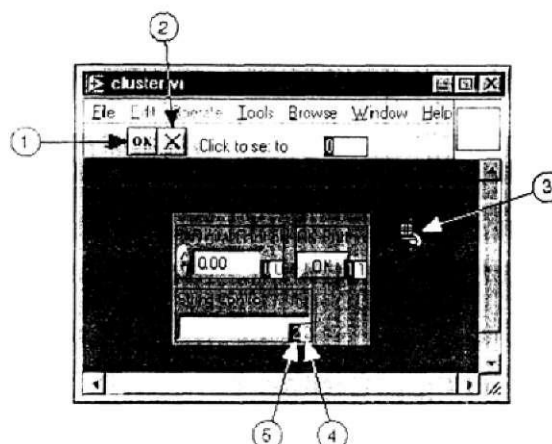
Нижче показаний кластер, який містить три елементи управління.



Порядок елементів у кластері. Кожен елемент кластера має свій логічний порядковий номер, не пов'язаний з положенням елемента в шаблоні. Першому вміщеної у кластер елементу автоматично присвоюється номер 0, другого елементу - 1 і так далі. При видаленні елемента порядкові номери автоматично змінюються.

Порядок елементів у кластері визначає те, як елементи кластера будуть розподілені по терміналах функцій **Bundle (об'єднання)** і **Unbundle (поділу)** на блок-діаграмі.

Переглянути та змінити порядковий номер об'єкта, поміщеного в кластер, можна, клацнувши пра-вої кнопкою миші по краю кластеру та вибравши з контекстного меню пункт **Reorder Controls In Cluster**. Панель інструментів і кластер приймуть вигляд, показаний нижче на рисунку.



У білому полі (4) із поточним порядковий номер елемента, в чорному (5) - новий порядковий номер. Для установки порядкового номера елемента потрібно в полі введення тексту **Click to set to** ввести число і натиснути на елемент. Порядковий номер елемента

зміниться. При цьому коригуються порядкові номери інших елементів. Зберегти зміни можна, натиснувши кнопку **ОК (підтвердити)** на панелі інструментів.

Створення кластеру констант. На блок-діаграмі можна створити кластер констант, вибравши на панелі **Functions** шаблон **Cluster Constant** і помістивши в нього числову константу або інший об'єкт даних, логічний або рядковий.

Функції роботи з кластерами. Для створення і управління кластерами використовуються функції, розташовані на палітрі **Functions** »**Cluster**. Функції **Bundle** (Зв'язати) і **Bundle by Name** (Зв'язати за назвою) використовуються для побудови та управління кластерами. Функції **Unbundle** (Розділити) і **Unbundle by Name** (Розділити за назвою) використовуються для розбирання кластерів.

Ці функції також можна викликати, клацнувши правою кнопкою миші по терміналу даних кластеру та вибравши з контекстного меню підменю **Cluster Tools** (Інструменти кластерів). Функції **Bundle** і **Unbundle** автоматично містять правильне кількість полів вводу / виводу даних. Функції **Bundle by Name** і **Unbundle by Name** в полях вводу / виводу даних містять ім'я першого елемента кластеру.

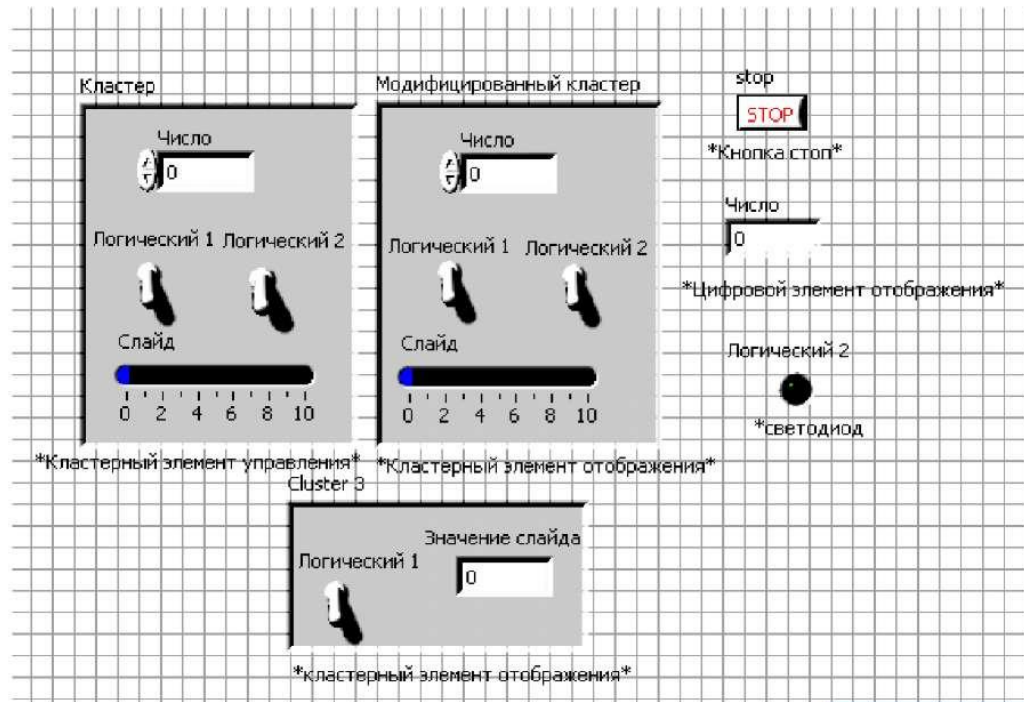
Іноді зручно поміняти масиви на кластери і навпаки, оскільки LabVIEW включає в себе набагато більше функцій, що працюють з масивами, ніж з кластерами. Для перетворення кластера в масив служить функція **Кластер в масив (Cluster to Array)**. Зворотна операція здійснюється за допомогою функції **Масив в кластер (Array to Cluster)**. Функція Кластер в масив конвертує кластер з кількістю елементів N одного типу даних у масив з кількістю елементів N того ж типу даних. Індекс масиву відповідає порядковому номеру в кластері (тобто нульовий елемент кластеру стає значенням масиву з індексом 0). Слід звернути увагу, що при використанні цієї функції всі елементи в кластері повинні бути одного типу.

Функція **Масив у кластер** перетворює одновимірний масив з числом елементів N в кластер з числом елементів N того ж типу даних. Для включення цієї функції необхідно натиснути правою кнопкою миші по терміналу **Масив в Кластер** і вибрати опцію **Розмір кластера (Cluster Size)** для встановлення розміру вихідного кластера, оскільки кластери, на відміну від масивів, не встановлюють свій розмір автоматично. Розмір кластера за замовчуванням дорівнює 9. Якщо масив має меншу кількість елементів, ніж це визначено розміром кластера, LabVIEW автоматично створить додаткові елементи кластеру зі значеннями за замовчуванням для типу даних кластеру. Однак, якщо кількість елементів вхідного масиву більше величини, встановленої у вікні розміру кластера, то провідник блок-діаграми, що йде до вихідного кластеру, буде розірваним, поки не буде відрегульований його розмір.

Завдання 6.1. ВП Робота з кластерами

Передня панель

1 Відкрийте новий ВП і створіть лицьову панель, як показано нижче:



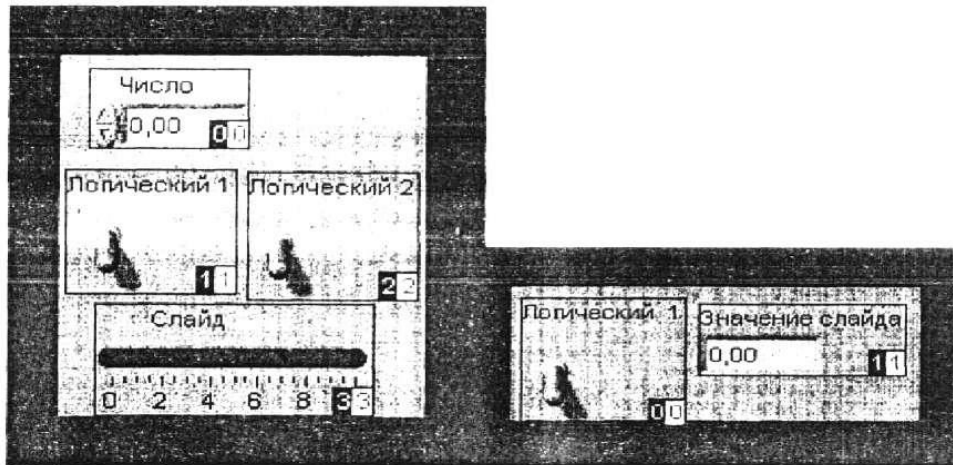
- Розташуйте на лицьову панель кнопку «Стоп» та круглий світлодіод;
- з палітри Controls »Array & Cluster виберіть шаблон кластеру;
- об'єкти передньої панелі, показані на ілюстрації, Розташуйте в шаблон кластеру;
- створіть і перейменуйте копію елемента Кластер в Модифікований Кластер.

Після цього клацніть правою кнопкою миші по межі шаблону кластеру Модифікований Кластер і виберіть з контекстного меню пункт Change to Indicator;

д) повторите пункт г) для створення елемента Маленький кластер. Змініть його, як показано на рисунку.

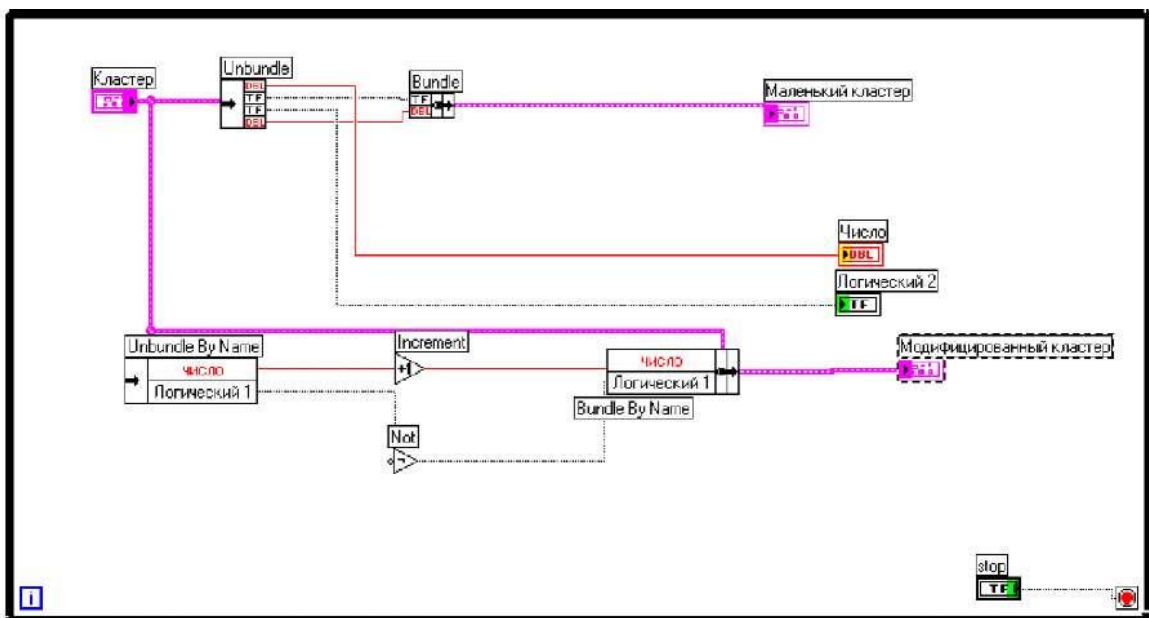
2 Перевірте порядкові номери елементів у кластерах Кластер і Маленький кластер. Порядкові номери елементів кластерів Модифікований кластер і Кластер повинні збігатися:

- клацніть правою кнопкою миші по межі шаблону кожного кластера, з контекстного меню виберіть пункт Reorder Controls in Cluster;
- порядкові номери елементів установіть, як показано нижче на рисунку.



Блок-діаграма

3 Створіть блок-діаграму, як показано нижче.



З палітри **Functions** »**Cluster** виберіть функцію **Unbundle**. Ця функція роз'єднує кластер Кластер. Змініть розмір цієї функції до чотирьох полів введення даних або з'єднайте термінал даних кластера з функцією для автоматичного додавання полів введення даних.

З палітри **Functions** »**Cluster** виберіть функцію **Bundle**. Ця функція об'єднує елементи в кластер Маленький кластер.

З палітри **Functions** »**Cluster** виберіть функцію **Unbundle by Name**. Ця функція виділить два елементи з кластера Кластер. Змініть розмір функції до двох полів виводу даних. Якщо імена в полях виведення даних відрізняються від показаних на

ілюстрації, слід клацнути правою кнопкою миші по імені елемента й у контекстному меню увійти до розділу **Select Item**.

З палітри **Functions** »**Numeric** виберіть функцію **Increment**. Ця функція додасть 1 до значення елемента Число.

З палітри **Functions** »**Boolean** виберіть функцію **Not**. Ця функція видасть логічне заперечення елемента Логічний 1.



З палітри **Functions** »**Cluster** виберіть функцію **Bundle by Name**. Ця функція змінить значення елементів Число і Логічний в кластері Кластер і створить кластер Модифікований кластер. Змініть розмір цієї функції на два поля вводу даних. Якщо імена в полях виведення даних відрізняються від показаних на зображенні, слід натиснути правою кнопкою миші по імені елемента й у контекстному меню увійти до розділу **Select Item**.

- 4 Збережіть ВП під ім'ям Робота з кластерами, ві.
- 5 Перейдіть на лицьову панель і запусіть ВП.
- 6 Поміняйте значення елементів в кластері Кластер і запусіть ВП.
- 7 Закрийте ВП.

Контрольні питання

- 1 З яких основних компонентів складається Ваш ВП?
- 2 Що розуміється під терміном Кластер?
- 3 Які типові прийоми створення кластерів Ви знаєте?
- 4 Які функції відображення кластерів Ви знаєте?
- 5 Як створити кластер на передній панелі?
- 6 Як збираються і демонтуються кластери?
- 7 Які функції обробки кластерів Ви знаєте?
- 8 Що таке поліморфізм у кластерах?
- 9 Як створити модифікований кластер?
- 10 Яке основна відмінність кластеру від масиву?
- 11 Який порядок розміщення елементів у кластері?
- 12 Як можна побачити і змінити порядковий номер об'єкта, поміщеного в кластер?
- 13 Як змінити кількість полів вводу виводу в кластері?
- 14 Як встановлюють розмір кластера?
- 15 Який розмір кластера за умовчанням?
- 16 Як створити та перейменувати копію елемента Кластер?

ЗАНЯТТЯ 7

ВИВЧЕННЯ СТРОКОВИХ РЕГУЛЯТОРІВ ТА ІНДИКАТОРІВ.

Рядки і файловий ввід-вивід

- МЕТА РОБОТИ:** - вивчення строкових регуляторів та індикаторів.
- використання строкових функцій.
- виконання операцій файлового вводу-виводу.

Теоретичний матеріал

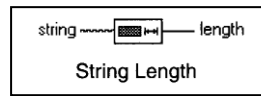
Рядок - послідовність відображуваних або неотоображаємих символів. Часто, необхідно використовувати такі рядки більш ніж для виведення простого тексту (наприклад, ASCII повідомлення). Наприклад, при інструментальному управлінні, числові дані передаються як символні рядки. Потім ці рядки перетворюються на числа. У багатьох випадках, збереження числових даних на диску також здійснюється у вигляді рядків, це означає, що спочатку необхідно перетворити числа в рядки перед записом їх у файл на диску.

Строкові регулятори та індикатори перебувають в субпалітрі **String & Table** палітри **Controls**. Текст вводиться або змінюється всередині строкового регулятора, використовуючи Робочий інструмент або інструмент Мітка. Збільшувати рядкові регулятори та індикатори, можна переміщаючи кут елемента інструментом **Позиціонування**. Щоб мінімізувати простір, який займають рядки на лицьовій панелі, використовується опцію **Show Scrollbar** з строкового контекстного меню. Якщо ця опція блякла, необхідно збільшити вертикальний розмір вікна. Також можна конфігурувати рядкові засоби керування та індикатори для різних типів дисплея. Наприклад, можна вибирати дисплей пароля, включивши опцію **Password Display** з контекстного меню рядка. При виборі цієї опції, на дисплеї з'являються тільки зірочки. На блок-схемі, строкові дані відображають те, що вводилося. Строкові кошти управління і індикатори також можуть відображати і приймати символи, які є зазвичай неотоображаємими, типу повернень на один символ, повернень каретки, міток табуляції, і так далі. Щоб відображати ці символи, необхідно вибрати пункт **' Codes Display** з контекстного меню рядка.

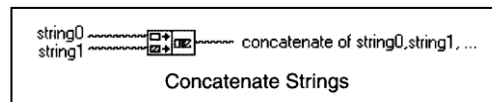
У режимі **' Codes Display**, неотоображаємиє символи з'являється як згадано вліво, супроводжувана відповідним кодом. Символи, які містяться в строкових засобах управління і індикаторах представляються внутрішньо у форматі ASCII. Щоб переглядати фактичні коди ASCII в шістнадцятковому форматі, необхідно вибрати пункт **Hex Display** з контекстного меню рядка.

Строкові функції.

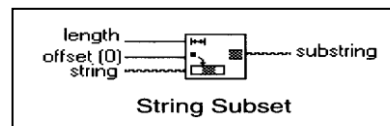
String Length - повертає число символів у рядку.



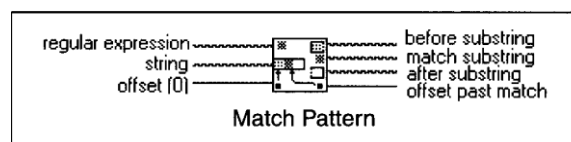
Concatenate Strings - об'єднує всі вхідні рядки і масиви рядків, в одну вихідну рядок.



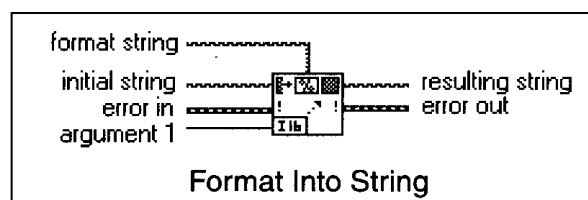
String Subset - повертає підрядок, що починається з **offset** (зсув) і містять **length** (число) символів довжини. Перше символічне зсув - нуль.



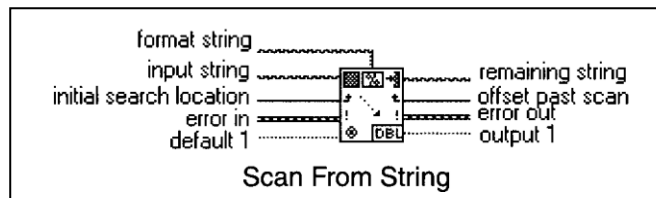
Match Pattern - повертає узгоджену підрядок. Функція шукає **regular expression** в **string** починаючи з **offset**, і якщо знаходить відповідність, розбивають рядок на три підрядка. Якщо ніяке відповідність не знайдено, підрядок відповідності порожня, **offset past match is -1**



Format Into String - перетворює будь-який формат **argument** (наприклад, числовий) до певного формату **resulting string**. Можна розширювати функцію, щоб мати кілька значень, перетворені в одиночну рядок. Функція може формувати вихідну рядок з **initial string** і **argument** (s), що базуються на **format string**.



Scan From String - перетворює рядок, що містить допустимі цифри (від 0 до 9, +, -, e, E, і точку) на число. Функція починає проглядати вхідну рядок з **input string at initial search location**. Функція може переглядати вхідні рядок для різних типів даних (наприклад, цифрові або Булевського) заснованих на рядку формату. Ця функція розширювана, щоб отримати кілька виходів.



LabVIEW має багато вбудованих функцій і **Vis**, щоб обробити файл Введення - висновок. Всі функції файлового Введення - виведення знаходяться в субпалітрі **File I/O** палітри **Functions**. Ці функції і **Vis** організовані в три рівні ієрархії:

- a. **Vis** файлових функції високого рівня;
- b. Проміжні файлові функції;
- c. Просунуті файлові функції.

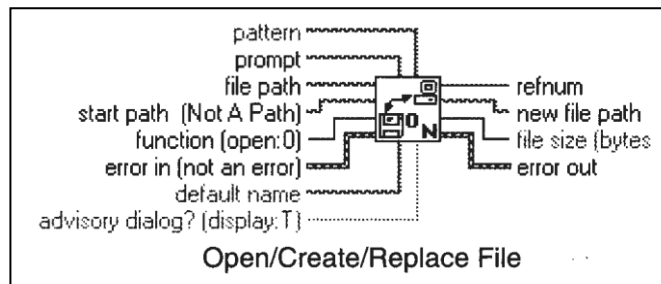
Vis файлових функції високого рівня - дев'ять **Vis** файлів високого рівня знаходиться у верхній частині субпалітри **File I/O**, який включає субпалітру для **Binary File Vis**. Ці **Vis** викликають проміжні файлові функції як **subVIs**. Вони спрощують найбільш загальні типи файлового введення - виведення, з яким стикаються при роботі з LabVIEW, обробляючи функції нижнього рівня. **Vis** також створюють спрощені засоби обробки помилок. Якщо помилка файлового введення - виведення відбувається протягом виконання одного з цих **Vis**, діалогове вікно, показує помилку.

Проміжні файлові функції знаходяться в другому рядку субпалітри **File I/O**. Вони забезпечують в основному більшу кількість функціональних можливостей ніж **Vis** високого рівня, типу **programmatic file opening i closing and direct managing** для файлового читання і запису маркерів.

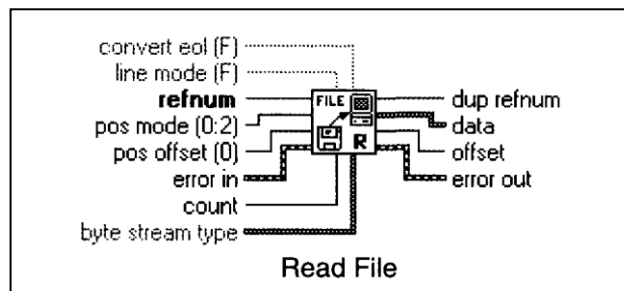
Просунуті файлові функції знаходяться в субпалітрі **File I/O** палітри **Functions**. Ці вбудовані функції детально обробляють файлу операції **I/O** LabVIEW і забезпечують гнучкість в управлінні файловим вводом - виводом.

Базовий процес файлового вводу - виводу на проміжному рівні повинен відкривати або створити файл, читати з або записувати в файл, і потім, закрити його. Основними **VI** є **Open/Create/Replace File VI**, **Read File Vis**, **Write File Vis**, і **Close File VI**. Також важливим є **VI** обробника помилок (**Simple Error Handler VI**).

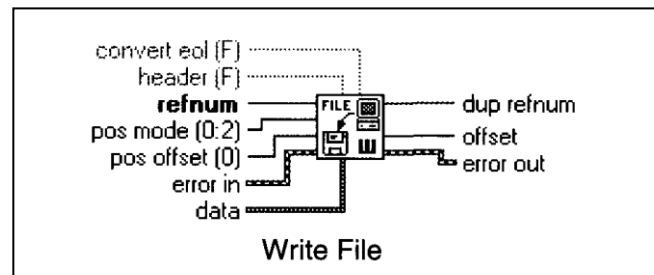
Open/Create/Replace File відкриває, замінює існуючий файл або створює новий файл. Якщо Ви залишаєте шлях файлу непідключеним, **VI** відображає файлове діалогове вікно, з якого можна вибирати новий або існуючий файл. Після того, як файл відкривається або створюється, можна читати дані з файлу або записувати в нього дані, використовуючи Файл **Read** і Опції **Write File**. Ви можете читати або записувати будь-який тип даних, використовуючи **Read File** і **Write File** функції.



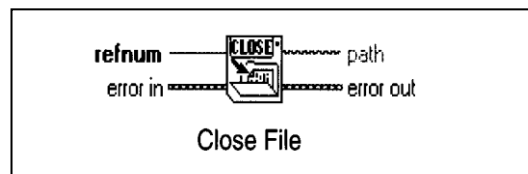
Read File читає **count** байт даних з файлу, специфікованому **refnum** і повертає їх у **data**. Читання починається з точки, визначеної **pos mode** і **pos offset**.



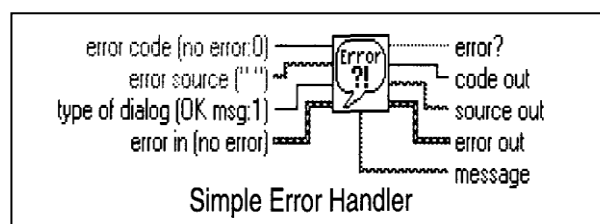
Write File записує файл, який **refnum** визначає. Запис починається з положенні, визначеному **pos offset** і **pos mode**.



Close File закриває файл, пов'язаний з **refnum**. Цей VI закриває файли даних всіх типів.



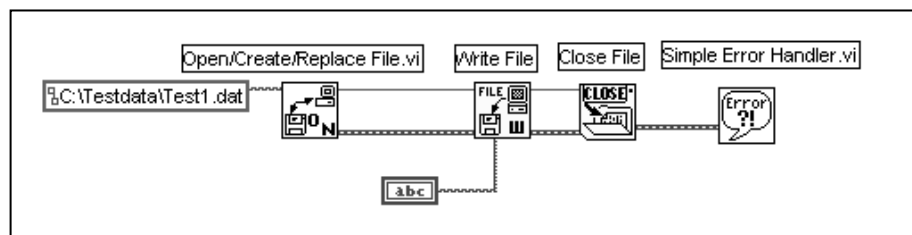
Simple Error Handler (Time & Dialog субпалітра) перевіряє помилки у файлових операціях і відображає діалогове вікно, якщо помилка відбувається.



Збереження даних у новому або існуючого файлі.

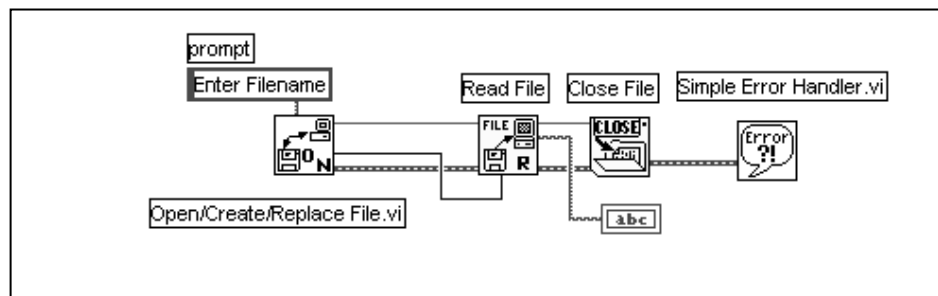
Збереження даних у новому або існуючому файлі - процес з трьома кроками: а - відкрити або створити файл, б - записати дані у файл, с - закрити файл. Використовуючи файлові Vis, можна записувати будь-який тип даних у файл, який відкритий або створений. Якщо інші користувачі або програми повинні звернутися до файлу, необхідно записати строкові дані в ASCII ФОРМАТІ у файл. Можна звертатися до файлів або програмно, або через діалогове вікно. Щоб звертатися до файлу через інтерактивне діалогове вікно файлу, необхідно залишити шлях файлу невідключеним в **Open/Create/Replace File VI.** Можна програмно зв'язати ім'я файлу та ім'я шляху в **VI.**

Приклад запису строкових даних в існуючий файл при зв'язку імені файлу та імені шляхи:



Читання даних з файлу.

Коли необхідно прочитати дані з файлу, спочатку відкривається існуючий файл, читається зміст файлу функцією **Read File**, і закривається файл. Також необхідно визначити кількість даних, які потрібно читати. Наступний приклад показує кроки для читання всього змісту строкового файлу, використовуючи інтерактивне діалогове вікно файлу, щоб вибрати файл:

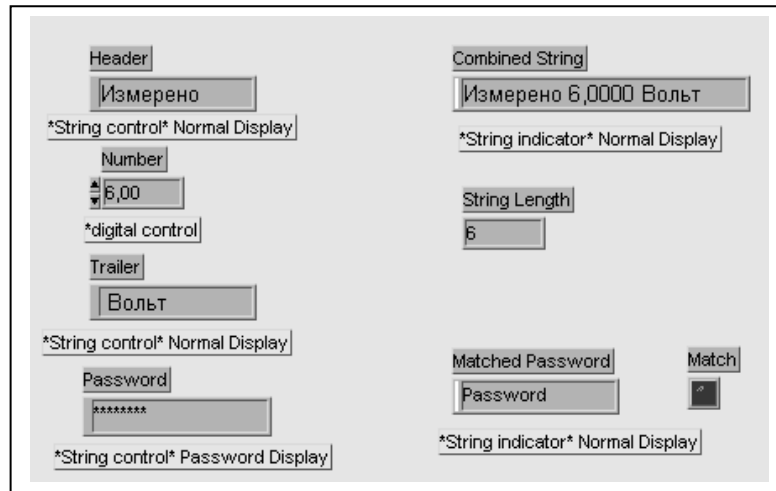


Завдання 7.1

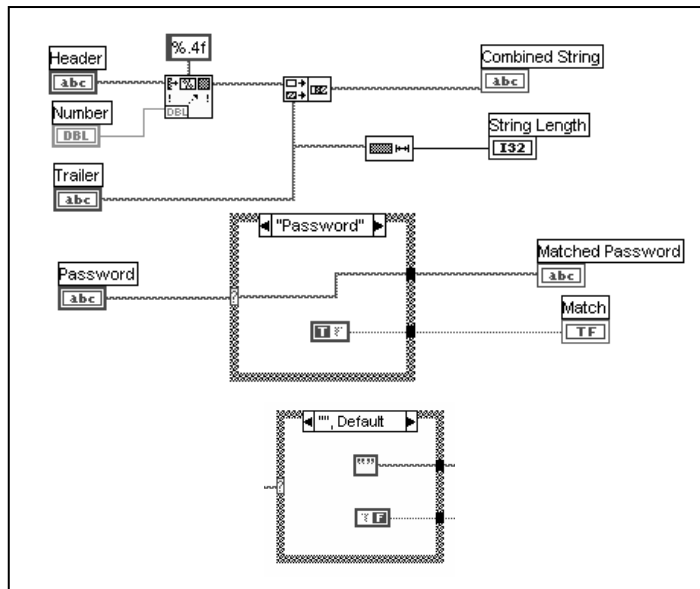
Побудувати **VI** перетворює рядок у число, що поєднує кілька рядків в один рядок, що визначає довжину рядка і перевіряючий пароль.

Відкрити новий **VI**.

Побудувати лицьову панель, як показано нижче.



Побудувати блок - діаграму, як показано нижче:



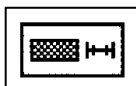
При побудові діаграми використовувати наступні функції:



Функція **Format Into String** (String субпалітра) перетворить число, специфікується в цифровому регуляторі в рядок. Для створення рядка форматування `%.4 f`, необхідно з контекстного меню функції вибрати пункт **Edit Format String**.

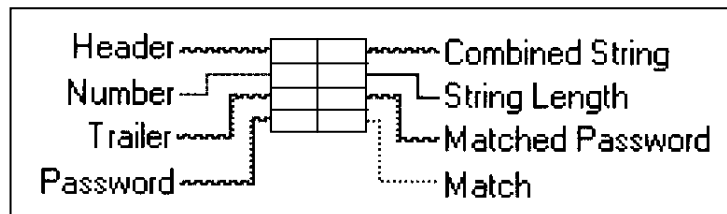


Функція **Concatenate Strings** (String субпалітра), об'єднує всі вхідні рядки в одну вихідну.



Функція **String Length** (String субпалітра), повертає число символів в об'єднаній рядку.

Створити **SubVI**, відкривши лицьову панель, і побудувавши ікону - з'єднувач.



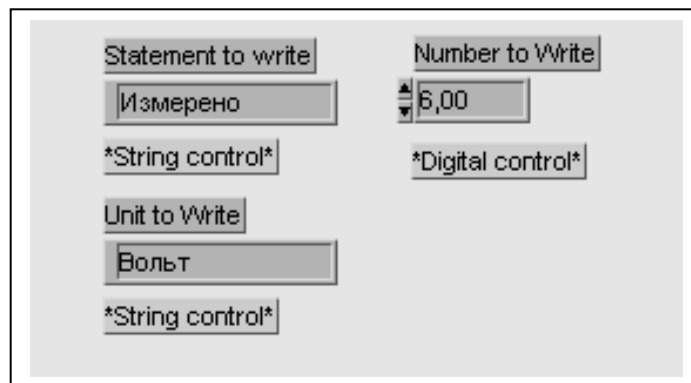
Зберегти VI під ім'ям Build String. Vi

Запустити VI, і досліджувати його роботу.

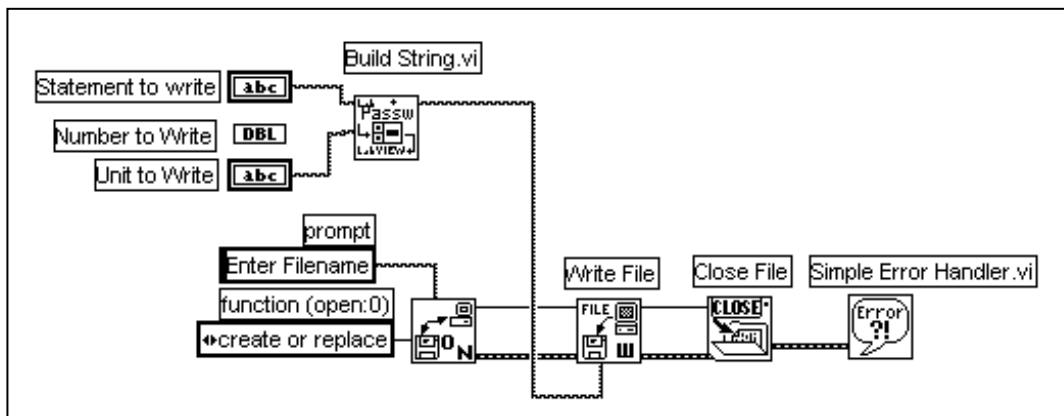
Доопрацювати VI, для вставки пропусків, між **Header, Number, і Trailer**.

Побудувати VI, який об'єднує повідомлення, рядок і одиниці вимірювань, і записуючий результат у файл, і VI зчитує файл, якщо введений пароль.

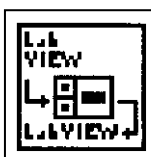
Відкрити новий VI, і побудувати лицьову панель, як показано нижче:



Побудувати блок - діаграму, як показано нижче:



При побудові використовувати наступні елементи:



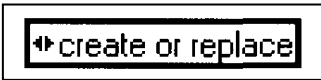
Build String.vi (Select a VI ... субпалітра) - VI побудований попередньо.



Open/Create/Replace File VI (File I/O субпалітра) - відображає інтерактивний файловий діалог відкриття або закриття файлу.



A - контекстне меню VI prompt і вибрати пункт **Create Constant**

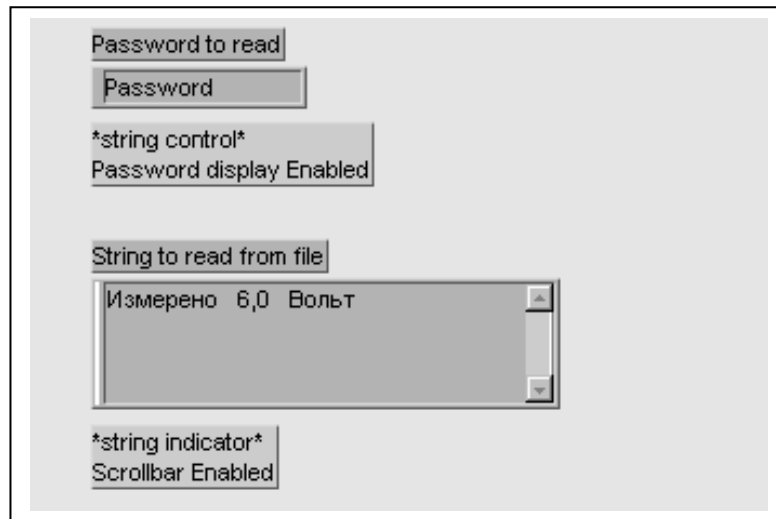


B - контекстне меню VI **function** і вибрати пункт **Create Constant**, використовуючи **Operating tool** змінити термінал на **creat or replace**.

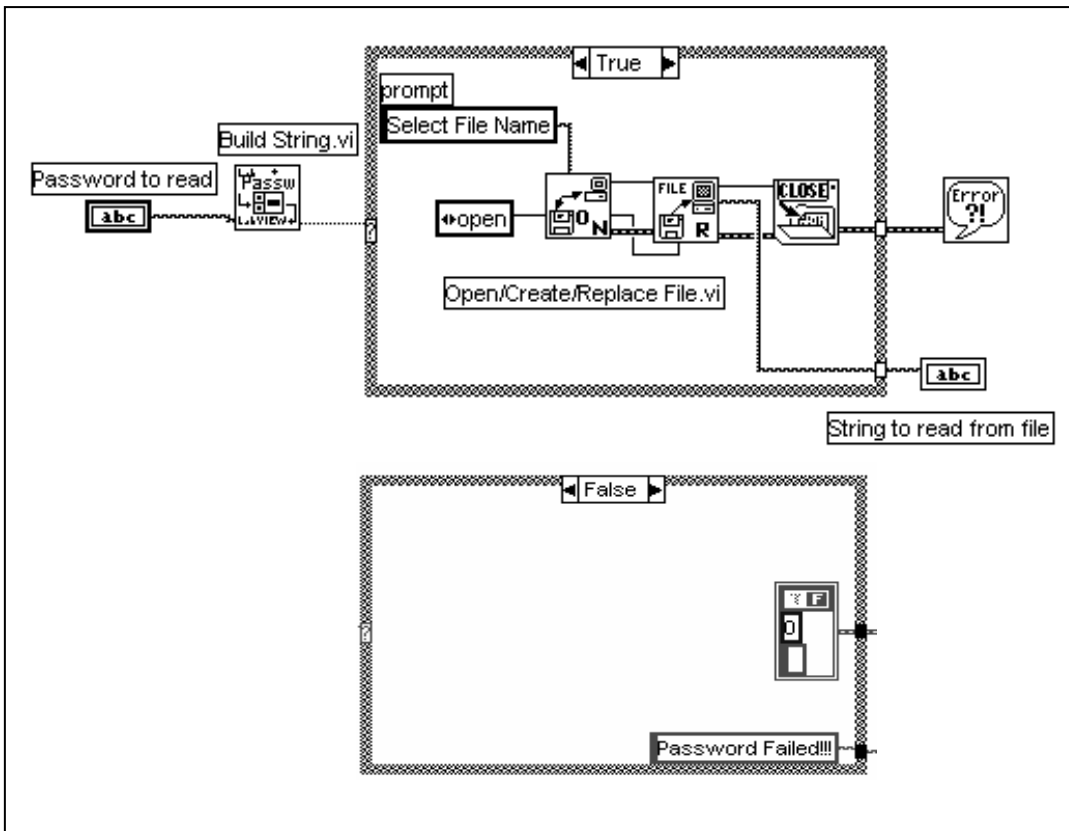
Зберегти VI під ім'ям **File Writer. Vi**.

Ввести значення в поля, і запустити VI. У діалоговому вікні ввести ім'я файлу **demofile.txt**, натиснути кнопку ОК.

Відкрити новий VI, і побудувати лицьову панель, як показано нижче:



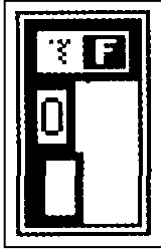
Побудувати блок - діаграму, як показано нижче:



При побудові діаграми, використовувати такі функції:



Simple Error Handler VI (Time & Dialog субпалітра). Перевіряє помилки і відображає діалогове вікно, у разі виявлення помилок.



Константа Error Cluster, відкривається з контекстного меню тунелю при виборі пункту Create Constant фрейму False.

Зберегти VI, запустити VI, і досліджувати роботу.

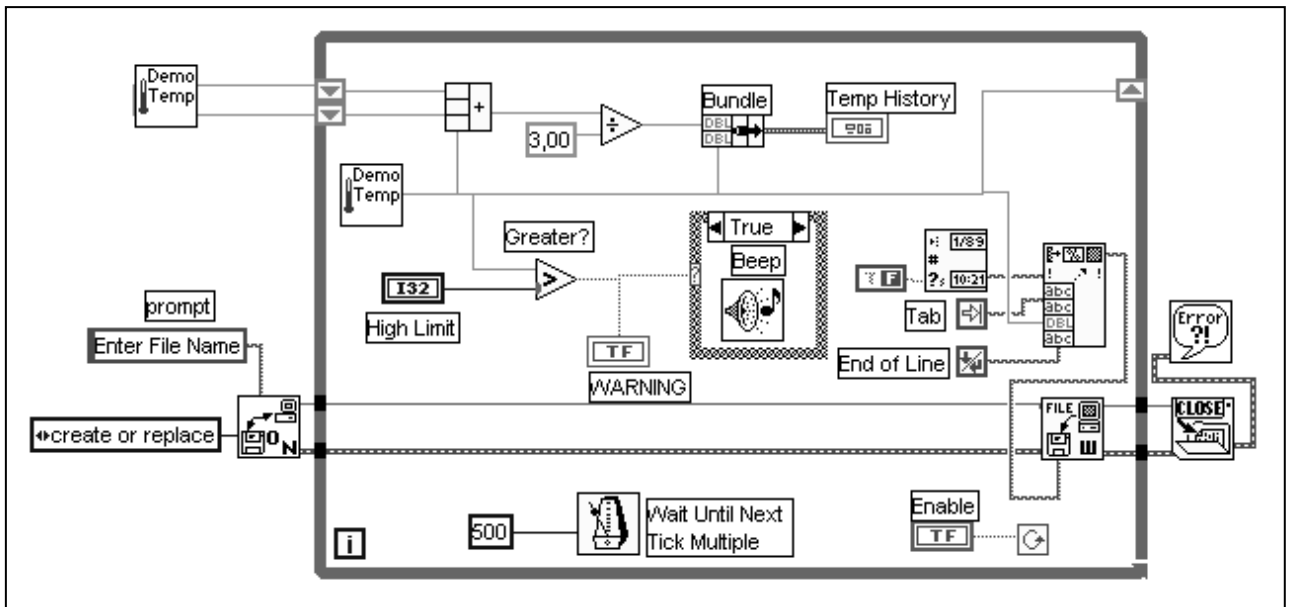
Модернізувати VI таким чином, щоб число виділялося з рядка, і записувалася в числовий індикатор.

Побудувати VI реєструючий температуру і зберігає дані на диску.

Відкрити VI Temperature Control.

Зберегти VI під ім'ям Temperature Logger.vi.

Модернізувати блок - діаграму, як показано нижче:



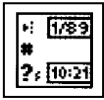
При побудові діаграми використовувати наступні елементи:



Константа **Tab (String субпалітра).**



Константа **End of Line (String субпалітра).**



Функція **Get Date/Time String (Time & Dialog** субпалітра), повертає час (у строковому форматі), коли проводиться вимірювання. Булевський константа дозволяє включати або вимикати запис секунд.



Функція **Format into String** - перетворить вимір (число), в рядок, і буде форматований рядок.

Запустити VI, і на запит імені файлу, ввести Temp.txt.

Через деякий час зупинити VI, і засобами Windows переглянути вміст файлу Temp.txt.

Зберегти VI.

Контрольні питання

1. Функції роботи з рядками.
2. Об'єднання і перетворення рядків.
3. Збереження файлів.
4. Читання файлів.

СПИСОК ЛІТЕРАТУРИ

- 1 Использование виртуальных инструментов Lab VIEW / Под ред. К.С. Демирчяна и В.Г. Миронова. М.: Солон-Р; Радио и связь, Горячая линия - Телеком, 1999. 268 с.
- 2 Евдокимов Ю. К., Линдваль В. Р., Щербаков Г. И. LabVIEW для радиоинженера: от виртуальной модели до реального прибора. Практическое руководство для работы в программной среде LabVIEW. - М.: ДМК Пресс, 2007. - 400 с. (Электронный вариант).
- 3 Батоврин В. К., Бессонов А. С», Мошкин В. В., Папуловский В. Ф. LabVIEW: практикум по основам измерительных технологий; Учебное пособие для вузов. - М: ДМК Пресс, 2005. - 208 с.: ил. (Электронный вариант).
- 4 Тревис Дж. Lab VIEW для всех / Джеффри Тревис: Пер. с англ. Клушин Н. А. - М.: ДМК Пресс; ПриборКомплект, 2005. - 544 с.: ил. (Электронный вариант).