

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ
по виконанню лабораторних робіт
з дисципліни
«Моделювання динамічних об'єктів»
для студентів 4 курсу
Освітньо-кваліфікаційний рівень – «Бакалавр»
Напрямок підготовки – «Комп'ютерні науки»
Спеціальність – інформаційні управляючі системи та технології

Затверджено на засіданні
методичної комісії ф-ту
Комп'ютерних наук
Протокол №__ від _____
Декан факультету
_____ Коваленко Л.Б.

Затверджено на засіданні
кафедри Інформатики
Протокол №__ від _____
Зав. кафедри _____ Мещеряков В.І.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ
по виконанню лабораторних робіт
з дисципліни
«Моделювання динамічних об'єктів»
для студентів 4 курсу
Освітньо-кваліфікаційний рівень – «Бакалавр»
Напрямок підготовки – «Комп'ютерні науки»
Спеціальність – інформаційні управляючі системи та технології

Одеса, 2015

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ
по виконанню лабораторних робіт
з дисципліни
«Моделювання динамічних об'єктів»
для студентів 4 курсу
Освітньо-кваліфікаційний рівень – «Бакалавр»
Напрямок підготовки – «Комп'ютерні науки»
Спеціальність – інформаційні управляючі системи та технології

Затверджено на засіданні
методичної комісії ф-ту
Комп'ютерних наук
Протокол №__ від _____

Одеса, 2015

Методичні вказівки до виконання лабораторних робіт з дисципліни «Моделювання динамічних об'єктів» для студентів 4 курсу, освітньо-кваліфікаційний рівень – «Бакалавр», напрям підготовки – «Комп'ютерні науки», спеціальність – інформаційні управляючі системи та технології

Укладачі:

Мещеряков В.І., д.т.н., професор, зав. кафедри інформатики

Черепанова К.В., асистент кафедри інформатики.

ЗМІСТ

Передмова	6
Теоретичні відомості	7
Лабораторна робота №1	9
Запуск LabVIEW.....	9
Лабораторна робота №2	17
Середовище проектування LabVIEW	17
Лабораторна робота №3	26
Створення, редагування та налагодження ВП	26
Лабораторна робота №4	34
Створення підпрограм віртуального приладу (ВП)	34
Лабораторна робота №5	37
Управління виконанням програм за допомогою структур.....	37
Лабораторна робота №6	51
Складові дані LabView: масиви	51
Лабораторна робота №7	64
Складові дані LabView: кластери.....	64
Література.....	74

Передмова

Мета методичних вказівок – поглиблене вивчення та закріплення лекційного матеріалу щодо вивчення дисципліни «Моделювання динамічних об'єктів».

Метою дисципліни є ознайомлення студентів з основними задачами моделювання динамічних об'єктів.

Метою викладання дисципліни також є формування у студентів знань про новітні інформаційні системи та методи моделювання, їх використання для вирішення практичних завдань, набуття навичок роботи з програмним забезпеченням, що використовується у сучасній практиці.

У результаті вивчення дисципліни «Моделювання динамічних об'єктів» студент повинен:

ЗНАТИ:

- принципи моделювання об'єктів
- об'єктно-орієнтоване програмування

ВМІТИ:

- розробляти віртуальні автоматичні пристрої
- розробляти моделі програмних продуктів

По кожній лабораторній роботі студент повинен скласти звіт, якій містить в собі:

1. Назву роботи. Мету.
2. Умову завдання згідно варіанта.
3. Хід виконання роботи.
4. Відповіді на контрольні питання.

Оформлений звіт захищається студентом усно.

Варіант індивідуального завдання надається викладачем.

Перед виконанням лабораторних робіт у комп'ютерному класі студенти зобов'язані пройти інструктаж з техніки безпеки та охорони праці.

Правила техніки безпеки та охорона праці

Згідно з «Правилами техніки безпеки в лабораторіях інформатики» студентам забороняється:

- з'являтися та знаходитись приміщенні в нетверезому стані;
- ставити поруч з клавіатурою ємності з рідиною;
- перебувати в приміщенні в верхній одежі та завалювати нею робочі столи та стільці;
- працювати в лабораторії більше 6-ти годин на день (для вагітних жінок – більше 4-х годин);
- за власною ініціативою змінювати закріплені за ними робочі місця та знаходитись в приміщенні під час роботи іншої учбової групи;
- самостійно виконувати вмикання електроживлення лабораторії та заміну складових частин ПК, що вийшли із ладу.

У випадку виявлення несправностей обчислювальної техніки студент повинен сповістити про це викладача чи будь-кого з навчально-допоміжного персоналу лабораторії.

Теоретичні відомості

LabVIEW або Laboratory Virtual Instrument Engineering Workbench переводиться як середовище розробки лабораторних віртуальних приладів. Являється середовищем графічного програмування, яке широко використовується в промисловості, освіті і науково-дослідних лабораторіях в якості стандартного інструменту для збору даних і управління приладами. LabVIEW - потужне і гнучке програмне середовище, яке вживають для проведення вимірів і аналізу отриманих даних. LabVIEW - багатоплатформене середовище: ви можете використати її на комп'ютерах з операційними системами Windows, MacOS, Linux, Solaris і HP - UX. Персональні комп'ютери є гнучкішими інструментами, ніж традиційні вимірювальні прилади, тому створення власної програми на LabVIEW, або віртуального приладу (ВП), є досить нескладною справою, а інтуїтивно зрозумілий, призначений для користувача інтерфейс в середовищі LabVIEW, робить розробку програм і їх застосування дуже цікавою і захоплюючою.

Концепція LabVIEW сильно відрізняється від послідовної природи традиційних мов програмування, надаючи розробнику легку у використанні графічну оболонку, яка включає в себе весь набір інструментів, необхідних для збору даних, їх аналізу та представлення отриманих результатів. За допомогою графічної мови програмування LabVIEW, іменованого G (Джі), ви можете програмувати вашу задачу з графічної блок-діаграми, яка компілює алгоритм в машинний код. Будучи чудовим програмним середовищем для незліченних застосувань в області науки і техніки, LabVIEW допоможе вам вирішувати завдання різного типу, витрачаючи значно менше часу і зусиль у порівнянні з написанням традиційного програмного коду.

LabVIEW знаходить застосування в найрізноманітніших сферах людської діяльності. У відповідності зі своєю назвою він спочатку використовувався в дослідницьких лабораторіях, та і в даний час є найбільш популярним програмним пакетом як в лабораторіях фундаментальної науки, так і в галузевих промислових лабораторіях. Все більш широке застосування LabVIEW знаходить в освіті - в університетських лабораторних практикумах - особливо з предметів електротехніки, механіки та фізики.

Поширення LabVIEW за межами лабораторій пішло по всіх напрямках: вгору (на борту космічних апаратів), вниз (на підводних човнах) і по горизонталі (від бурових установок в північному морі до промислових підприємств у новій Зеландії). У зв'язку із зростанням можливостей Internet сфера застосування LabVIEW стала розширюватися не тільки в географічному, а й у віртуальному просторі (cyberspace). Все більше число розробників створює віртуальні прилади, що допускають віддалене управління і спостереження через Internet. Вимірювальні системи на основі віртуальних приладів відрізняються своєю багатофункціональністю, гнучкістю і низькою вартістю як з точки зору обладнання, так і з точки зору витрат часу на розробку.

Мабуть, найкращим способом пояснити причини такого широкого (можна сказати, лавиноподібного) поширення пакета LabVIEW буде узагальнення способів його використання. У всіх видах людської діяльності існують області, де не обійтися без певних видів вимірювань - дуже часто це температурні вимірювання, наприклад в печах, холодильниках, парниках, технологічних приміщеннях і навіть ... в каструлі з супом. Крім температури, часто вимірюють тиск, силу, просторове зміщення, механічне напруження, рН і т.д. - Список величезний! Зараз персональні комп'ютери проникли практично в усі сфери життєдіяльності. LabVIEW прискорює впровадження комп'ютера в вимірювальні системи - і не тільки тому, що полегшує проведення вимірювань, він також дає можливість проаналізувати виміряні величини, відобразити їх на графіках і в звітах і при бажанні опублікувати.

Після вимірювання та аналізу будь-якої величини наступним логічним шагом часто є управління, тобто зміна певних параметрів залежно від отриманих результатів. Наприклад, вимірявши температуру об'єкта, можна включити пристрій для його охолодження або

нагрівання. І знову LabVIEW значно полегшує вирішення цього завдання: моніторинг і управління процесами є основними функціями цього програмного продукту. Управління процесами може бути прямим або здійснюється через спеціальні програмовані логічні контролери (programmablelogicalcontrollers - plc), що прийнято називати диспетчерським управлінням і збором даних (supervisorycontrolanddataacquisition - scada).

Що таке LabVIEW

LabVIEW - це середовище програмування, за допомогою якої ви можете створювати додатки, використовуючи графічне представлення всіх елементів алгоритму, що відрізняє її від звичайних мов програмування, таких як C, C++ чи Java, де програмують, використовуючи текст. Однак LabVIEW являє собою значно більше, ніж просто алгоритмічний мову. Це середовище розробки та виконання додатків, призначена для дослідників - науковців та інженерів, для яких програмування є лише частиною роботи.

Потужний графічний мову програмування LabVIEW дозволяє в сотні разів збільшити продуктивність праці. Створення закінченого додатка за допомогою звичайних мов програмування може відняти дуже багато часу - тиждень або місяці, тоді як з LabVIEW потрібно лише кілька годин, оскільки пакет спеціально розроблений для програмування різних вимірів, аналізу даних і оформлення результатів. Так як LabVIEW має гнучкий графічний інтерфейс і простий для програмування, він також відмінно підходить для моделювання процесів, презентації ідей, створення додатків загального характеру і просто для навчання сучасного програмування.

Розробка додатків в середовищі LabVIEW відрізняється від роботи в середовищах на основі C або Java однією дуже важливою особливістю. Якщо в традиційних алгоритмічних мовах програмування засноване на введенні текстових команд, послідовно утворюють програмний код, в LabVIEW використовується мова графічного програмування, де алгоритм створюється в графічній іконній формі (pictorialform), що утворює, так звану блок-діаграму (block-diagram), що дозволяє виключити безліч синтаксичних деталей. Застосовуючи цей метод, ви можете сконцентрувати увагу лише на програмуванні потоку даних; спрощений синтаксис тепер не відволікає вас від аналізу самого алгоритму.

У LabVIEW використовується термінологія, малюнки іконок і основні ідеї, знайомі вченим і інженерам. Ця мова базується на графічних символах, а не на тексті для опису програмованих дій. Основоположний для LabVIEW принцип потоку даних (dataflow), згідно з яким функції виконуються лише тоді, коли вони отримують на вхід необхідні дані, однозначно визначає порядок виконання алгоритму.

Лабораторна робота №1

Запуск LabVIEW

Мета - навчитися запускати LabVIEW і аналізувати найпростіші приклади.

Програми в LabVIEW називаються *віртуальними приладами* (ВП, virtual instruments - VI), так як вони функціонально і зовні подібні реальним (традиційним) приборам. Віртуальний прилад складається з трьох основних частин:

- **лицьова панель** (FrontPanel) являє собою інтерактивний користувальницький інтерфейс віртуального приладу і названа так тому, що імітує лицьову панель традиційного приладу. На ній можуть знаходитися ручки управління, кнопки, графічні індикатори та інші **елементи управління** (controls), які є засобами введення даних з боку користувача, і **елементи індикації** (indicators) - вихідні дані з програми. Користувач вводить дані, використовуючи мишу і клавіатуру, а потім бачить результати дії програми на екрані монітора;

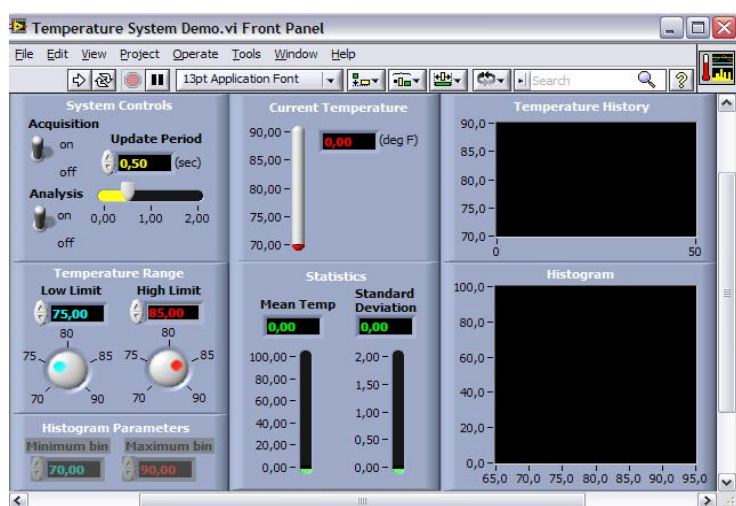


Рис. 1.1 Лицьова панель

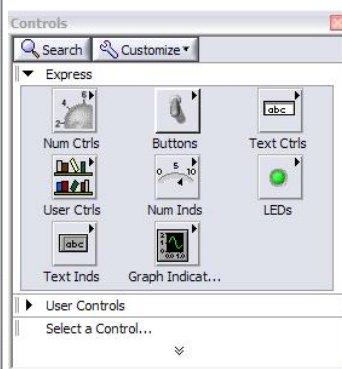


Рис. 1.2 Палітра елементів управління

Лицьова панель - інтерфейс користувача створюється з використанням палітри Елементів (Controls).

Ці елементи можуть бути або засобами введення даних - елементи управління, або засобами відображення даних - елементи відображення.

Елементи управління - кнопки, перемикачі, повзунки та інші елементи введення.
Елементи відображення - графіки, цифрові табло, світлодіоди і т.д.

- **блок-діаграма** (BlockDiagram) є вихідним програмним кодом ВП, створеним на мові графічного програмування LabVIEW, G (Джі). Блок-діаграма являє собою реально виконувану додаток. Компонентами блок-діаграми є: *віртуальні прилади нижчого рівня, вбудовані функції LabVIEW, константи та структури управління виконанням програми.* Для того щоб задати потік даних між певними об'єктами або, що те ж саме, *створити зв'язок* між ними, ви повинні намалювати відповідні *провідники* (wires). Об'єкти на лицьовій панелі представлені на блок-діаграмі у вигляді відповідних *терміналів* (terminals), через які дані можуть надходити від користувача в програму і назад;

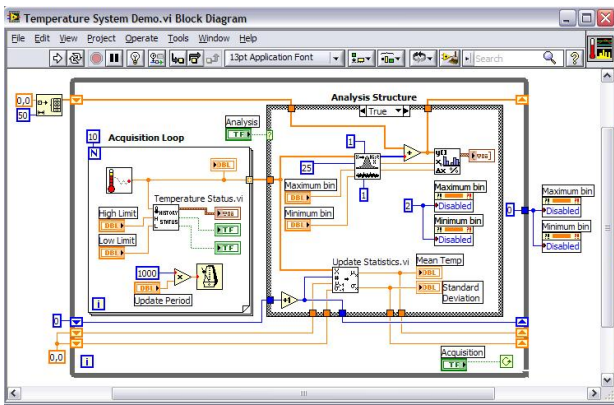


Рис. 1.3. Блок-діаграма

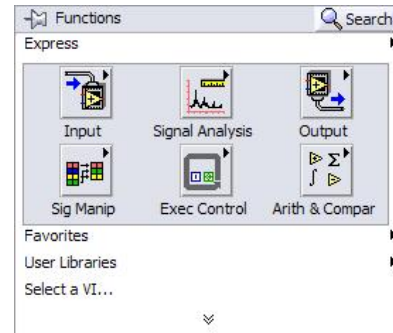


Рис. 1.4 Палітра функцій

Програмування на блок-діаграмі ВП здійснюється з використанням палітри Функцій (Functions), яка включає графічне представлення функцій для управління об'єктами на лицьовій панелі.

- для того щоб використовувати деякий ВП в якості підпрограми (підприбору) в блок-діаграмі іншого ВП, необхідно визначити його *іконку* (icon) і *сполучну панель* (connector). Віртуальний прилад, який застосовується всередині іншого ВП, називається віртуальним підприбором (ВПП, SUBVI), який є аналогом підпрограми традиційних алгоритмічних мовах. Іконка є однозначним графічним представленням ВП і може використовуватися в якості об'єкта на блок-діаграмі іншого ВП. Сполучна панель являє собою механізм передачі даних в ВП з іншої блок-діаграми, коли він застосовується в якості підприбору - ВПП. Подібно аргументам і параметрам підпрограми, сполучна панель визначає вхідні і вихідні дані віртуального приладу.

Для обох панелей доступна панель ToolsPalette (рис. 1.5), що включає набір керуючих кнопок для зміни режиму редагування.



Рис.1.5 Палітра інструментів

Вправа 1.1. Демонстрація вимірювання температури

1. Запустіть LabVIEW. У меню Help виберіть Findexamples (пошуковик прикладів). На вкладці Browse виберіть пункт IndustryApplications -Analysis. Подвійним клацанням відкрийте приклад TemperatureSystemDemo, (демонстрація вимірювання температури) показаний на рис. 1.1

2. Запустіть ВП, клацнувши по кнопці Запуск (Run) на інструментальній лінійці ВП. Кнопка поміняє зовнішній вигляд, вказуючи, що ВП виконується. Інші кнопки на інструментальній лінійці також можуть змінити свій зовнішній вигляд або зникнути, тому що деякі функції доступні тільки під час роботи ВП (наприклад, кнопка Перервати - Abort), а інші в цей час незастосовні (функції редагування). Кнопка Abort на інструментальній лінійці стає активною. Її можна натиснути, щоб екстрено перервати виконання ВП. Прилад Температурі Систем Demo імітує додаток відстеження (моніторингу) температури. Віртуальний прилад вимірює температуру і показує миттєві відліки на температурному індикаторі, а тимчасову залежність - на графіку. Хоча відліки температури в цьому прикладі

генеруються випадковим чином, ви легко можете змінити програму і вимірювати реальну температуру. Повзунковий регулятор Періоду опитування (UpdatePeriod) задає частоту отримання нових відліків температури. ВП також показує верхній і нижній межі температур на графіку; ви можете змінити ці межі, використовуючи ручку Діапазон температур (TemperatureRange). Якщо поточна величина температури виходить за встановлені межі, то поряд з термометром загоряються світлодіоди. Віртуальний прилад працює до тих пір, поки ви не клацніть мишею по перемикачу Опитування (Acquisition), перевівши його в стан вимкнено (off). Ви також можете вмикати та вимикати режим аналізу даних. Секція Статистика (Statistics) показує поточний середнє і стандартне відхилення, а на графічному індикаторі Гістограма (Histogram) викреслюється частота появи кожного значення температури.

3. Використовуючи курсор, який під час роботи віртуального приладу працює як інструмент управління (палець), для того, щоб змінити величини верхнього і нижнього температурних меж. Виділіть колишнє значення подвійним клацанням миші, потім введіть нове значення і клацніть на кнопці введення даних, яка розташована поряд з кнопкою запуску на лінійці інструментів. Можна також, утримуючи кнопку миші, перемістити ручку управління в потрібне значення.

4. Змініть значення, що задається повзунковим регулятором Період опитування (UpdatePeriod). Наведіть покажчик інструменту управління на повзунок, натисніть кнопку миші і, не відпускаючи, перемістіть повзунок в нове положення. Встановити значення повзункового елемента управління за допомогою інструменту управління можна, клацнувши по вибраній точці на відлікової лінійці для миттєвого переміщення повзунка в це положення або клацаючи мишею по кнопках прокручування для послідовного зміни значення (значення збільшується, якщо клацати по кнопці прокрутки стрілка вгору, і зменшувати, якщо клацати по стрілці вниз). З тією ж метою можна клацнути мишею в полі цифрового індикатора (digitaldisplay) даного регулятора і ввести потрібне число з клавіатури.

5. Спробуйте регулювати і інші елементи управління подібним чином.

6. Зупиніть ВП клацанням по перемикачу Опитування (Acquisition).

Дослідження блок-діаграми

Блок-діаграма (рис. 1.2) - представляє собою завершений додаток LabVIEW. В даний момент не обов'язково розуміти призначення і взаємодію всіх елементів блок-діаграми - завдання полягає в ознайомленні з її структурою.

1. Відкрийте блок-діаграму, вибравши пункт Показати діаграму (ShowDiagram) з меню Вікно (Window).

2. Вивчіть різні об'єкти у вікні діаграми. Всі вони будуть пояснені крок за кроком при подальшому вивченні.

3. Використовуючи пункт Показати контекстну довідку (ShowContextHelp) з меню Довідка (Help), відкрийте плаваюче вікно контекстно-орієнтованої довідки. Подивіться, як змінюється вікно контекстної довідки при наведенні курсору на різні об'єкти блок-діаграми. Якщо об'єкт є вбудованою функцією LabVIEW або ВВП, в цьому вікні з'явиться також опис її вхідних і вихідних контактів.

4. Увімкніть режим підсвічування виконання: кнопка з лампочкою HighlightExecution на інструментальній лінійці. У цьому режимі можна побачити, як рухаються по провідникам дані (маленькі бульбашки) під час роботи програми.

Вправа 1.2.

У цій вправі Ви створите ВП, який генерує сигнал і виводить цей сигнал на графічний індикатор (графік осцилограм). Після того, як ви завершите вправу, лицьова панель віртуального приладу повинна виглядати так, як показано на малюнку 1.6.

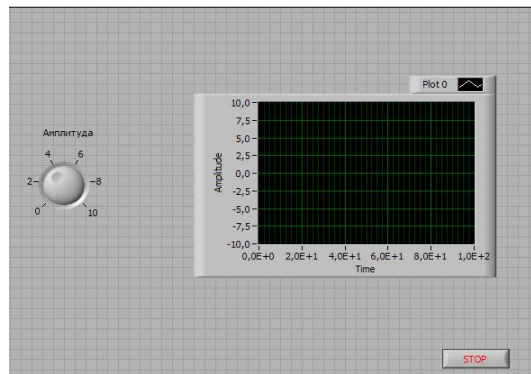


Рис.1.6 Лицьова панель ВП Отримання сигналу

Створення нового ВП з шаблону

У LabVIEW містяться готові шаблони ВП, що містять віртуальні подприбори (ВПП), функції, структури та об'єкти лицьовій панелі, які можуть знадобитися при створенні базових додатків для виконання різних вимірювань.

Виконайте наступні кроки для створення ВП, який генерує сигнал і відображає його на лицьовій панелі.

1. У вікні GettingStarted клацніть по посиланню VI from Template (ВП з Шаблону) щоб з'явилася діалогове вікно New.

2. Зі списку CreateNew виберіть VI - »FromTemplate -» Tutorial (GettingStarted) - »GenerateandDisplay. Цей шаблон призначений для створення ВП, який генерує і відображає сигнал.

3. Клацніть ОК, щоб створити ВП з шаблону. Також для цього можна двічі клацнути за назвою шаблону ВП в списку CreateNew.

4. Вивчіть лицьову панель ВП. Інтерфейс користувача, або лицьова панель, має сірий фон робочого простору і включає елементи керування та індикатори. Рядок заголовка лицьовій панелі вказує, що це вікно - лицьова панель для ВП Генерація та Відображення.

6. Виберіть меню Window-»ShowBlockDiagram (Вікно» Показати блок-діаграму) і вивчіть блок-діаграму ВП. Блок-діаграма має білий фон робочого простору і містить ВП і структури, які управляють об'єктами лицьовій панелі. Рядок заголовка блок-діаграми вказує, що це вікно - блок-діаграма для ВП Генерація та Відображення.

7. На панелі інструментів інтерфейсу користувача клацніть кнопку Run (Пуск), показану зліва. На графічному індикаторі лицьовій панелі відобразиться синусоїда.

8. Клацніть кнопку "Стоп" (показана ліворуч) на лицьовій панелі, щоб зупинити ВП.

Додавання елемента керування на Лицьову панель

Елементи управління лицьовій панелі моделюють пристрої введення на фізичному приладі і забезпечують даними блок-діаграму ВП. Більшість фізичних приладів мають ручки управління, повертаючи які можна міняти входні значення.

Щоб додати ручку управління, необхідно виконати наступні кроки:

1. Якщо палітра елементів, показана на малюнку 4, що не видна на лицьовій панелі, виберіть меню View ->ControlsPalette.

2. Якщо ви вперше запускаєте LabVIEW, палітра елементів управління за замовчуванням з'явиться з відкритою підпалітрою Express. Якщо цю підпалітру не видно, клацніть Express в палітрі елементів управління.

3. Переміщайте курсор миші по значках на Express, щоб знайти підпалітру числових елементів управління Numeric. Коли ви переміщайте курсор по значках в палітрі елементів управління, назва підпалітри, елементу управління / індикатора з'являється в полі підказки під значком.

4. Клацніть по значку NumericControls, щоб відобразити підпалітру числових елементів управління / індикаторів.

5. Клацніть по обертової ручці (Knob) в палітрі Numeric, щоб прикріпити елемент управління до курсору, потім помістіть ручку на лицьову панель, ліворуч від графіка. Ця ручка буде використовуватися в наступній вправі для управління амплітудою сигналу.

6. Виберіть меню File »SaveAs (Файл) Зберегти Як) і збережіть ВП як Отримання сигнала.vi у заздалегідь визначеному місці.

Зміна типу сигналу

На блок-діаграмі присутня іконка блакитного кольору, з назвою SimulateSignal. Ця іконка представляє експрес ВП SimulateSignal. За замовчуванням експрес ВП SimulateSignal моделює синусоїду.

Виконайте наступні кроки, щоб змінити цей сигнал на пилоподібний.

1. Перейдіть на блок-діаграму, натиснувши комбінацію клавіш <Ctrl+E> або клацнувши кнопкою миші по блок-діаграмі. Знайдіть на ній експрес ВП SimulateSignal, показаний зліва. Експрес ВП-такий елемент блок-діаграми, який ви можете конфігурувати для виконання стандартних вимірювальних завдань. А саме, експрес ВП SimulateSignal моделює сигнал, на підставі тієї конфігурації, яку ви задасте.

2. Клацніть правою кнопкою миші на експрес ВП SimulateSignal і виберете Properties (Властивості) з контекстного меню, щоб відобразити діалогове вікно ConfigureSimulateSignal (настройка моделюємого сигналу).

Щоб з'явилось діалогове вікно ConfigureSimulateSignal (настройка моделюємого сигналу), можна також двічі натиснути правою кнопкою миші по експрес ВП SimulateSignal.

Якщо під'єднати провідники з вхідними значеннями до експрес ВП SimulateSignal і запустити його, експрес ВП покаже реальні дані в діалоговому вікні налаштування. Якщо закрити і знову відкрити експрес ВП, він буде показувати приклад даних безвідносно вхідних значень в діалоговому вікні налаштування до тих пір, поки СП не буде знову запущений.

3. Виберіть Sawtooth (пилоподібний сигнал) з випадаючого меню Signaltype (тип сигналу). Форма сигналу на графіку, в розділі попереднього перегляду результату, зміниться на пилоподібну. Діалогове вікно ConfigureSimulateSignal (настройка сигналу) повинно виглядати так само, як на малюнку 1.7.

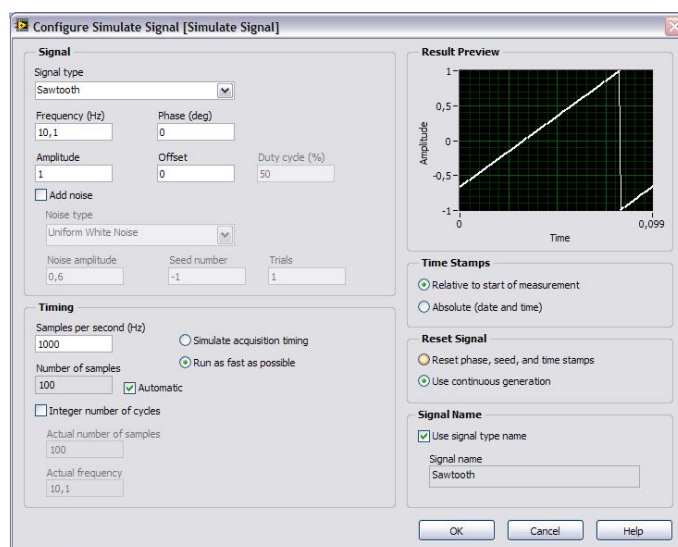


Рис. 1.7. Діалогове вікно ConfigureSimulate

4. Натисніть кнопку ОК, щоб зберегти поточні налаштування і закрити діалогове вікно ConfigureSimulateSignal.

5. Наведіть курсор миші на подвійні стрілки внизу іконки експрес ВП SimulateSignal. Стрілки показують, що ВП має приховані вхідні і вихідні термінали, які можна зробити видимими, якщо розтягнути кордон ВП.

6. Коли з'явиться двонаправлена стрілка, показана ліворуч, перемістіть кордон експрес ВП так, щоб додалися два рядки. Після виконання даної операції з'явиться вхід Amplitude (амплітуда).

Тепер у вас з'явилася можливість регулювати амплітуду пилоподібного сигналу.

Зверніть увагу, що амплітуда - це опція діалогового вікна ConfigureSimulateSignal (малюнок 1.7). Коли входи, наприклад, амплітуда, присутні і на блок-діаграмі і в діалоговому вікні настройки, їх значення можна змінювати в будь-якому місці.

З'єднання об'єктів на блок-діаграмі

Щоб використовувати обертову ручку для зміни амплітуди сигналу, необхідно з'єднати її з входом Amplitude (Амплітуда) експрес ВП SimulateSignal.

Для цього виконайте наступні кроки:

1. На блок-діаграмі наведіть курсор на термінал елемента управління Knob (Обертається ручка), показаний зліва і позначає цикл. Курсор миші стане стрілкою (інструментом Позиціонування), як показано зліва. Цей інструмент використовується для вибору, розміщення або зміни розмірів об'єктів.

2. Виділіть термінал Knob (Обертається ручка) за допомогою інструменту Позиціонування. Переконайтеся, що він знаходиться зліва від експрес ВП SimulateSignal і всередині сірого контуру, показаного зліва. Термінали всередині циклу є представленням елементів управління, розташованих на лицьовій панелі. Термінали - точки вводу / виводу, через які відбувається обмін інформацією між лицьовою панеллю і блок-діаграмою.

3. Зніміть виділення з терміналу Knob, клацнувши в порожньому місці блок-діаграми. У майбутньому, щоб використовувати інші інструменти при роботі з об'єктами, необхідно попередньо зняти виділення з об'єкта і потім переключитися на потрібний інструмент.

4. Помістіть курсор над стрілкою на терміналі Knob (Обертається ручка), як показано зліва. Курсор набуде вигляду котушки, або Wiringtool (Інструмент з'єднання). Використовуйте цей інструмент для з'єднання об'єктів на блок-діаграмі.

5. Коли з'явиться інструмент Сполуки, клацніть лівою кнопкою миші по стрілці на терміналі Knob, а потім по стрілці на вході Амплітуда експрес ВП SimulateSignal, як показано зліва, щоб з'єднати ці об'єкти разом. З'явиться провідник, що з'єднує об'єкти. Дані будуть передаватися по цьому провіднику від терміналу Knob (ручка що обертається) до експрес ВП.

6. Виберіть меню File »Save (Файл) Зберегти), щоб зберегти ВП.

Запуск ВП

Виконання коду починається з запуском ВП.

Виконайте наступні кроки, щоб запустити ВП Отримання сигналу.

1. Перейдіть на лицьову панель, клацнувши по ній.

2. Натисніть кнопку Run (Пуск) для запуску ВП.

3. Помістіть курсор над ручкою управління. Курсор набуде вигляду руки, тобто перейде в режим Operatingtool (інструмент Управління), як показано зліва. Використовуйте його, щоб міняти значення елемента керування.

4. Використовуючи Інструмент Управління, поверніть ручку, щоб скорегувати амплітуду пилоподібного сигналу. Амплітуда пилоподібного сигналу змінюється настільки, наскільки Ви повертаєте ручку. При цьому інструмент Управління показуватиме поле підказки, в якому буде відображатися поточний числове значення. Вісь Y на графіку масштабується автоматично, підлаштовуючись під зміну амплітуди. Щоб показати, що даний ВП виконується, кнопка Run (Пуск) змінюється на затемнену стрілку, яка показана зліва. Під

час виконання ВП, можна змінювати значення більшості елементів управління, але редагувати ВП можна.

5. Щоб зупинити ВП, натисніть на кнопку STOP (Стоп), яка показана зліва. Кнопка STOP зупиняє ВП після того, як він закінчить виконання поточної ітерації. Кнопка AbortExecution (перервати виконання), яка показана ліворуч, зупиняє ВП негайно, до закінчення поточної ітерації. Переривання роботи ВП, що використовує зовнішні ресурси, наприклад зовнішні апаратні засоби, може залишити ресурси в невідомому стані, не скинувши або не звільнять їх належним чином. Забезпечите ВП, який ви створюєте, кнопкою зупинки, щоб обійти цю проблему.

Варіанти завдань:

Варіант 1

1. Створити віртуальний прилад об'єднуючий елемент управління "Numeric" і всі наявні елементи відображення;
2. Створити віртуальний прилад об'єднуючий елемент управління "Dial" і всі наявні елементи відображення.

Варіант 2

1. Створити віртуальний прилад об'єднуючий елемент управління "PointerSlide" і всі наявні елементи відображення;
2. Створити віртуальний прилад об'єднуючий елемент управління "Knob" і всі наявні елементи відображення;

Варіант 3

1. Створити віртуальний прилад об'єднуючий елемент управління "Knob" і всі наявні елементи відображення;
2. Створити віртуальний прилад об'єднуючий елемент управління "Dial" і всі наявні елементи відображення;

Варіант 4

1. Створити віртуальний прилад об'єднуючий елемент управління "FillSlide" і всі наявні елементи відображення;
2. Створити віртуальний прилад об'єднуючий елемент управління "Numeric" і всі наявні елементи відображення;

Варіант 5

1. Створити віртуальний прилад об'єднуючий елемент управління "PointerSlide" і всі наявні елементи відображення;
2. Створити віртуальний прилад об'єднуючий елемент управління "Knob" і всі наявні елементи відображення;

Варіант 6

1. Створити віртуальний прилад об'єднуючий елемент управління "Numeric" і всі наявні елементи відображення;
2. Створити віртуальний прилад об'єднуючий елемент управління "FillSlide" і всі наявні елементи відображення;

Варіант 7

1. Створити віртуальний прилад об'єднуючий елемент управління "FillSlide" і всі наявні елементи відображення;
2. Створити віртуальний прилад об'єднуючий елемент управління "Dial" і всі наявні елементи відображення;

Варіант 8

1. Створити віртуальний прилад об'єднуючий елемент управління "PointerSlide" і всі наявні елементи відображення;
2. Створити віртуальний прилад об'єднуючий елемент управління "Numeric" і всі наявні елементи відображення;

Контрольні питання:

1. З яких основних компонентів складається Ваш ВП?
2. З яких підпалітр складається палітра Controls?
3. З яких підпалітр складається палітра Functions?
4. Як здійснюється пуск розробленого ВП?
5. Назвіть призначення кнопок на блок-діаграмі.
6. Назвіть призначення кнопок на лицьовій панелі.
7. Що таке елемент відображення і елемент управління?
8. Як ввести число в елемент управління?

Лабораторна робота №2

Середовище проектування LabVIEW

Мета - вивчити, як три складові віртуального приладу - лицьова панель, блок-діаграма та іконка / сполучна панель працюють разом.

Завдання

- Навчитися користуватися лицьовою панеллю, блок-діаграмою і іконкою / сполучної панеллю.
- Побачити різницю між елементами управління і індикаторами.
- Навчитися розпізнавати на блок-діаграмі термінали елементів управління і індикаторів.
- Зрозуміти ідею програмування потоку даних.
- Познайомитися з системою меню.
- Навчитися користуватися панеллю інструментів, палітрою інструментів, палітрою елементів управління і індикації, палітрою функцій і підпалітрами.
- Виконати вправи.

Вправа 2.1.

Створимо простий ВП, зображений на рис. 2.1, який обчислює суму двох чисел. Його блок-діаграма містить приклади терміналів, вузлів і провідників даних.

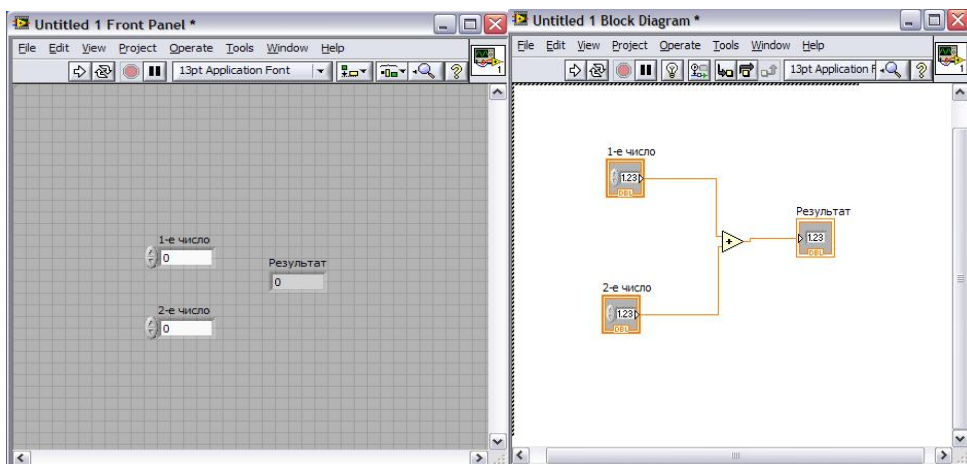


Рис. 2.1. Лицьова панель Add.vi

Рис. 2.2. Блок-діаграма Add.vi

Вікно блок-діаграми містить вихідний графічний код ВП. Блок-діаграма LabVIEW відповідає рядкам тексту звичайної мови програмування начебто C або Basic - це такий же реально виконуваний код. Конструювання блок-діаграми здійснюється шляхом з'єднання між собою об'єктів, виконують певні функції. Розглянемо різні компоненти блок-діаграми: термінали (terminals), вузли (nodes), провідники даних (wires).

1. Запустіть LabVIEW.
2. У вікні GettingStarted клацніть по посиланню Blank VI (Новий ВП), перед Вами з'явиться Лицьова панель з назвою Untitled 1.
3. Помістіть на неї 2 елементи управління (NumericControls) і 1 елемент відображення (NumericIndicator). Використовуйте для цього палітру елементів управління (Controls). Коли поміщається елемент керування або індикатор на лицьову панель, LabVIEW автоматично створює на блок-діаграмі відповідний термінал.

4. Перейдіть в блок-діаграму шляхом вибору пункту Показати блок-діаграму (ShowDiagram) з меню Вікно. Ви виявите на ній два термінали - рис.2.2.

За замовчуванням не можна видалити з блок-діаграми термінал, який відповідає елементу управління або індикатору, хоча можна спробувати це зробити. Термінал зникає лише тоді, коли ви видалите відповідний елемент керування або індикатор з лицьовій панелі.

Рамка терміналів елементів управління виділена жирною лінією, в той час як межа терміналів індикаторів позначена тонкою лінією - рис.2.3. Дуже важливо не плутати ці два типи терміналів, оскільки вони функціонально абсолютно різні (*керуючий елемент = введення, індикатор = висновок, тому не можна замінити один іншим*).



Рис. 2.3. Блок-діаграма з терміналом елемента управління (введення даних), терміналом елемента відображення (висновок даних) і провідником для передачі даних

Ви можете розглядати термінали як порти введення і виведення блок-діаграми або як джерела та приймачі даних. Дані, які ви вводите в елемент управління NumericControl 1 (див. Рис. 2.3), виходять з лицьової панелі і надходять у блок діаграму через термінал NumericControl 1. Потім дані з терміналу NumericControl 1 по провіднику надходять на функції Скласти (Add) . Аналогічно дані надходять на суматор і з другого терміналу елемента управління. Після того як функція Скласти виконає обчислення, вона створить нове значення на свій вихід. Це значення надійде на термінал Це позначення (NumericIndicator) і буде виведено на лицьову панель, де його може спостерігати користувач.

5. Перейдіть до палітри і виберіть опцію Add (Functions-> Express->Arithmetic& Comparison->Numeric). Помістіть її на блок-діаграму.

6. Тепер виберіть інструмент з'єднання "котушка" з палітри Інструменти. З'єднайте всі елементи, як показано на рис. 2.2.

7. Перейдіть на лицьову панель і запустіть віртуальний прилад. Поміняйте кілька разів числа і постежте за результатом. Збережіть ВП.

Вузли даних

Вузол даних - це просто узагальнююча назва будь-якого виконуваного елемента програми. Вузли аналогічні операторам, функціям і підпрограмам в традиційних мовах програмування. Функції Скласти (Add) і Відняти (Subtract) представляють один вид вузла. Іншим видом вузла є Структура (Structure), яка може виконувати код циклічно або за умовою, точно так само, як цикли і умовні конструкції в традиційних мовах програмування. LabVIEW містить і спеціальні типи вузлів, наприклад, вузол Формула (FormulaNode), призначений для роботи зі складними математичними формулами і виразами. Відзначимо певний тип вузлів, EventStructure (Структура подій), який дозволяє реагувати на дії лицьовій панелі і блок-діаграми.

У середовищі LabVIEW провідники даних використовуються для з'єднання численних терміналів даних. Поля введення / виводу повинні бути сумісними з типами даних, що передаються їм по провідниках. Наприклад, не можна з'єднувати поле виведення масиву з полем введення даних чисельного типу. Крім того, характер з'єднання повинен бути коректним. Провідники повинні бути приєднані лише до одного джерела даних і, принаймні, до одного полю введення даних. Наприклад, не можна з'єднувати два елементи відображення. Компонентами, визначальними сумісність сполуки, є: тип даних елемента управління та / або відображення і тип даних поля вводу / виводу.

Ви можете розглядати термінали як порти введення і виведення блок-діаграми або як джерела та приймачі даних. Дані, які ви вводите в елемент управління NumericControl 1 (див. Рис. 2.3), виходять з лицьової панелі і надходять у блок діаграму через термінал NumericControl 1. Потім дані з терміналу NumericControl 1 по провіднику надходять на функції Скласти (Add) . Аналогічно дані надходять на суматор і з другого терміналу елемента управління. Після того як функція Скласти виконає обчислення, вона створить нове значення на свій вихід. Це значення надійде на термінал Це позначення (NumericIndicator) і буде виведено на лицьову панель, де його може спостерігати користувач.

5. Перейдіть до палітри і виберіть опцію Add (Functions-> Express->Arithmetic& Comparison->Numeric). Помістіть її на блок-діаграму.

6. Тепер виберіть інструмент з'єднання "котушка" з палітри Інструменти. З'єднайте всі елементи, як показано на рис. 2.2.

7. Перейдіть на лицьову панель і запустіть віртуальний прилад. Поміняйте кілька разів числа і постежте за результатом. Збережіть ВП.

Типи даних.

В даному курсі використовуються такі типи даних:

- Numeric (чисельний тип);
- Floatingpoint - число з плаваючою комою
- Integer - цілочисельний тип
- Boolean - логічний тип, відображається у вигляді зелених терміналів.
- String - строковий тип, відображається у вигляді рожевих терміналів.
- Path - шлях до файлу, відображається у вигляді терміналів
- Array - масиви включають типи даних складових елементів і приймають відповідний їм колір;

• Cluster - кластери включають різні типи даних. Кластерний тип даних відображається коричневим кольором, якщо всі його елементи чисельні, якщо ж елементи кластера є даними різних типів, він відображається рожевим;

Провідники даних

Віртуальний прилад LabVIEW являє собою єдине ціле за рахунок провідників даних (wires), що з'єднують вузли і термінали. Провідники є каналами проходження даних від терміналу-джерела до одного або декількох терміналів-приймачів. Якщо ви спробуєте приєднати до провідника більш ніж одне джерело або взагалі жодного джерела, то LabVIEW "не схвалить" ваших дій, і провідник стане пошкодженим (broken). Може бути тільки одне джерело даних, але кілька приймачів.

Кожен провідник має свій стиль і колір залежно від типу даних, що проходять по ньому. У таблиці рис. 2.5 наведено кілька типів провідників і відповідні їм типи переданих даних.

	Скаляр (Scalar)	(1D Array)	(2D Array)	Цвет
Число с плавающей точкой				Оранжевый
Целое число				Синий
Логическое				Зеленый
Строка				Розовый

Рис. 2.4. Основні стилі провідників на блок-діаграмах

Програмування потоку даних - рух разом з потоком

Оскільки LabVIEW не є текстовим мовою програмування, його код не може виконуватися "рядок за рядком". Принцип, який управляє виконанням програми LabVIEW,

називається потоком даних (dataflow). Код вузла виконується тільки тоді, коли дані надходять на всі його вхідні термінали, по закінченні роботи вузол передає дані на всі вихідні термінали, і дані негайно надходять від даного джерела на термінали наступних приймачів. Принцип потоку даних сильно відрізняється від методу потоку управління (controlflow) в текстових мовах програмування, де інструкції виконуються в тій послідовності, в якій вони написані. До цієї відмінності слід звикнути. Таким чином, якщо традиційний потік управління здійснюється за допомогою інструкцій, обробка потоку даних управляється самими даними, тобто залежить від даних (datadependent).

Вправа 2.2.

1. Створіть новий ВП (Blank VI). На екрані з'явиться лицьова панель з назвою Untitled 1. Перейдіть до підпалітри Express палітри Елементи управління. У підпалітрі Графіки виберіть Розгортка осцилограми (WaveformChart). Помістіть графік на лицьову панель

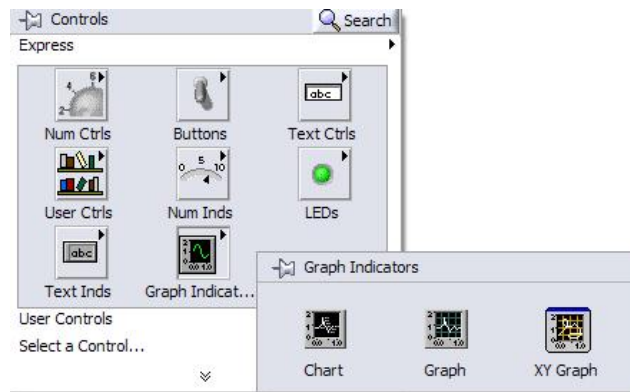


Рис. 2.3. Підпалітра графіки

2. Поверніться до палітри Елементи керування, виберіть підпалітру Modern -Boolean, потім вкажіть елемент Вертикальний перемикач (VerticalToggleSwitch). Встановіть його поруч з графічним індикатором, як показано на рис. 2.4.

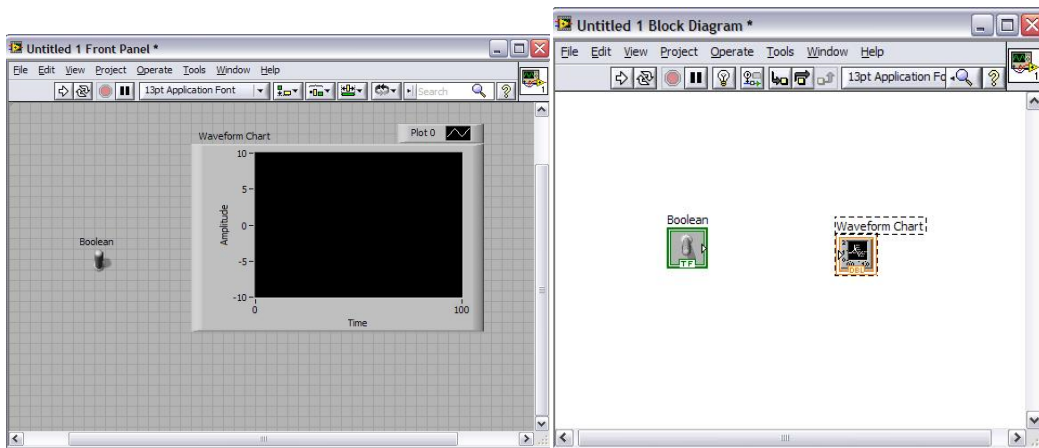


Рис. 2.4. Передня панель з графіком і перемикачем. Рис. 2.5. Термінали

3. Виберіть інструмент управління в палітрі Інструменти. Тепер змініть масштаб графіка. Виділіть число 10, двічі клацнувши по ньому інструментом управління. Надрукуйте 1.0 і клацніть по кнопці введення, яка з'явиться на панелі інструментів у верхній частині вікна.

4. Перейдіть в блок-діаграму шляхом вибору пункту Показати блок-діаграму (ShowDiagram) з меню Вікно. Ви виявите на ній два термінали - рис.2.5.

5. Тепер помістіть термінали всередину циклу за умовою, щоб забезпечити повторення виконання сегмента вищій програми. Перейдіть в підпалітру Programming->Structures (Структури) в палітрі і виберіть Цикл за умовою (WhileLoop).

Курсор змінить вигляд і перетвориться на маленьку іконку циклу. Тепер охопіть термінали DBL і TF: натисніть і утримуйте кнопку миші під час переміщення курсору від верхнього лівого кута в нижній правий кут, охоплюючи об'єкти, які хочете помістити в цикл. Після звільнення кнопки миші пунктирна лінія - слід переміщення курсору - трансформується в цикл за умовою - рис. 2.5. Зробіть поле більше, щоб усередині було деякий вільний простір.

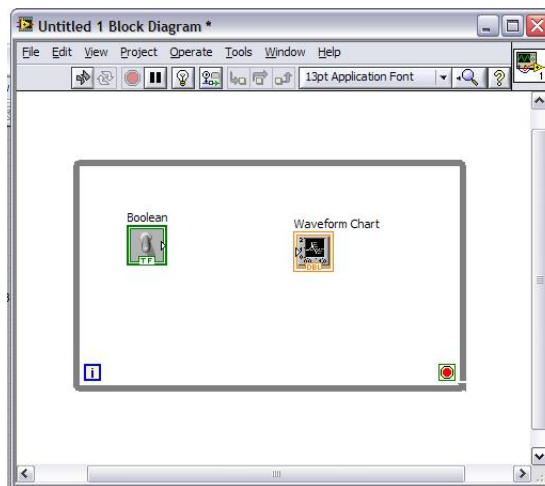


Рис. 2.5. Формування циклу за умовою

6. Перейдіть до палітри і виберіть опцію Randomnumber (0-1) в підпалітрі Programming->Numeric (Числові). Помістіть його всередину циклу за умовою. Цикл за умовою є особливою структурою Lab VIEW, яка повторює код, що знаходиться всередині його кордонів, до тих пір, поки зчитує значення БРЕХНЯ. Це свого роду еквівалент циклу Do-While в звичайній мові програмування.

7. Виберіть інструмент переміщення з палітри Інструменти і розташуйте об'єкти на вашій блок-діаграмі таким чином, щоб вони виглядали аналогічно першій блок-діаграмі рис. 2.6.

8. Тепер виберіть інструмент з'єднання "котушка" з палітри Інструменти. Клацніть мишею один раз на терміналі Випадкове число (0-1), перемістіть курсор на термінал DBL і ще раз клацніть мишею

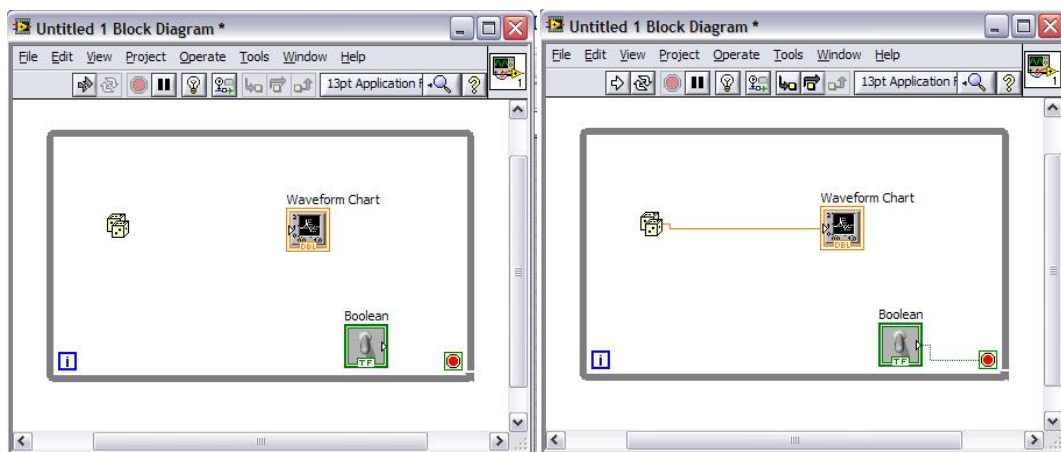


Рис. 2.6. Блок-діаграма з елементом Випадкове число і з'єднаними терміналами

9. Тепер дві іконки будуть з'єднані суцільною помаранчевою лінією, як показано на рис. 2.13. Якщо ви зробили щось неправильно, виділіть провідник або його частину за допомогою інструменту переміщення, натисніть на клавішу <Delete>, щоб видалити його. Тепер з'єднайте термінал Boolean TF з терміналом умови виходу з циклу. Цикл почне виконуватися, якщо перемикач на лицьовій панелі знаходиться в стані ІСТИНА (перемикач у верхньому положенні), і зупиниться, якщо перемикач перевести в стан БРЕХНЯ (нижнє положення).

10. Запустіть програму. Ви побачите послідовність випадкових чисел, безперервно що викреслюються на графіку. Якщо ви хочете зупинити виконання програми, натисніть тумблер.

Вправа 2.3. Основні елементи лицьовій панелі і блок-діаграми

1. Створіть новий ВП і переключіться з лицьової панелі на блок-діаграму.
2. Змініть розмір вікон так, щоб одночасно бачити лицьову панель і блок-діаграму.
3. Створіть числовий елемент управління (DigitalControl), строковий елемент управління (String) і логічний індикатор (Controls-> Modern-> Boolean) на передній панелі шляхом вибору їх з палітри Елементи управління.

Зверніть увагу на те, як під час створення об'єкта на лицьовій панелі Lab VIEW створює відповідні термінали на блок-діаграмі. Також відзначте, що числові термінали з плаваючою комою мають помаранчевий колір (цілі числа будуть синього кольору), строкові дані - рожевий колір, а логічні - зелений. Ця колірна гамма полегшує розпізнавання типу даних.

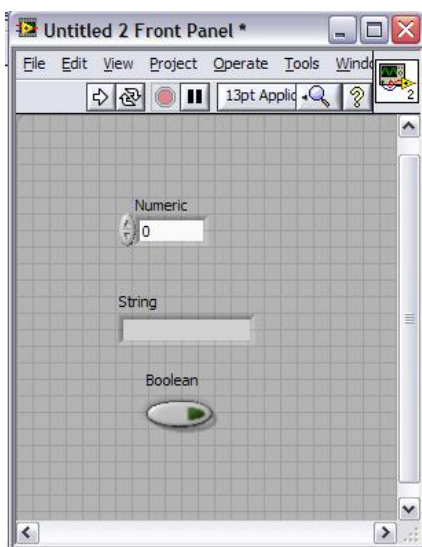


Рис. 2.7 Вибір елемента управління

4. Тепер викличте контекстне меню числового елемента управління шляхом натискання правої кнопки миші і виберіть опцію Змінити індикатор. Відзначте, як змінюється зовнішній вигляд елемента лицьовій панелі (маленькі стрілки зникають) і як змінюється термінал на блок-діаграмі (кордон індикатора набагато тонше). Перемикайте режим роботи об'єкта між управлінням і індикацією до тих пір, поки не побачите різницю - як на передній панелі, так і на блок-діаграмі. Зверніть увагу, що елементи лицьовій панелі деяких об'єктів управління і відображення (наприклад, булевих) виглядають однаково, але їх термінали на блок-діаграмі завжди будуть різними.

5. Виберіть інструмент переміщення (стрілка) з підпалітри Інструменти, потім виділіть їм будь-який об'єкт на лицьовій панелі. Натисніть клавішу <delete> для його видалення. Видаліть всі об'єкти на лицьовій панелі, щоб отримати порожню лицьову панель і порожню блок-діаграму.

6. Помістіть інший числовий елемент управління з підпалітри Числові палітри Елементи управління на лицьову панель. Після цього не робіть клацання мишею, і ви побачите маленьке виділене вікно над елементом управління. Надрукуйте Number 1, і цей текст з'явиться у вікні. Клацніть мишею по кнопці Введення панелі інструментів, щоб ввести текст. Ви тільки що створили ярлик. А тепер створіть інший числовий елемент управління з ім'ям Number 2, числовий елемент відображення з ім'ям N1 + N2 і числовий елемент відображення з ім'ям N1-N2. За допомогою інструменту управління OperationTool клацайте по стрілці елемента Number 1 поки його значення не буде рівним 2.00. Задайте Number 2 значення 3.00.

7. Поверніться до блок-діаграмі. Помістити функцію **Додавання** (Add) з підпалітри Числові палітри **Функції** на блок-діаграму (подібно створенню об'єктів на лицьовій панелі). Повторіть операцію і помістіть функцію **Віднімання** (Substract) на діаграмну панель.

Викличте контекстне меню функції **Додавання** і виберіть опцію **Видимі елементи Термінали** (раніше ця опція не була позначена, засвідчуючи про те, що термінали в даний момент невидимі). Як тільки ви їх побачите, подивіться, як розташовуються входи і виходи, а потім знову відтворіть стандартну іконку шляхом вибору Видимі елементи Термінали. На цей раз опція з'являється з міткою, вказуючи на те, що входи видимі.

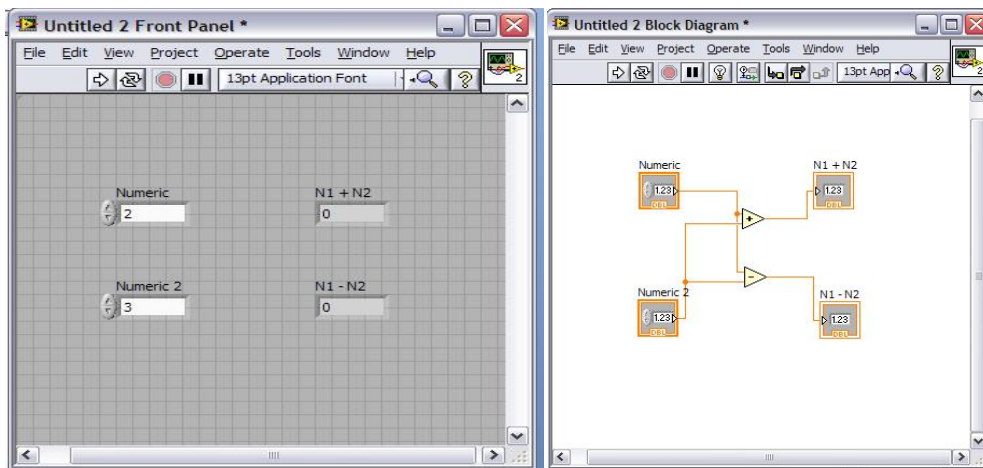


Рис. 2.8 Рис. 2.9

8. Виведіть контекстне вікно допомоги, використовуючи клавіатуру або команду Показати вікно контекстної допомоги з меню Довідка. Встановіть курсор на функції Додавання. Вікно контекстної допомоги дає вичерпну інформацію про використання даної функції і моделі з'єднання. Тепер переведіть курсор на функцію Віднімання і простежте за змінами у вікні допомоги.

9. С допомогою інструменту з'єднання підключіть термінали. Спочатку виберіть потрібний інструмент з палітри Інструменти. Потім для того, щоб намалювати лінію з'єднання, клацніть мишею один раз по терміналу DBL і один раз за відповідним терміналу функції Додавання. На екрані з'явиться суцільна лінія оранжевого кольору. Якщо ви зробите помилку і на екрані з'явилася пунктирна лінія чорного кольору, виберіть фрагмент провідника за допомогою інструменту переміщення та натисніть кнопку <delete>, потім повторіть операцію. Клацніть мишею один раз і потім опустіть її, почавши процес з'єднання, потім клацайте мишею кожен раз, коли хочете додати новий сегмент (який буде утворювати з попереднім прямий кут), і, нарешті, клацніть мишею на кінцевій точці з'єднання.

10. Запустіть ВП. Індикатор N1 + N2 покаже значення 5.00, а індикатор N1-N2 покаже значення -1. Ви можете змінити вхідні значення і знову запустити ВП.

Варіанти завдань:

Варіант 1

1. Створити віртуальний прилад для розрахунку опору по закону Ома;
2. Створити віртуальний прилад, який дозволяє проводити розрахунок величини ємнісного опору X

$$X = \frac{1}{\omega \cdot C} = \frac{1}{2\pi \cdot \nu \cdot C}$$

Варіант 2

1. Створити віртуальний прилад для розрахунку струму I за законом Ома;
2. Створити віртуальний прилад, який дозволяє проводити розрахунок величини ємнісний провідності B :

$$B = \frac{1}{X} = \omega \cdot C = 2\pi \cdot \nu \cdot C$$

Варіант 3

1. Створити віртуальний прилад для розрахунку напруги U за законом Ома.
2. Створити віртуальний прилад, який дозволяє проводити розрахунок повного опору послідовного з'єднання конденсатора і резистора Z :

$$Z = \sqrt{R^2 + X^2}$$

Варіант 4

1. Створити віртуальний прилад для розрахунку потужності струму P .
2. Створити віртуальний прилад, який дозволяє проводити розрахунок повної провідності паралельного з'єднання конденсатора і резистора Y :

$$Y = \sqrt{g^2 + B^2}, \text{ де } g = 1 / R - \text{ провідність резистора, Ом-1 або Див.}$$

Варіант 5

1. Створити віртуальний прилад для розрахунку сили Архімеда.
2. Створити віртуальний прилад, який дозволяє проводити розрахунок кінетичної енергії

$$E_k = \frac{mV^2}{2}$$

Варіант 6

1. Створити віртуальний прилад для розрахунку сили Гука
2. Створити віртуальний прилад, який дозволяє проводити розрахунок формули Ціолковського

$$v = u * \ln\left(\frac{M_0}{M}\right)$$

Варіант 7

1. Створити віртуальний прилад для розрахунку першого закону Ньютона
2. Створити віртуальний прилад, який дозволяє проводити розрахунок формули Торрічеллі:

$$v = \sqrt{2gh}$$

Варіант 8

1. Створити віртуальний прилад для розрахунку закону всесвітнього тяжіння
2. Створити віртуальний прилад, який дозволяє проводити розрахунок механічної роботи

$$A = F * s * \cos \alpha$$

Контрольні питання:

1. Що таке термінали даних і які функції вони виконують?
2. Що таке провідник даних?
3. Що таке вузол даних?
4. Як відбувається рух даних по лицьовій панелі і блок-діаграмі?
5. Як з'єднуються об'єкти провідниками даних?
6. Які типи даних Ви знаєте?

Лабораторна робота №3

Створення, редагування та налагодження ВП

Мета - навчитися використовувати різні типи даних і створювати, змінювати, з'єднувати і запускати власні ВП

Завдання

- Оволодіти технікою редагування в LabVIEW
- Вивчити різні типи елементів управління і індикаторів і їх спеціальні опції
- Навчитися основам створення ВП, тобто з'єднанню терміналів на блок-діаграмі і редагуванню
- Створити і запустити найпростіші ВП

Створення віртуальних приладів

Розміщення об'єктів на лицьовій панелі

Створення програми зазвичай починається з розміщення елементів управління і індикаторів на лицьовій панелі. Це необхідно для того, щоб ввести ваші вхідні дані і отримати вихідні дані віртуального приладу. Якщо провести курсором по палітрі Елементи управління (Control), то у верхній частині палітри можна побачити імена подпалітри (рис. 3.1). Клацніть правою кнопкою миші і утримуйте її покажчик на одній з кнопок для доступу до відповідної подпалітри. Імена об'єктів з'являться в верхній частині подпалітри при проходженні по них курсору. Виберіть потрібний об'єкт в подпалітрі, відпустивши кнопку миші.

Тепер клацніть мишею по лицьовій панелі в тому місці, де ви хочете розмістити об'єкт, - і ось він тут (рис. 3.2).

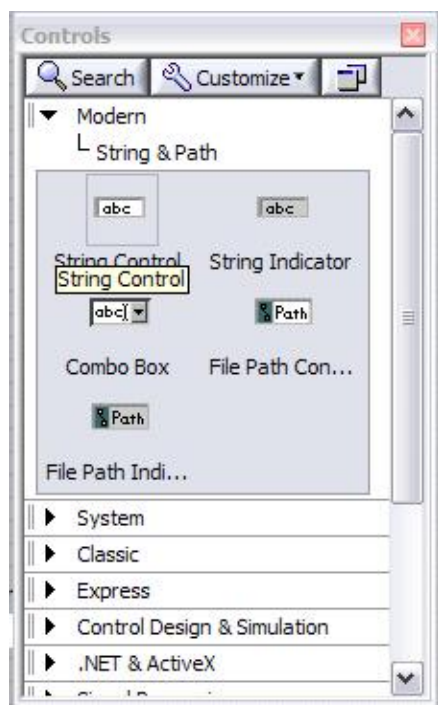


Рис. 3.1



Рис. 3.2

А тепер створіть новий ВП і помістіть числовий елемент управління (NumericControl) на лицьовій панелі. Пам'ятайте, що при розміщенні об'єкта на лицьовій панелі відповідний термінал з'являється на блок-діаграмі. Можливо, буде корисно вибрати пункт Розмістити

ліворуч і праворуч (TileLeftandRight) в меню Вікно, щоб одночасно бачити лицьову панель і вікно блок-діаграми.

Основні елементи управління та індикатори

LabVIEW має чотири типи простих елементів управління і індикаторів: **числовий (numeric)**, **логічний (Boolean)**, **строковий (string)** і рідко застосовуваний **шлях (path)**. Дещо пізніше ви зустрінете кілька більш складних типів даних, таких як масиви, групи, таблиці, діаграми і графіки.

Палітру Елементи управління можна побачити тільки тоді, коли активна лицьова панель, тому будьте уважні і не шукайте можливості користуватися палітрою Елементи управління, коли ви працюєте з панеллю блок-діаграми.

Коли вам необхідно ввести числові або текстові величини в будь-який елемент керування або індикатор, скористайтеся інструментами управління або введення тексту. Новий або змінений текст реєструється тоді, коли ви натиснете клавішу <enter> на цифровій клавіатурі, або клацніть по кнопці Введення на панелі інструментів, або клацніть мишею за межами об'єкта. Натискання на клавішу <enter> на буквено-цифровій клавіатурі викличе переклад рядка, але не реєстрацію вашого зміни. Для введення тексту в LabVIEW необхідно задіяти саме клавішу <enter> на цифровій клавіатурі.

Числові елементи керування та індикатори

Числові елементи управління дають можливість ввести числові дані в ВП, **числові індикатори** демонструють **числові значення**. LabVIEW має багато типів числових об'єктів: круглі ручки управління, повзункові перемикачі, резервуари, термометри, числовий дисплей. Щоб користуватися цими елементами, виберіть їх з підпалітри Числові палітри Елементи управління. Кожен з числових елементів може бути і елементом управління, і елементом відображення, хоча за замовчуванням для кожного об'єкта прийнятий певний вид. Наприклад, термометр (thermometer) за замовчуванням - індикатор, оскільки в майбутньому ви напевно станете користуватися тільки цим режимом його роботи. І навпаки, кнопка з'являється на лицьовій панелі в якості елемента управління, так як зазвичай режим кнопки зазвичай є інструментами введення даних (рис. 3.3).

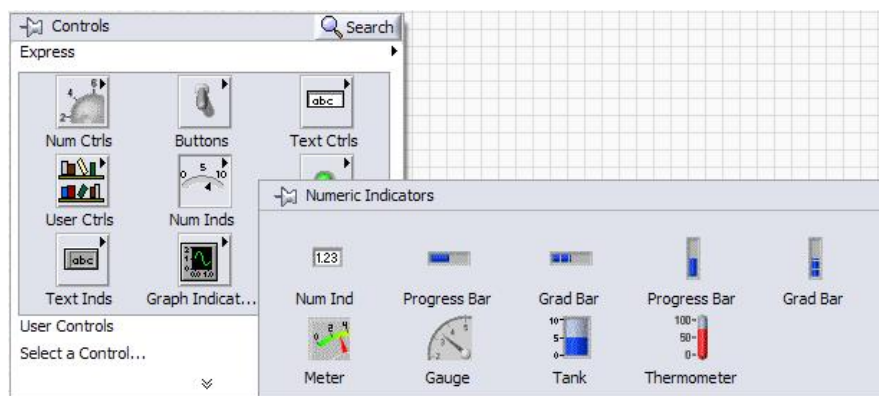


Рис. 3.3.

Представлення даних

Зовнішній вигляд терміналів числових елементів управління і індикаторів на блок-діаграмі залежить від уявлення даних. Різні типи даних передбачають різні методи їх зберігання, що допомагає використовувати пам'ять більш ефективно, так як різні уявлення числових даних можуть задіяти різну кількість байтів пам'яті. Крім цього, подання визначає, чи розглядати дані як числа зі знаком (допустимі негативні величини) і без знака (можливі тільки нульові або позитивні значення). Термінали містять кілька букв, що описують тип даних, наприклад "DBL" для чисел з плаваючою комою з подвоєною точністю.

Ви можете змінити тип представлення числових постійних, елементів управління і індикаторів, викликавши опцію Представлення (Representation). Виклик контекстного меню здійснюється клацанням по об'єкту правою кнопкою миші. Після цього ви можете зробити вибір з підпалітри, показаної на рис. 3.4.

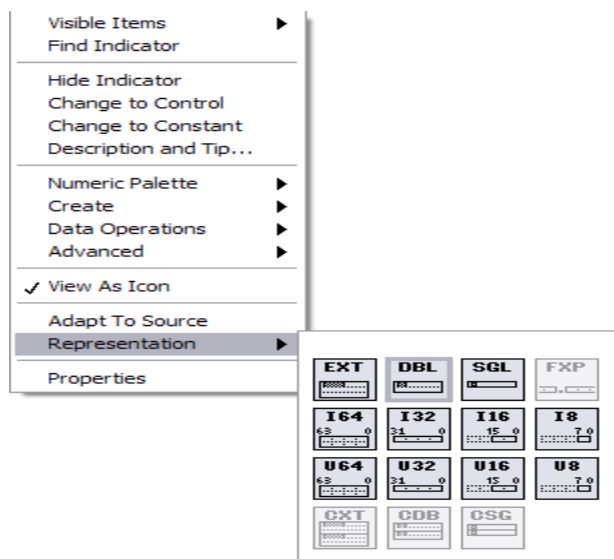


Рис. 3.4.

Логічні елементи

Під логічними значеннями ми будемо розуміти положення "on" або "off". Логічні значення можуть мати одне з двох станів: ІСТИНА (True) або БРЕХНЯ (False). LabVIEW пропонує безліч перемикачів, світлодіодів і кнопок для логічних елементів управління і індикаторів, які знаходяться в підпалітрі Логічні.

Як і числові елементи, кожен логічний елемент може бути елементом управління і індикатором, але має тип за замовчуванням, які базуються на його можливе використання (наприклад, перемикачі є елементами управління, а світлодіоди - елементами відображення).

Логічні термінали пофарбовані на блок-діаграмі в зелений колір і містять букви "TF". Термінали елементів управління мають жирні кордону, в той час як індикатори - тонкі. Дуже важливо розрізняти ці два випадки, оскільки вони функціонально не еквівалентні.

Строкові дані

Строкові елементи керування та індикатори (string) демонструють текстові дані. Рядки найчастіше містять дані у форматі ASCII, який представляє собою стандартний спосіб зберігання алфавітно-цифрових символів. Строкові термінали і провідники, по яких проходять строкові дані, пофарбовані в рожевий колір. Термінали містять букви "abc". Ви можете знайти рядки в підпалітрі Text Indicators палітри Елементи управління (рис. 3.5).

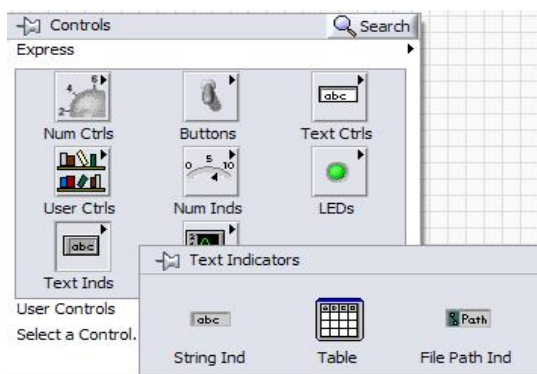


Рис. 3.5.

Хоча строкові елементи керування та індикатори можуть містити числові символи, вони не включають числових даних. Ви не можете здійснювати які-небудь числові операції над рядковими даними. Якщо вам потрібно використовувати числову інформацію, що міститься в рядку (для вчинення, наприклад, арифметичних операцій), перетворіть її в числовий формат за допомогою необхідних функцій.

Підключення

Рационально організована лицьова панель ВП, наповнена елементами управління і індикаторами, не принесе ніякої користь, поки ви не з'єднайте всі елементи на блок-діаграмі, щоб програма могла функціонувати. Користуйтеся інструментом ("котушка") для з'єднання терміналів.

Щоб з'єднати один термінал з іншим, клацніть інструментом з'єднання по першому терміналу, перемістіть інструмент на другий термінал і зробіть клацання. Не має значення, з якого терміналу ви спочатку клацаєте. Як тільки вістря курсору розташується над терміналом, область введення починає мерехтіти. Клацання кнопкою миші з'єднує провідник з цим терміналом (рис. 3.6).



Рис. 3.6.

Як тільки ви здійснили перше підключення до одного з терміналів, LabVIEW малює провідник при переміщенні курсору по блок-діаграмі, як ніби нитка змотується з котушки. При цьому вам не потрібно утримувати кнопку миші натиснутою.

Щоб почати з'єднання від вже існуючого провідника, здійсніть дії, описані вище, починаючи або закінчуючи на вже існуючому провіднику. При розташуванні котушки над провідником від починає мерехтіти.

Пошкоджені провідники

При здійсненні помилки під час з'єднання елементів з'являється пошкоджений (**broken**) провідник у вигляді чорної пунктирною лінії замість зазвичай пофарбованого провідника. До тих пір поки помилки не будуть усунуті, кнопка запуску програми залишається пошкодженою і програма не виконується. Ви можете видалити пошкоджену провідник, виділивши і видаливши його. Можна одночасно видалити несправні провідники шляхом вибору опції Видалити пошкоджені провідники (RemoveBrokenWires) в меню Редагування. Іноді несправні провідники можуть бути не більш ніж фрагментами, прихованими під іншими об'єктами, або настільки малими, що ви можете їх не побачити. У цьому випадку просто виберіть опцію Видалити пошкоджені провідники.

Якщо вам невідомо, з якої причини провідник виявився пошкодженим, то клацніть мишею по несправній кнопці запуску програми або викличте контекстне меню несправного провідника і виберіть опцію Список помилок (ListError). З'явиться діалогове вікно з описом проблеми.

Поради по з'єднанню елементів

Наступні підказки можуть трохи полегшити процес з'єднання елементів на блок-діаграмі:

- без клацання миші ви можете змінити напрямок з'єднання на 90° (зігнути його) тільки один раз,
- клацніть мишею для повороту провідника і зміни його напрямку,
- міняйте напрям виходу провідника з терміналу натисканням клавіші пробілу,

- для початку і завершення процесу з'єднання елементів двічі клацніть інструментом з'єднання по відкритій області,
- при перетині провідників з'являється невеликий зазор в першому провіднику, як ніби він знаходиться під другим провідником (рис. 3.7).

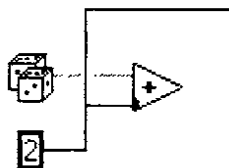


Рис. 3.7.

Для усунення цього ефекту ви можете вибрати Інструменти Опції (Options) увійти в меню Блок-діаграма (BlockDiagram), потім зазначити вікно Показувати точки в місцях перетину провідників (Showdotsatwirejunctions),

- клацніть правою кнопкою миші для видалення провідника під час процесу з'єднання,
- скористайтеся вікном контекстної допомоги для отримання більшої інформації про об'єкт і полегшення процесу з'єднання.

Створіть числові елементи управління на лицьовій панелі і підключіть їх до входів функції Додавання. Поки не підключайте вихід.

Автоматичне додавання констант, елементів управління і індикаторів

Замість того, щоб створювати константу, елемент керування або індикатор шляхом вибору їх в палітрі, а потім з'єднання їх вручну з терміналом, ви можете клацнути на вході терміналу і вибрати опцію Створити Константу (CreateConstant), Створити Елемент управління (CreateControl) або Створити Індикатор (CreateIndicator) для автоматичного створення об'єкта з відповідним типом даних для цього терміналу. Новий об'єкт підключається автоматично.

Створіть елемент відображення для демонстрації результатів операції Додавання, викликавши контекстне меню цієї функції і вибравши Створити Індикатор. LabVIEW створить термінал індикатора на блок-діаграмі, з'єданого з виходом функції Додавання, а також відповідний елемент відображення на лицьовій панелі.

Запуск віртуального приладу

Ви можете запустити ВП, вибравши команду Запуск (Run) з меню Управління або клацнувши мишею по кнопці Запуск (Run). Під час виконання програми кнопка Запуск змінює свій вигляд.

Якщо кнопка Запуск пофарбована в чорний колір і виглядає так, як ніби вона рухається, то в цей момент програма виконується на найвищому рівні.

Якщо кнопка Запуск має маленьку стрілку всередині великої стрілки, це означає, що програма виконується в якості ВПП, викликаного іншим ВП.

Якщо ви хочете, щоб програма виконувалася безперервно, натисніть кнопку Безперервний запуск (ContinuousRun), але будьте обережні: це не той прийом програмування, який слід прийняти на озброєння. Ви можете випадково загнати програму в нескінченний цикл і будете змушені перезавантажити комп'ютер, щоб вийти з цього стану. Якщо ви все-таки опинилися в такому становищі, спробуйте зробити наступне: натисніть клавішу, що запускає команду Перервати (Abort): <ctrl> + <.>.

Натисніть кнопку Перервати, щоб припинити виконання високорівневого ВП. Якщо ВП використовується більш ніж одним працюючим високорівневим ВП, то кнопка стає сіркою. Натискання на кнопку Перервати викликає негайне припинення виконання програми і не є хорошою практикою програмування, оскільки отримані дані можуть виявитися недійсними. Ви повинні передбачити можливість програмної зупинки роботи ВП, яка акуратно згорне виконання програми. Як це зробити ви дізнаєтеся пізніше.

Кнопка Пауза (Pause) тимчасово зупиняє виконання програми, а потім відновлює виконання, якщо ви її знову натиснете.

Ви можете запустити безліч програм в один і той же час. Після початку роботи ВП перейдіть у вікно лицьовій панелі або блок-діаграми наступного ВП і починайте його виконання, як описано вище. Зверніть увагу, що якщо ви виконуєте ВПП як високорівнева ВП, то всі віртуальні прилади, які його викликають як ВПП, зупиняють свою роботу до завершення його виконання. Ви не можете одночасно змусити працювати ВПП у якості високорівневого ВП і підприбору.

Підсумки

LabVIEW містить спеціальні інструменти і методики редагування, відповідні його графічній природі. Інструмент управління змінює значення об'єкта. Інструмент переміщення виділяє, видаляє і переміщує об'єкти. Інструмент з'єднання створює провідники, які з'єднують об'єкти на блок-діаграмі. Інструмент введення тексту створює і змінює власні й вільні ярлики. Власні ярлики належать певного об'єкту і не можуть бути вилучені або пересунуті незалежно, тоді як вільні ярлики не мають таких обмежень.

У LabVIEW є чотири типи елементів управління і індикаторів: числові, логічні, рядкові і шляхи. Кожен має окремий тип даних і особливі опції контекстного меню. Термінали елементів управління і індикаторів, а також провідники на блок-діаграмі володіють колірними кодами відповідно до типу даних. Термінали і провідники, що працюють з числами з плаваючою комою, пофарбовані в оранжевий колір, з цілими числами - в синій, з логічними значеннями - в зелений.

Об'єкти на лицьовій панелі або блок-діаграмі знаходяться в палітрах Елементи управління або Опції відповідно. Ви можете отримати доступ в ці палітри, клацнувши правою кнопкою миші по вільному місцю лицьовій панелі або блок-діаграмі.

Для запуску ВП клацніть мишею по кнопці Запуск або виберіть функцію Запуск (Run) в меню Управління. Якщо кнопка Запуск не працює, значить у вашому ВП щось не в порядку.

Вправа 3.1

Самостійно створіть ВП, який би порівнював два вхідних числа. Якщо вони рівні, то запалюється світлодіод на лицьовій панелі (Рис. 3.26). Назвіть його ComparisonPractice.vi.

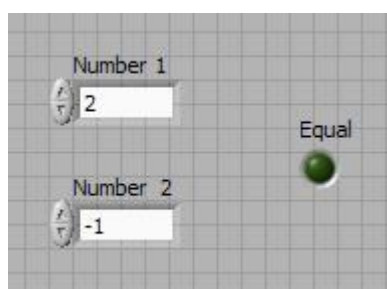


Рис. 3.8.

Вправа 3.2

Перетворення °C в °F

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

Лицьова панель

1. Виберіть пункт головного меню File »New» VI, щоб відкрити нову лицьову панель.
2. Помістіть цифровий елемент управління на лицьову панель. В поле власної мітки елемента управління надрукуйте «Град С».

3. Помістіть елемент відображення даних на лицьову панель. Він буде використаний для відображення значень температури в ° F. В поле власної мітки елемента управління надрукуйте «Град F» і клацніть мишею у вільному просторі лицьовій панелі або натисніть кнопку Enter.

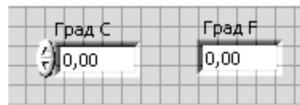


Рис. 3.9.

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

Блок-діаграма

4. Перейдіть на блок-діаграму, вибравши пункти головного меню Window »ShowDiagram.
5. Виберіть функцію Multiply (Множення) з палітри Функцій у розділі Functions »Numeric (Арифметичні функції). Помістіть її на блок-діаграму.
6. Виберіть функцію Add (Додавання) з палітри Функцій у розділі Functions »Numeric (Арифметичні функції). Помістіть її на блок-діаграму.
7. Виберіть числову константу з палітри Функцій у розділі Functions »Numeric (Арифметичні функції). Помістіть два числові константи на блок-діаграму. Після розміщення числової константи на блок-діаграмі полі введення її значень підсвічується і готово для редагування. Однією константі надайте значення 1,8, інший 32,0.
8. З'єднайте об'єкти блок-діаграми за допомогою інструменту З'ЄДНАННЯ.
9. Перейдіть на лицьову панель, вибравши в головному меню пункт Window »ShowPanel.
10. Збережіть ВП, він буде використовуватися пізніше

Запуск ВП

1. Введіть число в елемент управління і запустіть ВП:
 - а) для введення числа в елемент управління слід використовувати інструмент УПРАВЛІННЯ чи інструмент введення тексту;
 - б) натисніть кнопку Run, щоб запустити ВП;
 - в) введіть декілька різних значень температури і запустіть ВП.
2. Закрийте ВП, вибравши пункт головного меню File »Close.

Індивідуальне завдання:

Створити віртуальний прилад, який знаходить значення виразу а, b, якщо x, y, z ввести с клавіатури.

Варіант 1
$$a = \frac{\sqrt{|x-1|} - \sqrt{|y|}}{1 + x^2/2 + x^2/4}, b = x(\arctg(z) + e^{-(x+3)})$$

Варіант 2
$$a = \frac{3 + e^{y-1}}{1 + x^2|y - \text{tg}(z)|}, b = 1 + |y - x| + \frac{(y-x)^2}{2} + \frac{(y-x)^3}{3}$$

Варіант 3
$$a = (1 + y) \frac{x + y/(x^2) + 4}{e^{-x-2} + 1/(x^2 + 4)}, b = \frac{1 + \cos(y-2)}{x^4/2 + \sin^2 z}$$

$$\text{Варіант 4 } a = y + \frac{x}{y^2 + \left| \frac{x^2}{y + x^3/3} \right|}, b = (1 + \operatorname{tg}^2 \frac{z}{2})$$

$$\text{Варіант 5 } a = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2 y}, b = 1 + \frac{z^2}{3 + z^2/5}$$

$$\text{Варіант 6 } a = \frac{1 + \sin^2(x + y)}{2 + |x - 2x/(1 + x^2 y^2)|} + x, b = \cos^2(\operatorname{arctg} \frac{1}{z})$$

$$\text{Варіант 7 } a = \ln \left| (y - \sqrt{|x|}) \left(x - \frac{y}{z + x^2/4} \right) \right|, b = x - \frac{x^2}{3} + \frac{x^3}{4}$$

$$\text{Варіант 8 } a = \frac{13 + e^{y-1}}{z^2 + x^2 \left| \frac{x}{y} - \operatorname{tg}(z) \right|}, b = 4 - |z + x| + \frac{(y - x)^2}{2} + \frac{(z - x)^3}{7}$$

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. Які типи даних Ви знаєте?
7. Як з'єднуються об'єкти провідниками даних?
8. Як здійснюється пуск розробленого ВП?
9. Що таке елемент управління і елемент відображення?

Лабораторна робота №4

Створення підпрограм віртуального приладу (ВП)


Мета - вивчити підпрограму ВП; створити іконку і налаштувати сполучну панель для можливості використання ВП в якості підпрограми ВП.

Наступний крок після створення блок-діаграми і формування лицьовій панелі ВП - створення іконки ВП і настройка 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 налагодженням блок-діаграм.

Будь VI може бути використаний як підпрограма при створенні в подальшому інших віртуальних інструментів. Для об'єднання декількох функціональних блоків розроблюваної блок-діаграми в підпрограму досить виділити їх мишкою на діаграмі, утримуючи клавішу Shift, і потім вибрати у верхньому меню пункт Edit>CreateSubVI. При цьому вони об'єднуються в нову підпрограму з новим значком (іконкою) на функціональній панелі. Подвійний клік на даному значку дозволить викликати створену підпрограму, налаштувати її належним чином і зберегти з заданим ім'ям.

У подальшому даний модуль може бути багато разів використаний у різних VI.

Кожен віртуальний прилад в правому верхньому куті лицьової панелі і у вікні блок-

діаграми відображає іконку . Іконка - графічне представлення приладу. Вона може містити текст, малюнок або і те й інше одночасно. Якщо ВП використовується як підпрограма, то іконка ідентифікує його на блок-діаграмі іншого ВП.

Для редагування іконки створюваних ВП і підпрограм достатньо натиснути правою кнопкою миші на піктограмі ВП в правому верхньому куті і вибрати пункт EditIcon ... (рис.4.1).

За допомогою найпростіших функцій графічного редактора можна створити власний варіант іконки. Налаштування входів/виходів (терміналів) підпрограм здійснюється наступним чином.

Необхідно натиснути правою кнопкою миші на піктограмі vi в правому верхньому куті і вибрати пункт ShowConnector ... (рис. 4.1). При цьому піктограма розділиться на кілька прямокутників, загальний набір і вид яких можна редагувати за допомогою спливаючого меню піктограми (додати / 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 клацнути на потрібному терміналі, а потім - на тому елементі управління або відображення на лицьовій панелі, якій він буде відповідати. При цьому термінал забарвиться в колір, відповідний типу даних зазначеного елемента керування або відображення. В результаті всі термінали будуть пов'язані з певними вхідними або вихідними даними.

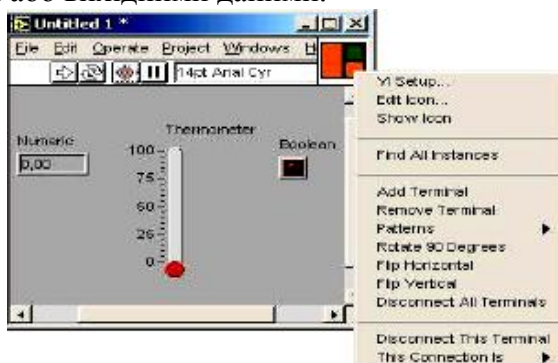


Рис. 4.1

Після створення ВП, оформлення його ікони і налаштування сполучної панелі 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.

Вправа 4.1.

ВП Перетворення °C в °F

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

Лицьова панель

1. Відкрийте файл з раніше створеним ВП Перетворення С в F (початок).vi.

Іконка і сполучна панель

2. Клацніть правою кнопкою миші по іконці ВП і в контекстному меню виберіть пункт EditIcon

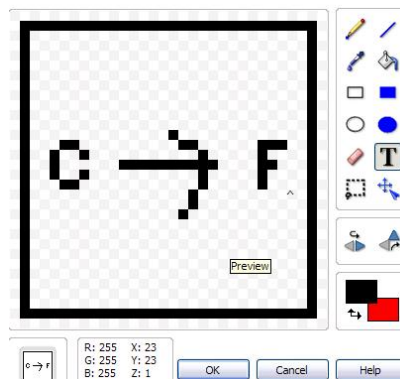
(Редагування іконки). З'явиться діалогове вікно редактора іконки IconEditor.

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 кордоном обраного кольору.

6. Створіть наступну іконку:



а) введіть текст інструментом введення тексту;

б) надрукуйте «С» 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 та утримувати клавішу <Shift>.

д) для пересування тексту і стрілки по полю редагування іконки використовуйте інструмент вибір і стрілки на клавіатурі;

е) у розділі Copyfrom (Копіювати з) виберіть B & W (чорно-білу) іконку і 256 Colors (256-кольоровий режим) для створення чорно-білої іконки, яку LabVIEW використовує в разі відсутності кольорового принтера;

ж) в розділі Copyfrom (Копіювати з) виберіть 16 Colors і 256 Colors;

з) після завершення редагування іконки натисніть кнопку ОК і закрийте IconEditor. Нова іконка з'явиться у правому верхньому куті обох панелей.

7. Перейдіть на лицьову панель, клацніть правою кнопкою миші на іконку, виберіть пункт ShowConnector (Показати поля введення / виведення даних) з контекстного меню. Кількість відображуваних LabVIEW полів вводу / виводу даних відповідає кількості елементів на лицьовій панелі. Наприклад, лицьова панель цього ВП має два елементи Град С і Град F і LabVIEW виводить в сполучної панелі два поля.

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

а) у пункті головного меню Help (Допомога) виберіть ShowContextHelp (показати контекстну довідку) і виведіть на екран вікно ContextHelp (контекстної довідки) для перегляду з'єднань;

б) клацніть лівою кнопкою мишки на лівому полі сполучної панелі. Інструмент управління автоматично зміниться на інструмент з'єднання, а вбрання поле забарвиться в чорний колір;

в) клацніть лівою кнопкою миші по елементу Град С. Ліве поле стане оранжевим і виділиться маркером;

г) клацніть курсором по вільному простору. Маркер зникне, і поле забарвиться в колір даних типу відповідного елемента 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 вводу / виводу даних. Вікно ContextHelp (контекстної довідки) покаже, що обидва поля відповідають типу даних подвійної точності з плаваючою комою.

9. Виберіть пункт головного меню File »Save. Збережіть ВП під ім'ям Перетворення С в F.vi.

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

Індивідуальне завдання:

Використовуючи попереднє завдання, створити 4 ПВП для перетворення різних значень даних, об'єднати їх в один ВП.

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?

Лабораторна робота №5

Управління виконанням програм за допомогою структур.

Мета: познайомитися з чотирма основними структурами LabVIEW: циклами по умові і фіксованим числом ітерацій, структурами варіанту і послідовності

Два типи структур циклів

Якщо ви вже програмували на якомусь мові, то вам, можливо, доводилося використовувати повторення виконання частини коду. LabVIEW пропонує дві структури циклів для полегшення реалізації цього процесу: цикл з фіксованим числом ітерацій (ForLoop) і цикл по умові (WhileLoop) для управління операціями, які повторюються, у віртуальному приладі. Цикл з фіксованим числом ітерацій виконується певну кількість разів, а цикл за умовою виконується до тих пір, поки певну умову більше не буде істинним. Обидва циклу знаходяться в підпалітрі **Структури** (Structures) палітри Функції.

Цикл з фіксованим числом ітерацій

Цикл з фіксованим числом ітерацій (ForLoop) виконує код всередині його границь (піддіаграму) деяке число ітерацій (count). Це число дорівнює величині, введений в термінал числа ітерацій (countterminal). Число відліків ви встановлюєте, подаючи певне значення ззовні циклу на термінал числа ітерацій. Якщо ви підключите до цього терміналу значення 0, цикл не буде виконуватись.

Термінал лічильника ітерацій (iterationterminal) містить поточне число завершених ітерацій циклу; 0 - під час першої ітерації, 1 - під час другої і т.д. до N-1, де N-кількість виконань циклу, яке ви задали.

Цикл з фіксованим числом ітерацій еквівалентний наступному псевдокоду:

For i = 0 to N-1

Executesubdiagram

Цикл за умовою

Цикл за умовою (WhileLoop) виконує код всередині його границь до тих пір, поки логічне значення (Booleanvalue), підключене до терміналу умови виходу з циклу не перейде в стан БРЕХНЯ (False). LabVIEW перевіряє термінал умови виходу після закінчення кожної ітерації. Якщо значення відповідає ІСТИНА (True), то виконується наступна ітерація. За замовчуванням термінал умови виходу перебуває в стані БРЕХНЯ. Якщо ви залишите його непідключеним, цикл виконуватися не буде (хоча в попередніх версіях LabVIEW цикл виконувався б тільки один раз).

Термінал лічильника ітерацій (iterationterminal) циклу за умовою поводитьсь точно так само, як і у випадку з циклом з фіксованим числом ітерацій.

Цикл за умовою еквівалентний наступному псевдокоду:

Do

Executesubdiagram

Whileconditionis TRUE

Ви можете змінити стан, який перевіряє термінал умови виходу з циклу. Якщо раніше цикл виконувався, поки на вхід терміналу надходило значення ІСТИНА (whiletrue), тепер цикл зупиниться, якщо на вхід надійде значення ІСТИНА (unlessit'strue). Щоб цього домогтися, клацніть правою кнопкою миші по терміналу умови і виберіть опцію Зупинити, якщо Істина (StopifTrue). Цикл за умовою буде виглядати, як показано на рис. 5.1.

Тут цикл еквівалентний наступному псевдокоду:

Do
Executesubdiagram
Whileconditionis TRUE

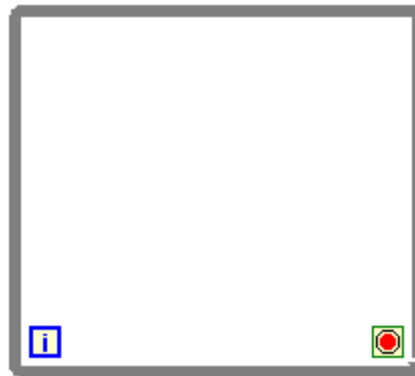


Рис. 5.1.

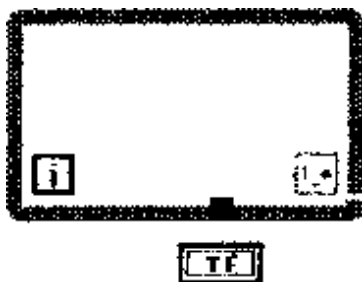
Термінали всередині циклів

Дані надходять в цикл і виходять з нього через маленьке вікно на кордоні циклу, зване точкою входу/виходу в структуру (tunnel). Оскільки Lab VIEW працює у відповідності з принципом потоку даних, то перш ніж цикл почне виконуватися, вхідні точки повинні передати в нього свої дані. Вихідні точки циклу виводять дані лише після завершення всіх ітерацій.

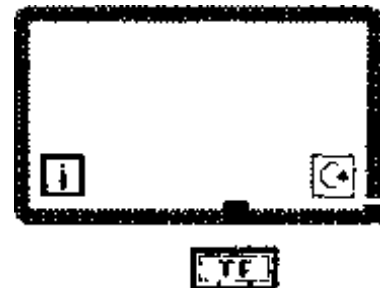
Також відповідно до принципу потоку даних, для того щоб термінал оновлював свої значення на кожній ітерації циклу, ви повинні помістити його всередину циклу. Наприклад, цикл за умовою, показаний на рис. 5.2а, перевіряє стан логічного елемента управління на кожній ітерації. Якщо з елемента зчитується значення БРЕХНЯ, то цикл завершує роботу.

Якщо ви помістите термінал логічного елемента управління за межами циклу за умовою, як це показано на рис. 5.2 б, то ви створите цикл або з нескінченним числом ітерацій, або з одноразовим виконанням - залежно від початкового значення логічного елемента. Відповідно до принципу потоку даних LabVIEW зчитує логічне значення перш, ніж воно надходить в цикл, а не в процесі виконання циклу або після його завершення.

Рис. 5.2а)



б)



Саме так числовий індикатор (DigitalIndicator) в циклі на рис. 5.3 буде оновлюватись під час кожної ітерації. Це позначення на рис. 5.4 оновиться тільки один раз після завершення циклу. Він міститиме випадкове число, що з'явилося при останній ітерації.

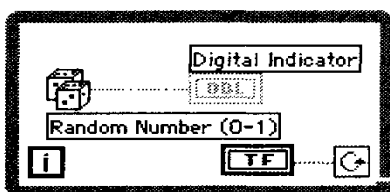


Рис. 5.3

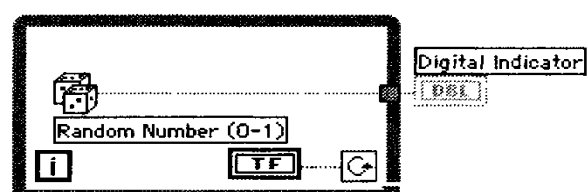


Рис. 5.4

Якщо ви хочете видалити цикл, не видаляючи його вмісту, клацніть правою кнопкою миші по його кордоні і виберіть опцію **Видалити цикл** за умовою (RemoveWhileLoop) або Видалити цикл з фіксованим числом ітерацій (RemoveForLoop) відповідно. Якщо ви просто виділите цикл інструментом переміщення і видаліть його, то всі об'єкти всередині будуть також видалені.

Ви можете створити масиви даних, використовуючи цикл, і зберегти їх на його кордонах з допомогою функції LabVIEW - автоматична індексація (auto-indexing).

Пам'ятайте, що під час першої ітерації обох типів циклів значення лічильника числа ітерацій дорівнює нулю. Якщо ви хочете показати, скільки повторень циклу залишилось, додайте одиницю до значення лічильника.

Вправа 5.1: рахунок за допомогою циклів

У цій вправі ви створите цикл з фіксованим числом ітерацій, який відобразить значення лічильника на розгортці осцилограми лицьовій панелі. Виберіть число ітерацій (опція NumberofIterations), і цикл почне відлік від нуля до цього числа мінус 1 (оскільки відлік починається з нуля). Потім створіть цикл за умовою, що буде рахувати до тих пір, поки ви не зупините його перемикачем логічних значень (Booleanswitch). Ви також побачите ефект, який виникає при приміщенні елементів управління і відображення за межами циклу за умовою.

1. Створіть нову лицьову панель.

2. Побудуйте лицьову панель і блок-діаграму, показану на рис. 5.5 і 5.6. Цикл з фіксованим числом ітерацій розташовується в підпалітрі Структури палітри Функції. Команда Розмістити ліворуч і праворуч (TileLeftandRight) з меню Вікно – це дозволить вам побачити одночасно лицьову і діаграмну панелі.

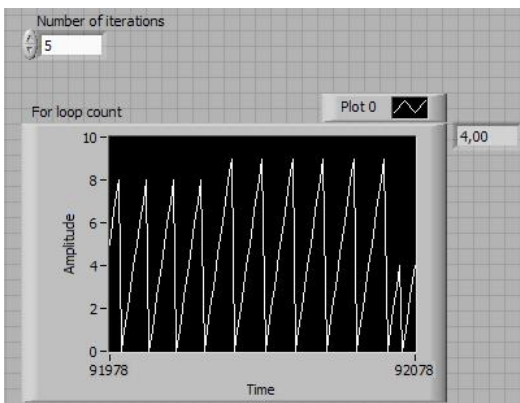


Рис. 5.5

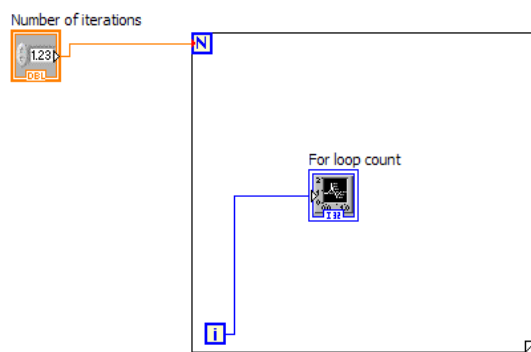


Рис. 5.6

Помістіть розгортку осцилограми (WaveformChart) з палітри Controls на лицьову панель ВП. Назвіть її Лічильник числа ітерацій. Використовуйте цифровий елемент управління з підпалітри Числові для управління значенням Число ітерацій.

3. Клацніть правою кнопкою миші по розгортці осцилограми і виберіть опцію **Автомасштабування** по осі Y (AutoScale Y) в меню Шкала Y (YScale), щоб графік автоматично підстроював масштаб для відображення зростаючого значення лічильника циклу. Знову клацніть правою кнопкою миші по розгортці осцилограми і виберіть Видимі елементи => Цифровий дисплей (VisibleItems => DigitalDisplay). Тепер введіть значення в елемент управління Число ітерацій і запустіть ВП. Зверніть увагу, що цифровий елемент відображення вважає від 0 до NI (де N - введене вами число). Кожен раз, коли цикл виконується, на графіку по осі Y відображається значення лічильника циклу щодо часу по осі X. В цьому випадку кожен відлік часу ініціює виконання однієї ітерації циклу.

4. Зверніть увагу на наявність маленької сірої точки в місці з'єднання терміналу числа ітерацій і провідника числа ітерацій. Вона називається точкою приведення типів, і ми

поговоримо про неї пізніше. Клацніть правою кнопкою миші по елементу управління числа ітерацій і виберіть опцію 132 Long в меню Подання (Representation) для її усунення.

5. При бажанні ви можете зберегти віртуальний прилад, але ми не будемо більше ним користуватися.

Вправа 5.2

1. Відкрийте нове вікно і створіть цикл за умовою.
2. Побудуйте ВП, зображений на рис. 5.8.

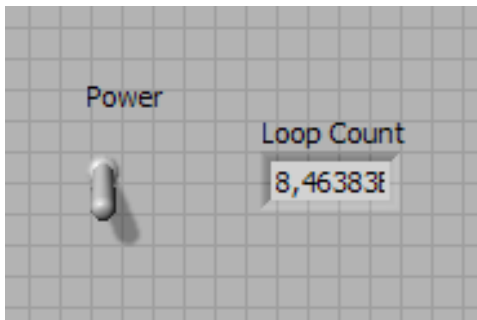


Рис. 5.7

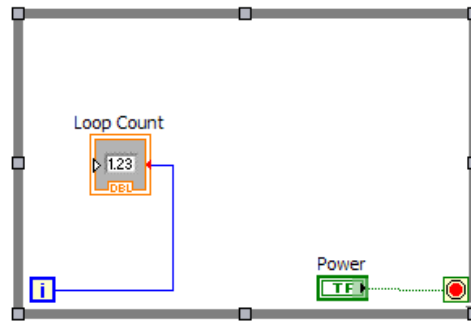


Рис.5.8

3. Переведіть перемикач в положення ІСТИНА, клацнувши по ньому інструментом управління («палець»), і запустіть ВП. Для того щоб зупинити виконання, переведіть перемикач в положення БРЕХНЯ. Значення елемента відображення Лічильник циклу буде оновлюватися під час кожної ітерації.

4. Коли перемикач знаходиться в положенні БРЕХНЯ, знову запустіть ВП. Відзначте, що цикл за умовою виконується тільки один раз. Пам'ятайте, що цикл перевіряє стан терміналу умови виходу в кінці повторення.

5. Тепер перейдіть до панелі діаграм і перемістіть елемент відображення лічильника числа ітерацій за межі циклу, як показано на рис. 5.9. При цьому точка виходу створюється автоматично під час виведення провідника із циклу.

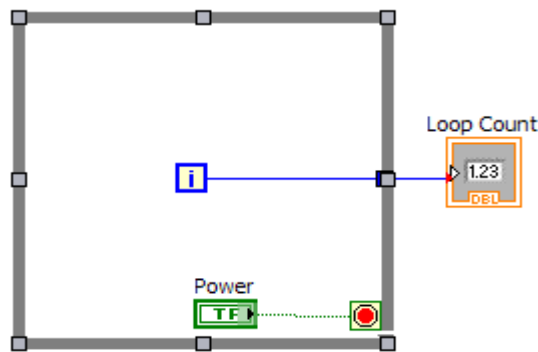


Рис. 5.9

6. Переконайтеся, що перемикач знаходиться в положенні ІСТИНА, і запустіть ВП. Зверніть увагу, що значення елемента відображення оновлюється лише після того, як цикл закінчився. Він містить кінцеве значення лічильника ітерацій, яке з'являється після завершення циклу.

Тепер, щоб показати, чого робити не слід, виведіть логічний перемикач із циклу (але залиште підключеним). Переконайтеся, що перемикач знаходиться в положенні ІСТИНА, і запустіть віртуальний прилад, потім зупиніть його виконання. Він не зупиняється? Як тільки LabVIEW входить в цикл, він не буде перевіряти значення елементів управління за межами циклу (точно так само, як він не оновлював елемент відображення лічильника ітерацій, поки цикл не завершився). Натисніть кнопку Перервати (Abort) на

інструментальній панелі для зупинення виконання. Якщо перемикач при запуску циклу знаходився в положенні БРЕХНЯ, то цикл виконається тільки один раз. Закрийте ВП без збереження змін.

Точка приведення типів

Пам'ятайте маленьку сіру точку в місці з'єднання терміналу числа ітерацій і провідника Число ітерацій в останній вправі? Це точка приведення типів (coercion dot). Вона названа так тому, що LabVIEW призводить (перетворює) одне числове подання до іншого. Якщо з'єднати два термінали різних числових уявлень, то LabVIEW перетворює одне з них в інше, яке має інший термінал. У попередній вправі термінал числа ітерацій мав уявлення 32-бітових цілих чисел, тоді як елемент управління числом ітерацій являє собою (за замовчуванням) число подвоєною точності з плаваючою комою. У цьому випадку LabVIEW перетворює число подвоєною точності з плаваючою комою в довге ціле число. Виконуючи цю операцію, LabVIEW створює в пам'яті нову копію числа вже у відповідному поданні. Така копія займає певний обсяг пам'яті. Незважаючи на те що додаткові витрати пам'яті для скалярних чисел нехтує малі, цей обсяг швидко збільшується при використанні масивів даних. Спробуйте звести до мінімуму появу точки приведення типів у великих масивах даних шляхом зміни представлення ваших елементів управління і відображення, щоб воно точно збігалося з поданням проходять по ним даних.

Коли ВП перетворює числа з плаваючою точкою в цілі, то він округлює їх до найближчого цілого числа. Число з десятковим значенням «.5» округлюється до найближчого парного цілого числа.

Найпростіше створити елемент управління для терміналу числа ітерацій з правильним типом даних і поданням, викликавши контекстне меню терміналу і вибравши опцію Створити константу (CreateConstant) для створення постійного числа на блок діаграмі або Створити елемент управління (CreateControl) для створення елемента управління на лицьовій панелі.

Структури варіанту

Структура варіанта (CaseStructure) є методом виконання тексту, що містить умову, тобто аналогом оператора імплікації (if-then-else). Ви можете знайти цю структуру в підпалітрі Структури палітри Функції. Структура варіанту, показана на рис. 5.10, має дві або більше піддіаграми або варіанта. Лише одна з них виконується в залежності від логічного, числового або строкового значення, яке ви подаєте на термінал селектора структури варіанту.

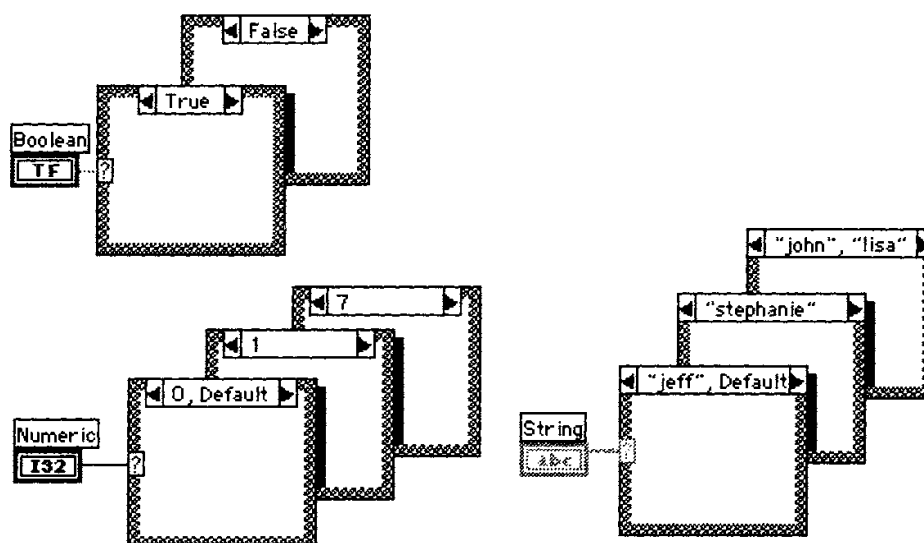


Рис. 5.10

Якщо до терміналу селектора структури варіанту підключено логічне значення, то структура матиме два варіанти: БРЕХНЯ і ІСТИНА.

Якщо до терміналу селектора підключені числові або строкові дані, то структура може мати майже необмежену кількість варіантів, починаючи з нульового. Спочатку існує лише два варіанти, але ви легко збільшите їх кількість. Допустимо використовувати кілька значень для одного варіанта, відокремлюючи їх комами, як показано на рис. 5.10. Крім того, ви завжди можете вибрати варіант За замовчуванням (Default), який буде виконуватися, якщо величина, що подається на термінал селектора структури, не відповідає ніякому іншому варіанту. Це досить зручно в тому випадку, якщо ви не хочете думати про який-небудь можливий варіант, але хочете використовувати загальний (відповідний для будь-яких випадків) варіант.

При розміщенні структури варіанту на лицьовій панелі в перший раз вона буде представлена в логічній формі. Для того щоб використовувати в структурі числові значення, необхідно подати числовий тип даних на термінал селектора.

Структури варіанти можуть мати численні піддіаграми, але одноразово ви зможете побачити тільки одну, як у колоді карт (зовсім не те, що було показано на рис. 5.10, де ми схитрували і показали кілька малюнків). Клацнувши мишею по лівій чи правій стрілкою селектора у верхній частині структури, ви побачите відповідно попередні чи наступні піддіаграми. Ви також можете зробити клацання мишею по селектору у верхній частині структури, щоб викликати меню, що випадає, що показує всі варіанти, а потім вибрати потрібний. Іншим способом перемикання варіантів є клацання правою кнопкою миші по межі структури і вибір опції Показати варіант (ShowCase).

Якщо ви подасте на термінал селектора число з плаваючою точкою, LabVIEW округлить це число до найближчого цілого. LabVIEW примусово призводить негативні числа до нуля і зменшує будь-яке значення, яке перевищує найбільший номер варіанта, для прирівнювання його до найбільшого номеру.

Ви можете помістити термінал селектора структури варіанту в будь-якому місці вздовж лівої межі. Цей термінал завжди повинен бути підключеним. Коли ви зробите це, селектор автоматично привласнить собі тип підводяться даних. Якщо ви зміните тип даних, що подаються на термінал селектора, з числових на логічні, то варіанти 0 і 1 зміняться на БРЕХНЯ і ІСТИНА. Якщо ж є інші варіанти (від 2 до n), то LabVIEW не скидає їх - раптом зміна в типі даних було випадковим. Проте ви повинні видалити зайві варіанти, перш ніж структура почне виконуватися.

Для строкових типів даних, що подаються на термінал селектора, потрібно точно визначити величини, що досягається приміщенням строкових даних в лапки. Єдиним винятком є слово Default, яке в лапки ніколи не укладається.

Підключення терміналів вводу / виводу

Дані в усіх вхідних терміналах (точках введення і терміналі селектора) структури варіанту доступні для всіх варіантів. При роботі з варіантами не обов'язково використовувати вхідні дані або виводити дані зі структури, але якщо в одному варіанті дані виводяться, то всі варіанти повинні видавати дані. При виведенні даних назовні з одного варіанта структури в усіх варіантах з'явиться не зафарбована вихідна точка в тому ж самому місці. Стрілка запуску ВП буде «зламаною» до тих пір, поки ви не подасте в цю точку дані з кожного варіанту. Тоді точка забарвиться в чорний колір (а потім в колір переданих даних), а стрілка запуску буде в робочому стані (за умови, що ви не зробили інших помилок). Переконайтеся, що ви підключили провідник безпосередньо до існуючої вихідній точці, інакше ви можете випадково створити велику їх кількість.

Чому треба підключати вихідні дані структури для кожного варіанту? Тому, що структура варіанти повинна передавати значення в наступний вузол даних незалежно від того, який варіант виконується. LabVIEW змушує вас вибирати потрібне значення, а не робить це сам, що корисно з точки зору отримання гарних навичок програмування.

Вправа 5.3: витяг квадратного кореня

Ця вправа дозволить вам придбати деякий навик роботи зі структурами варіанту і діалоговими вікнами. Створіть ВП, який обчислює квадратний корінь з позитивного вхідного числа. Якщо вхідна число є негативним, то ВП викличе діалогове вікно і поверне помилку.

1. Відкрийте нову панель.
2. Створіть лицьову панель, зображену на рис. 5.11.

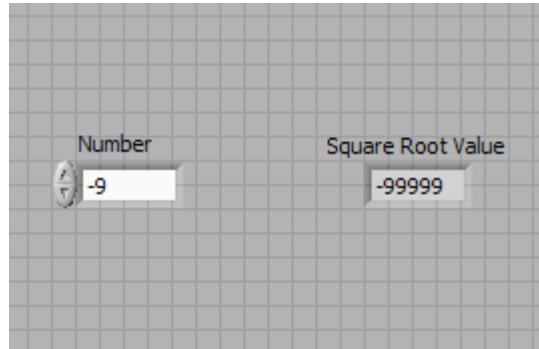


Рис. 5.11

За допомогою числового елемента управління буде вводиться число. Індикатор Величина квадратного кореня покаже квадратний корінь з числа.

3. Відкрийте вікно блок-діаграми. Створіть код, зображений на рис. 5.12 і 5.13.

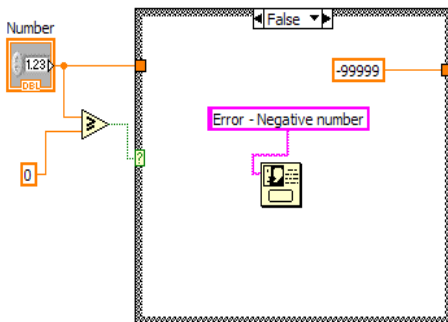


Рис. 5.12 Варіант БРЕХНЯ

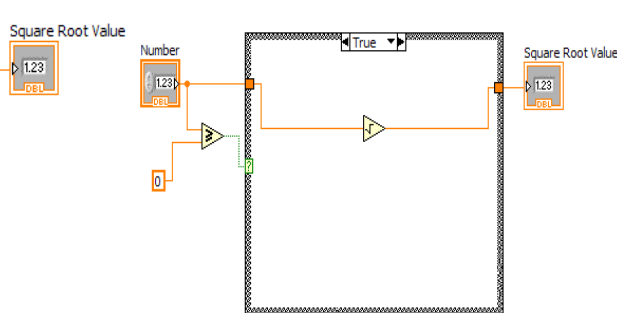


Рис. 5.13 Варіант ІСТИНА

4. Помістіть структуру варіанта (підпалітра Структури) у вікно блок-діаграми. Як і у випадку з циклом з фіксованим числом ітерацій і циклом за умовою, клацніть курсором обраної структури і, переміщаючи його, визначте потрібні кордону.

Пам'ятайте, що варіанти відображаються тільки по одному. Для зміни варіантів клацніть кнопкою миші по стрілках селектора структури варіанту. Зверніть увагу, що рис. 5.12 і 5.13 показують обидва варіанти структури. Таким чином, ви знаєте, що створювати. Не створюйте дві різні структури в цій вправі!

5. Виберіть інші об'єкти діаграми і з'єднайте їх так, як показано на рис. 5.12 і 5.13. Використовуйте вікно контекстної допомоги для відображення введів і висновків різних функцій.

У даній вправі ВП виконуватиме варіант ІСТИНА або варіант БРЕХНЯ структури варіанту. Якщо вводиться число більше або дорівнює нулю, то ВП виконуватиме варіант ІСТИНА, який обчислює квадратний корінь від цього числа. Якщо число менше нуля, то виконання варіанту БРЕХНЯ призводить до вихідного числа -99999.0 і відкриття діалогового вікна з повідомленням «Помилка ... Негативне число».

Пам'ятайте про те, що ви обов'язково повинні визначити значення, що подається у вихідну точку структури, для обох варіантів. Тому ми і потурбувалися подати число -99999.0

у разі помилки. Якщо ви створите вихідну точку для одного варіанта, ця ж точка виникне і для інших варіантів. Чи не поєднана ні з чим вихідна точка виглядає як порожній квадратик. Подбайте з'єднати з яким-небудь джерелом даних все Несполучені точки. Причому починайте з'єднання завжди з точки, інакше ви випадково створите нову вихідну точку.

Поверніться до лицьовій панелі і запустіть ВП. Спробуйте ввести одне число більше нуля, а інше - менше нуля.

Збережіть і закрийте ВП.

Логіка віртуального приладу, що обчислює квадратний корінь, така:

If (Число >= 0) then

Величина квадратного кореня = SQRT (Число) Else

Величина квадратного кореня = -99999.0

DisplayMessage «Помилка ... Негативне число» EndIf

Функція вибору

У простих випадках застосування логіки **if-then-else** іноді більш зручно користуватися функцією LabVIEW Вибір (Select), яка працює так само, як і структура варіанти.

Функція Вибір, що знаходиться в підпалітрі Порівняння палітри Функції, повертає значення t, якщо вхідне значення ІСТИНА, і значення f, якщо на вхід подається БРЕХНЯ. За допомогою цієї функції можна зробити те ж саме, що й у випадку структури варіанту в останній вправі - за винятком виклику діалогового вікна.

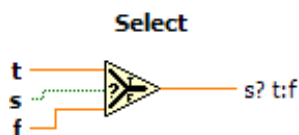


Рис. 5.14

Вправа 5.4: витяг квадратного кореня за допомогою функції Вибір.

Створіть ВП, аналогічне попередньому, використовуючи функцію Select.

Структура послідовності

Визначення порядку виконання програми шляхом організації її елементів в певну послідовність називається управлінням потоком даних. У звичайних мовах програмування, таких як Basic або C, завжди присутня управління потоком, оскільки оператори виконуються в тому порядку, в якому вони написані в програмі. Для здійснення управління потоком при обробці даних в LabVIEW використовується структура послідовності (SequenceStructure). Структура послідовності виконує кадр 0, за яким слідує кадр 1, потім кадр 2 і т.д., поки не виконається останній кадр. І лише після цього дані залишають структуру.

Структура послідовності, показана на рис. 5.15, вельми схожа на кадр кіноплівки. Її можна знайти в підпалітрі Структури палітри Функції. Так само як і в структурі варіанту, одноразово тут відображається тільки один кадр. Щоб побачити інші кадри, потрібно натискати на стрілки селектора у верхній частині структури. Також можна клацнути кнопкою миші по селектору, щоб подивитися існуючі кадри, а потім вибрати потрібний, або натиснути правою кнопкою миші по межі структури і вибрати опцію Показати кадр (ShowFrame). Коли ви в перший раз ставите структуру послідовності на блок-діаграму, вона буде мати тільки один кадр; отже, немає ні стрілок, ні номерів у верхній частині структури, щоб показати, який кадр слід. Створіть нові кадри, клацнувши правою кнопкою миші по межі структури і вибравши опцію Створити кадр після (AddFrameПослідовності After) або Створити кадр перед (AddFrameBefore) поточним кадром.

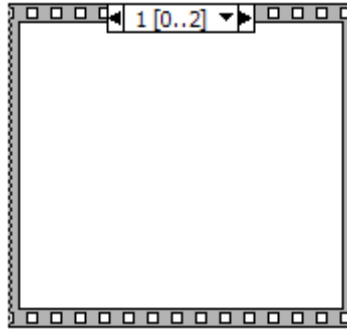


Рис. 5.15

Структуру послідовності використовують для керування порядком виконання вузлів даних, які не залежать одне від одного. В рамках кожного кадру, як і в решті частини блок-діаграми, залежність даних визначає порядок виконання вузлів даних.

Термінали вхідних і вихідних даних цієї структури можуть мати тільки одне джерело даних - на відміну від структури варіанту, вихідні термінали якого повинні мати окреме джерело даних для кожного варіанту. Вихідних дані можуть бути отримані з будь-якого кадру, однак дані виходять із структури тільки тоді, коли вона повністю завершує своє виконання, а не кожен її кадр. Дані вхідних терміналів доступні для всіх кадрів.

Термінали локальної змінної

Щоб передати дані з одного кадру в будь-який наступний, ви повинні скористатися так званим **терміналом локальної змінної** (sequenceLocal). Для створення терміналу локальної змінної виберіть опцію Створити локальну змінну (AddSequenceLocal) з контекстного меню кордону структури. Ця опція буде недоступна, якщо ви клацнете правою кнопкою миші досить близько від іншого терміналу локальної змінної або над вікном дисплея піддіаграми. Ви можете перемістити термінал локальної змінної в будь-який вільний місце на кордоні структури. Використовуйте команду Видалити (Remove) з контекстного меню терміналу локальної змінної для видалення терміналу або виділіть, а потім видаліть його.

Коли термінал локальної змінної вперше з'являється на блок-діаграмі, він являє собою маленький жовтий прямокутник. На рис. 5.16 - 5.17 показано термінал локальної змінної в різних формах. Коли ви підключаєте джерело даних до цього терміналу, в ньому з'являється стрілка, спрямована назовні, що говорить про те, що цей термінал містить джерело даних. Термінали в наступних кадрах містять стрілку, спрямовану всередину, - значить, цей термінал є джерелом даних для цих кадрів. У кадрах, що передують кадру джерела даних, ви не зможете використовувати термінал локальної змінної (хоча б тому, що на нього ще не було подано якоесь значення), і він з'являється у вигляді заштрихованого прямокутника.

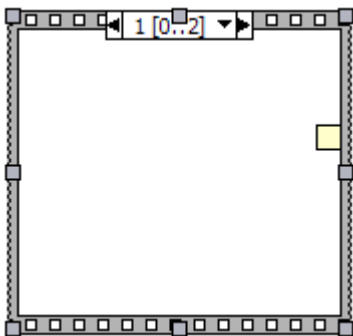


Рис. 5.16 Локальну змінну не можна використовувати приймач даних

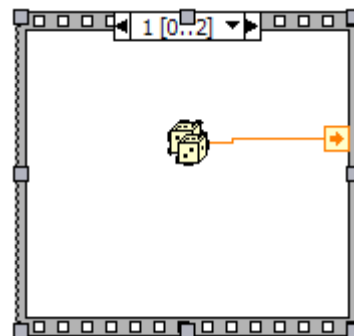


Рис. 5.17 Термінал локальної змінної

Регулювання та хронометраж часу виконання ВП

Іноді корисно відстежувати час роботи вашого ВП і керувати ним. З цими завданнями чудово справляються функції Затримка (мс), Лічильник часу (мс) і Затримка до наступного кратного інтервалу мс, розташовані в підпалітрі Час і діалоги палітри Функції.

Функція Затримка (мс) - Wait (ms) - змушує ВП чекати певне число мілісекунд, перш ніж продовжити виконання (рис. 5.18).

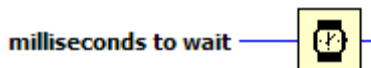


Рис. 5.18. Функція Затримка (мс)

Функція Затримка до наступного кратного інтервалу мс (WaitUntilNextmsMultiple) змушує LabVIEW очікувати, поки показання внутрішніх годин не зрівняються або не перевищать кратного кількості мілісекунд, поданих на вхід функції, перш ніж відновити виконання ВП (рис. 5.19).

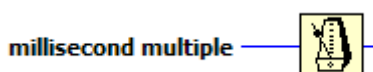


Рис. 5.19. Функція Затримка до наступного кратного інтервалу (мс)

Ця функція змушує цикли виконуватися через певні інтервали часу і дозволяє синхронізувати роботу. Дві названі функції схожі, але не ідентичні. Наприклад, затримка виконання за допомогою функції Затримка до наступного кратного інтервалу мс, можливо, буде менше заданого числа мілісекунд при першій ітерації, оскільки залежить від значення годин під час Упорядкування (тобто від того, скільки часу буде потрібно для переходу до наступної ітерації і відновлення роботи ВП). Крім того, якщо цикл все ще виконується, а годинник вже пройшли кратний мілісекунди інтервал, то ВП буде чекати до тих пір, поки годинник досягнутий наступного кратного мілісекундного інтервалу. Таким чином, ВП може вийти з синхронізації і сповільнитися. Переконайтеся, що ви все врахували при використанні цих функцій.



Рис. 5.20. Функція Лічильник часу (мс)

Функція Лічильник часу (мс) - TickCount (ms) - повертає значення внутрішнього годинника операційної системи в мілісекундах (рис. 5.20). Вона в основному використовується для підрахунку минулого часу. Майте на увазі, що внутрішній годинник не завжди мають велику роздільну здатність: один відлік годин може становити до 55 мс в Windows 95/98, 10 мс в Windows 2000 / NT, 17 мс в Linux, Solaris і MacOS.

Вправа 5.5: числа-збіги

Тепер у вас є можливість попрацювати зі структурою послідовності і однією з тимчасових функцій. Створіть ВП, що обчислює час, займане процесом збігу заданого числа з числом, згенерованим лічильником випадкових чисел. Запам'ятайте цей алгоритм - можливо, у вас виникне необхідність розрахувати час виконання ВП LabVIEW.

1. Відкрийте нову лицюву панель.
2. Створіть лицюву панель, як показано на рис. 5.21.

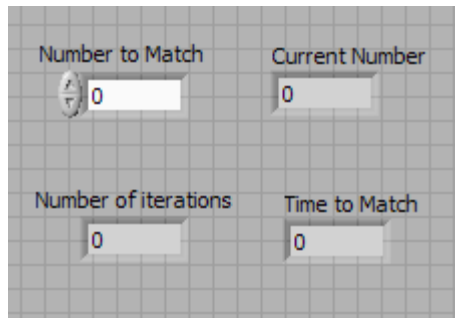


Рис. 5.21

3. Зменшіть точність елементів управління і відображення Число для збіги, Поточне число і Число ітерацій до нуля, вибравши опцію Формат і точність (Format&Precision) в їх контекстних меню. Введіть 0 для опції Розряди точності (DigitsofPrecision), так щоб праворуч від десяткової коми не з'явилося жодної цифри.

4. Відкрийте вікно діаграми і побудуйте діаграму, як показано на рис. 5.22- 5.23

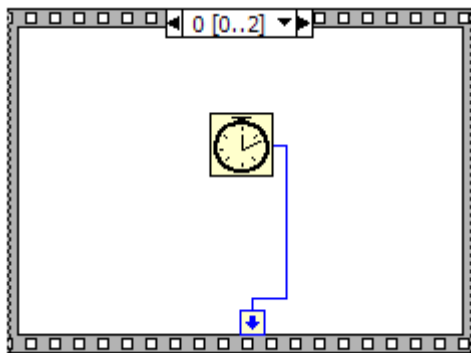


Рис. 5.22

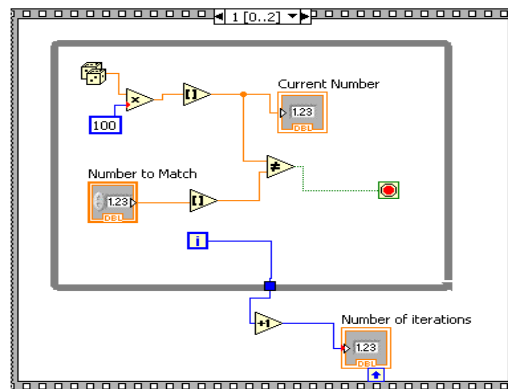


Рис. 5.23

5. Додайте структуру послідовності у вікні діаграми. Це робиться таким же чином, як і для циклу з фіксованим числом ітерацій і циклу за умовою - клацніть курсором у формі структури і задайте бажані кордону. Вам потрібно створити три окремих кадру структури послідовності. Для створення нового кадру клацніть правою кнопкою миші по межі кадру і виберіть опцію Створити кадр після контекстного меню.

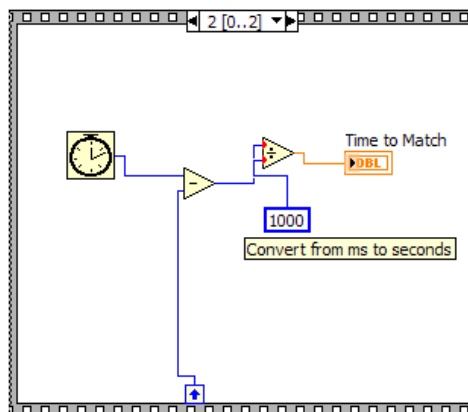


Рис. 5.24

6. Створіть термінал локальної змінної, клацнувши правою кнопкою миші по нижній межі нульового кадру і вибравши пункт Створити локальну змінну в контекстному меню.

Термінал локальної змінної відобразиться у вигляді порожнього квадрата. Стрілка всередині квадрата з'явиться автоматично, коли ви підключіться до локальної змінної.

7. Добудуйте діаграму. У процесі підключення використовуйте вікно контекстної допомоги для відображення введів і висновків функцій.

При виконанні нульового кадру функція Лічильник часу (мс) повертає свідчення внутрішнього годинника в мілісекундах. Це значення подається на термінал локальної змінної, тому воно буде доступним в наступних кадрах. У кадрі 1 ВП виконує цикл за умовою доти, поки введене число не співпаде з числом, повернутим функцією Випадкове число (0-1). У кадрі 2 функція Лічильник часу (мс) повертає новий відлік часу в мілісекундах. Віртуальний прилад віднімає старий відлік часу (отриманий в нульовому кадрі і переданий через термінал локальної змінної) з нового для обчислення минулого часу, а потім ділить на 1000 для перекладу одиниць виміру з мілісекунд в секунди.

Існує більш ефективний спосіб виконання цієї вправи, заснований на використанні двохкадрової структури послідовності. Однак нам би хотілося, щоб ви користувалися терміналами локальної змінної.

8. Увімкніть підсвітку виконання, яка уповільнює роботу ВП, для того, щоб побачити поточний генерується число на лицьовій панелі.

9. Введіть число в елемент управління Число для збіги і запустіть ВП. Для прискорення виконання вимкніть підсвічування.

Вузол Формула

Тепер, коли ви познайомилися з основними структурами управління потоками даних LabVIEW, ми представимо структуру, яка не впливає на потік даних програми. Вузол Формула є вікном із змінними розмірами для введення алгебраїчних формул безпосередньо в блок-діаграму. Ця особливість особливо корисна, коли для обчислення необхідно використовувати довгу формулу. Наприклад, розглянемо просте рівняння $y = x^2 + x + 1$. Якщо ви напишете код з використанням арифметичних функцій LabVIEW на блок-діаграмі для обчислення навіть такої простої формули, він буде досить складним для розуміння в порівнянні з текстовими рівняннями.

Ви можете ввести той самий вираз у вузол Формула, як це показано на рис. 5.25

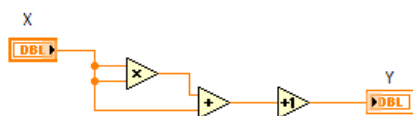


Рис. 5.25

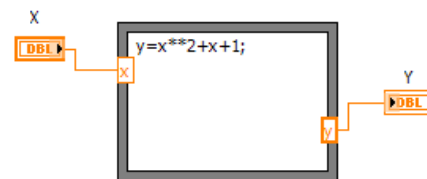


Рис. 5.26

За допомогою **вузла Формула** допустимо безпосередньо ввести формулу або формули у вікно замість створення складних підрозділів блок-діаграми. Вхідні і вихідні термінали вузла Формула можна створити, клацнувши правою кнопкою миші по межі вузла і вибравши опцію Додати введення (AddInput) або Додати висновок (AddOutput) у контекстному меню. Ім'я змінної чутливо до регістру букв. Кожен рядок у вузлі Формула повинна закінчуватися крапкою з комою (;).

Знайти функцію Вузол Формула можна в підпалітрі Структури палітри Функції.

Інформацію про операції та функціях, які можуть застосовуватися усередині вузла Формула, ви знайдете у вікні контекстної допомоги.

Наступний приклад показує вибір даних за умовою усередині вузла Формула. Подивіться на фрагмент коду, схожий на вправу 5.3, в якому обчислюється квадратний

корінь величини x , якщо x позитивне число. При цьому видається результат y . Якщо x негативне число, то код видає значення y , рівне -99 .

```
if (x >= 0) then
  y = sqrt(x) else
  y = -99 endif
```

Ви можете ввести фрагмент коду, використовуючи вузол Формула, як це показано на рис. 5.27.

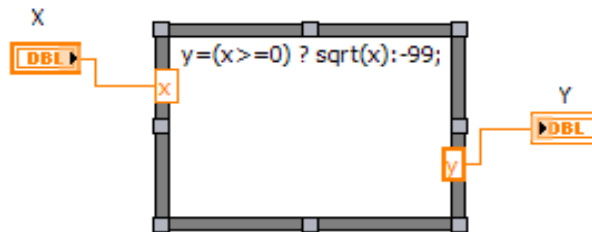


Рис. 5.27

Вправа 5.6: вправа з вузлом Формула

Ви можете створити ВП, який використовує вузол Формула для рішення рівняння $y = \sin(x)$ і видає результат у вигляді графіка.

1. Відкрийте нову лицьову панель. Виберіть Графік осцилограми (WaveformGraph) в підпалітрі Графіки палітри Елементи управління. Назвіть його Графік.
2. Побудуйте блок-діаграму, зображену на рис. 5.28.

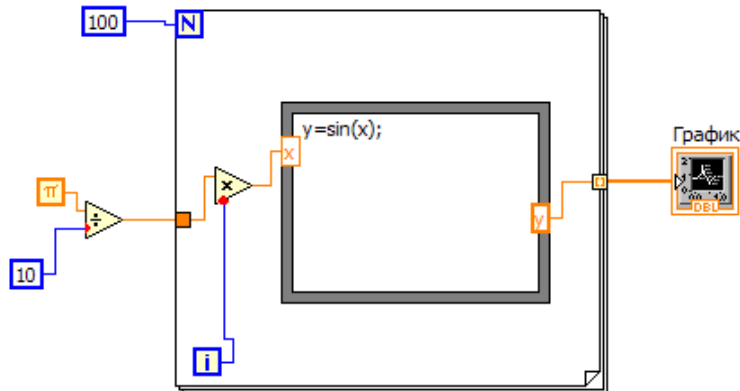


Рис. 5.28

За допомогою вузла Формула ви можете безпосередньо ввести математичні формули. Створіть термінал введення, клацнувши правою кнопкою миші по межі і вибравши функцію Додати введення в контекстному меню. Потім створіть термінал виведення, вибравши функцію Додати висновок в контекстному меню.

Коли термінали введення і виведення створені, дайте їм імена змінних, точно відповідні іменам змінних, які використовуються у формулі. Пам'ятайте, що імена чутливі до регістру букв, а кожен рядок закінчується крапкою з комою (;).

Постійна π розташована в палітрі Функції => Числові => Додаткові числові константи (AdditionalNumericConstants).

Під час кожної ітерації ВП примножує значення лічильника ітерацій на $\pi / 10$. Результат множення подається у вузол Формула, де від цього числа обчислюється синус. Потім ВП зберігає результат в масиві на кордоні циклу з фіксованим числом ітерацій. Після

того як цикл з фіксованим числом ітерацій закінчить виконання, віртуальний прилад відобразить дані масиву на графіку.

3. Поверніться до лицьовій панелі і запустіть ВП. Допустимо використовувати готову функцію Синус (Sine) в палітрі Функції => Числові => Тригонометричні (Trigonometric) для виконання тих же операцій, що і вузол Формула в цій вправі.

4. Збережіть ВП під ім'ям FormulaNodeExercise.vi. Закрийте ВП.

Логіка виконання ВП така:

```
for i = 0 to 99
x = i * (PI / 10)
y = sin (x)
array [i] = y next i
Graph (array)
```

Контрольні питання:

1. Опишіть принцип роботи з циклом For і циклом за умовою.
2. Що таке точка входу / виходу в структуру, для чого вона потрібна?
3. Що таке точка приведення типів?
4. Що таке структури варіанту?
5. Для чого можна використовувати лічильник часу?
6. Для чого потрібен вузол Формула?
7. Опишіть принцип роботи структури послідовності.

Варіанти індивідуального завдання:

1. Створіть ВП, що генерує випадкові числа в циклі While. Організуйте вихід з циклу після натискання кнопки на лицьовій панелі ВП
2. Порахуйте значення виразу $Y = X^2 + Z^3 - XZ + 10$ за допомогою блоку FormulaNode і ВП Formula Express, розташованого в палітрі Functions »Arith / Compare
3. Створіть ВП, що генерує 70 випадкових чисел в циклі FOR
4. Створіть ВП, який за допомогою FormulaNode вважає значення виразу $Y = x + \cos(x) - 10$ і якщо $Y > 0$, то $Z = \sqrt{Y}$
5. Використовуючи структуру Case, створіть ВП, який вважає різниця двох чисел і якщо отримане число ≥ 0 , то обчислюється значення кореня, інакше видається повідомлення про помилку
6. ВП здійснює почергове включення індикаторів на лицьовій панелі; проміжки між включеннями індикаторів 2, 3 і 7 с, відповідно. Використовуйте послідовність SequenceStructure і функцію TimeDelay, розташовану в палітрі Functions »TimeandDialog
7. Порахуйте значення виразу $Y = X^5 + \cos^2(Z) - XZ + 10$ за допомогою блоку FormulaNode і ВП Formula Express, розташованого в палітрі Functions »Arith / Compare (арміфметика / порівняння)
8. Використовуючи структуру Case, створіть ВП, який вважає значення виразу $y = ax + 14$, де $a = \text{const}$; і якщо $y \geq 0$, то обчислюється значення кореня, інакше видається повідомлення про помилку
9. Створіть ВП, який за допомогою FormulaNode вважає значення виразу $y = \sin(x)$, якщо y - позитивне число, то $z = y + A$, інакше $z = y - A$

Лабораторна робота №6

Складові дані LabView: масиви

Мета - вивчення двох більш складних складових типів даних - масивів.

Завдання:

- Вивчити вбудовані функції роботи з масивами
- Зрозуміти концепцію поліморфізму

Що таке масиви

До теперішнього часу ми мали справу лише зі скалярними числами (скалярна величина являє собою тип даних, які містять єдине значення), але зараз прийшов час поговорити про більш складні речі. Масив (array) LabVIEW являє собою набір елементів даних одного типу, так само як і в традиційних мовах програмування. Масив може мати одну або декілька розмірів, тобто бути одновимірним або багатовимірним, і включати до 231 елементів на одну розмірність (природно, в залежності від обсягу пам'яті). Елементом масиву може бути будь-який тип даних за винятком масиву, таблиці або графіка.

Доступ до елементів масиву здійснюється за допомогою їх індексів. Індекс {index} кожного елемента знаходиться в діапазоні від 0 до N-1, де N – повна кількість елементів в масиві. Одновимірний масив (1D), показаний у табл. 6.1, ілюструє цю структуру.

Індекс	0	1	2	3	4	5	6	7	8	9
10-елементный массив	12	32	82	8,0	4,8	5,1	6,0	1,0	2,5	1,7

Таблиця 6.1.

Зверніть увагу, що перший елемент має індекс 0, другий - 1 і т.д. Пізніше ви побачите, що осцилограми (і інші типи даних) часто зберігаються у вигляді масивів і кожна точка осцилограми містить елемент масиву. Масиви також використовуються для зберігання даних, згенерованих в циклах, де на кожній ітерації циклу створюється один елемент масиву.

Створення елементів управління і відображення масивів

Щоб створити елементи керування та індикатори для складних типів даних, таких як масиви і кластери, необхідно послідовно виконати дві дії. Спочатку слід вирішити, чи буде ваш масив складатиметься з елементів управління або індикаторів, а потім об'єднати шаблон масиву (arrayshell) з об'єктом даних, який може бути числовим, логічним, маршрутним або строковим (а також кластерним, але про це пізніше).

Шаблон масиву знаходиться в підпалітрі Масив і кластер (Array&Cluster) палітри Елементи управління.

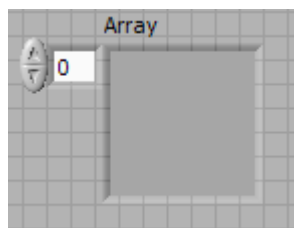


Рис. 6.1

Для створення масиву перемістіть об'єкт даних у вікно відображення елементів. Можете безпосередньо помістити туди об'єкт, клацнувши правою кнопкою миші всередині

вікна під час початкового вибору об'єкта даних з палітри Елементи управління. Вікно відображення елемента зміниться в розмірах, показуючи, що сталося узгодження типу даних, (рис. 6.2), але залишається сірим доти, поки ви не введете в нього дані. Зверніть увагу, що всі елементи масиву повинні бути або елементами управління, або індикаторами, але не їх комбінацією.

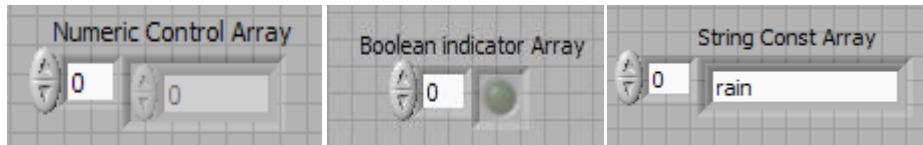


Рис. 6.2

Коли ви ставите шаблон масиву на лицьову панель, то його термінал на блок-діаграмі буде чорного кольору, що характерно для невизначеного типу даних. У терміналі показані дужки, (рис. 6.3а), які є способом відображення структури масиву в LabVIEW. Коли ви задасте масиву тип даних (помістивши елемент керування або відображення у вікно відображення елемента), термінал блок-діаграми масиву привласнить собі його колір і напис (хоча дужки зберігаються), як це показано на рис. 6.3б. Зауважте: провідники масиву товщі, ніж провідники, які переносять скалярні величини.

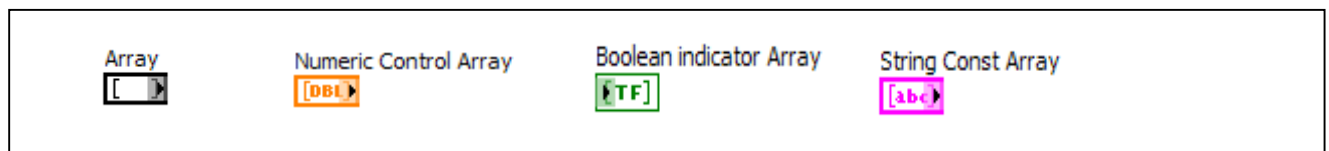


Рис. 6.3а)

б)

Ви можете ввести дані в масив, як тільки поставите їх тип. Використовуйте інструменти введення тексту або управління для введення даних або, якщо ваші дані мають числовий характер, клацніть по стрілках у вікні відображення елементів, щоб збільшити або зменшити їх значення.

Якщо потрібно змінити розміри об'єкта у вікні відображення, зверніться до інструменту переміщення і переконайтеся, що він перетворюється в стандартні дужки, коли ви ставите його в кутку вікна (можливо, вам доведеться помістити його всередину вікна, а потім трохи посувати для отримання відповідних дужок). Якщо ви хочете одночасно показати більшу кількість елементів масиву, наведіть інструмент переміщення близько кута вікна відображення до тих пір, поки не з'явиться курсор у вигляді сітки, а потім витягніть кут в горизонтальному або вертикальному напрямках (дані не зміняться через форми масиву). У цьому випадку відобразиться більшу кількість елементів. Найближча до індикатора індексу елемент завжди відповідає номеру, показаному на індикаторі. Створити масив констант на блок-діаграмі можна так само, як при створенні числових, логічних або строкових констант. Вибір опції Постійний масив (ArrayConstant) в підпалітрі Масив (Array) палітри Функції створює шаблон масиву, куди ви просто ставите відповідний тип даних (у вигляді константи). Ця можливість корисна при ініціалізації зсувних регістрів або при передачі типу даних у функції роботи з файлами або Internet.

Якщо ви захочете очистити елемент управління (відображення, константи даних) масивом, клацніть правою кнопкою миші по індикатору індексу (Але не по самому елементу) і виберіть опцію Операції з даними => Очистити масив (DataOperations =>EmptyArray).

Використання автоматичного індексування

Цикл з фіксованим числом ітерацій і цикл за умовою можуть автоматично індексувати і акумулювати масиви на їхніх кордонах шляхом додавання одного нового елемента в кожному повторенні циклу. Ця здатність називається автоіндексуванням (auto-indexing). Слід пам'ятати одну важливу річ: за замовчуванням автоіндексування включено в циклі з фіксованим числом ітерацій і вимкнено в циклі за умовою. На рис. 6.4 показано, як цикл з фіксованим числом ітерацій автоматично створює масив в його межах.

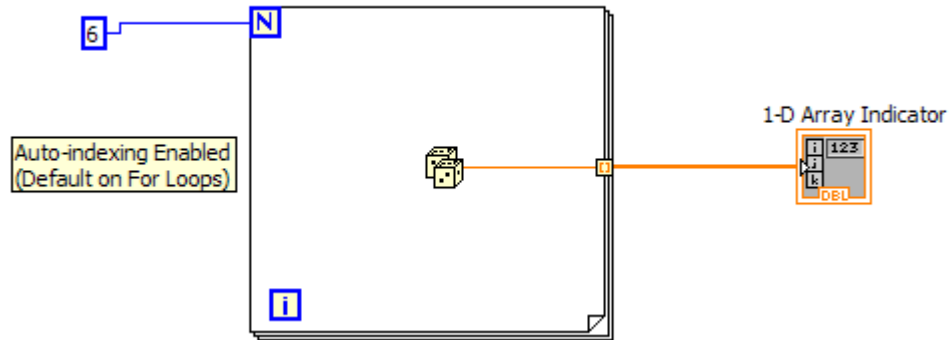


Рис. 6.4

Кожна ітерація створює наступний елемент масиву. Після завершення циклу масив виходить з нього і надходить на елемент відображення. Дані в масиві недоступні доти, поки цикл не завершиться. Зверніть увагу, що провідник стає товстим після перетворення в провідник даних масиву на кордоні циклу.

Якщо необхідно вивести скалярну величину з циклу з фіксованим числом ітерацій без створення масиву, відключіть автоіндексування шляхом виклику контекстного меню в точці вводу / виводу і вибору опції Відключити індексування (DisableIndexing).

Оскільки за замовчуванням автоіндексування в циклі за умовою не включене, то, якщо ви хочете вивести дані з цієї структури у вигляді масиву, потрібно викликати контекстне меню в точці вводу / виводу і вибрати опцію Включити індексування (EnableIndexing).

На рис. 6.6 автоіндексування відключено, і лише остання величина, яку повертає функція Випадкове число (0 - 1), виводиться з циклу. Зверніть увагу, що провідник зберігає свої розміри після виходу з циклу. Завжди дивіться на розмір провідника, оскільки автоіндексування є спільним джерелом проблем серед початківців: вони часто створюють масиви, не бажаючи цього, або не створюють їх, коли вони потрібні, а потім дивуються, чому провідник виявляється несправним.

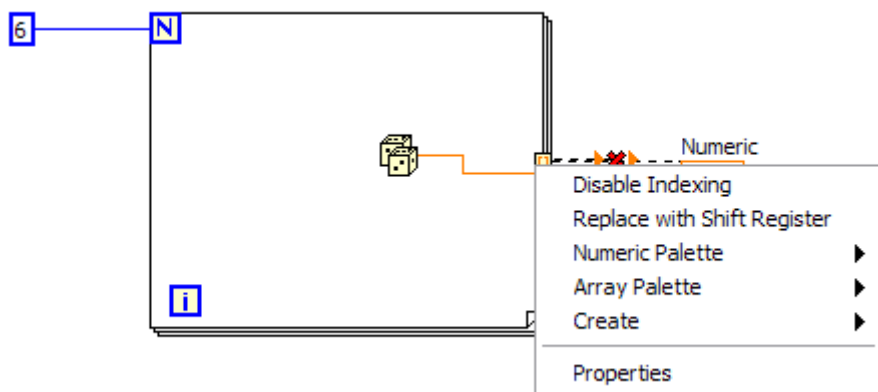


Рис. 6.5

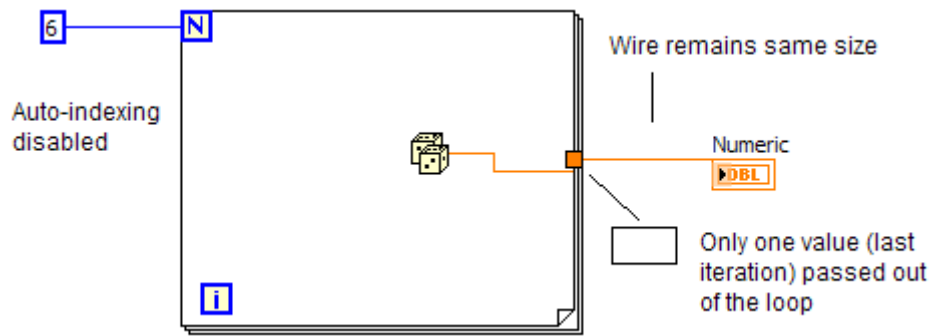


Рис. 6.6

Автоіндексування застосовується також у випадках, коли ви вводите масиви в цикли. Якщо індексування включено, як це показано в циклі (а) на рис. 6.7, то цикл буде переходити до наступного індексу масиву при кожній ітерації (зверніть увагу на потоншення провідника під час його входження в цикл). Якщо ж індексування відключено, як показано у випадку (б), то масив цілком передається в цикл.

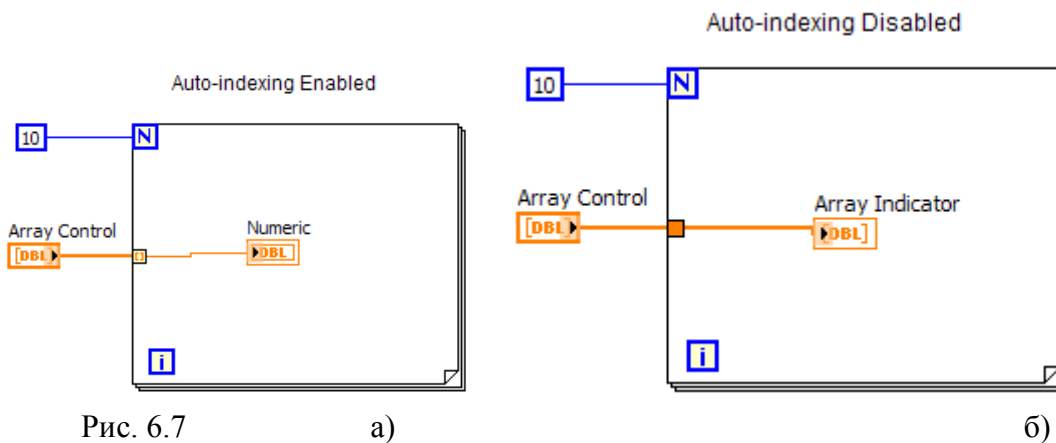


Рис. 6.7

Оскільки цикл з фіксованим числом ітерацій часто використовується для обробки масивів, в LabVIEW за замовчуванням включено Автоіндексування для цієї структури. Однак для циклу за умовою Автоіндексування за замовчуванням вимкнено. Якщо ви хочете його включити, викличте контекстне меню точки вводу / виводу масиву в цикл і виберіть опцію Включити індексування. Завжди перевіряйте стан індексації - в ньому джерело багатьох помилок.

Використання автоіндексування для встановлення кількості повторень циклів з фіксованим числом ітерацій

Коли ви дозволяєте Автоіндексування для масиву, що входить в цикл з фіксованим числом ітерацій, LabVIEW автоматично встановлює кількість повторень рівним розміру масиву, усуваючи таким чином необхідність підключення певного значення до терміналу числа ітерацій. Якщо ви задасте взаємовиключні значення, наприклад, задавши число і ввівши в дію авто індексування (або дозволивши автоіндексувати два масиви з різними розмірами), то LabVIEW встановить значення терміналу кількості ітерацій в найменший з варіантів. На рис. 6.8 показано, що розмір масиву, а не число, підключений до терміналу, визначає кількість ітерацій циклу, так як розмір масиву є найменшим з двох.

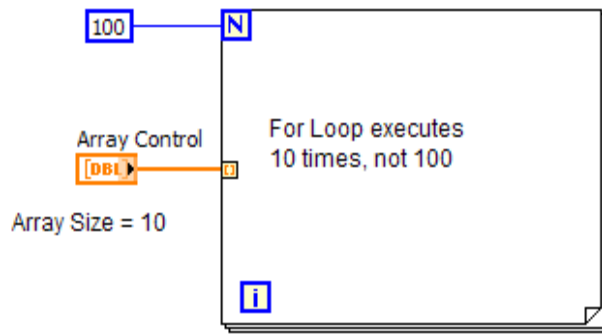


Рис. 6.8

Двовимірні масиви

Двовимірний (2D) масив зберігає елементи в структурі виду решітки. Для визначення місцезнаходження елемента необхідні два індексу: індекс по вертикалі і індекс по горизонталі, кожен з яких починається з нуля, як і всі елементи з індексами в LabVIEW. На рис. 6.9 показаний масив з чотирма рядками і шістьма стовпчиками, в якому зберігається 6x4 елементів.

Ви можете збільшити розмірність елементів управління / відображення масивів шляхом виклику контекстного меню елемента керування / відображення індексу масиву (але не елементу управління / відображення значеннями масиву) і вибору опції Додати розмірність (AddDimension). На рис. 5.10 показаний двовимірний масив цифрових елементів управління. Зверніть увагу, що тепер є два індексу для відображення кожного елемента. Ви можете використовувати інструмент переміщення у вигляді сітки для розширення вікна, що відображає значення масиву, в двох вимірах, що дозволить одночасно побачити більшу кількість елементів.

	0	1	2	3	4	5
0						
1						
2						
3						

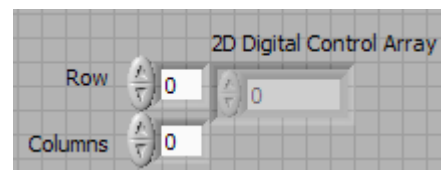


Рис. 6.10

Рис. 6.9 Масив з шістьма стовпчиками і чотирма строками - 24 елемента

Створення двовимірних масивів

Якщо ви не хочете вводити дані з лицьової панелі, то для створення двовимірного масиву можна використовувати два цикли з фіксованим числом ітерацій, один всередині іншого. Внутрішній цикл створює рядок, а зовнішній складає ці рядки для заповнення стовпців масиву. На рис. 6.11 показані два цикли з фіксованим числом ітерацій, які створюють двовимірний масив випадкових чисел із застосуванням Автоіндексування.

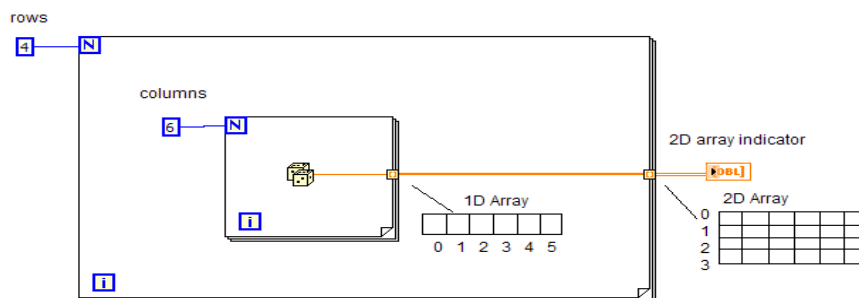


Рис. 6.11

Вправа 6.1: створення масивів за допомогою автоіндексування

Тепер у вас є можливість краще познайомитися з масивами і автоіндексуванням для роботи з ними. У цій вправі ви побачите ВП для створення масивів даних, в якому використовується авто індексування як в циклі з фіксованим числом ітерацій, так і в циклі за умовою.

1. Створіть наступний ВП. Цей прилад створює два масиви і відображає їх на лицьовій панелі, використовуючи при цьому цикл з фіксованим числом ітерацій для створення двовимірного масиву і цикл за умовою для створення одновимірного масиву. Цикл з фіксованим числом ітерацій виконується певну кількість разів, для зупинки циклу за умовою потрібно натиснути кнопку Стоп (або він зупиниться сам після 101 повторення).

2. Погляньте на лицьову панель, потім перейдіть до блок-діаграми. Зверніть увагу на те, як цикли з фіксованим числом ітерацій створюють рядки і стовпці двовимірного масиву на своїх кордонах, використовуючи автоіндексування. Зауважте також, як авто індексовані провідники товщують у міру виходу за межі циклу.

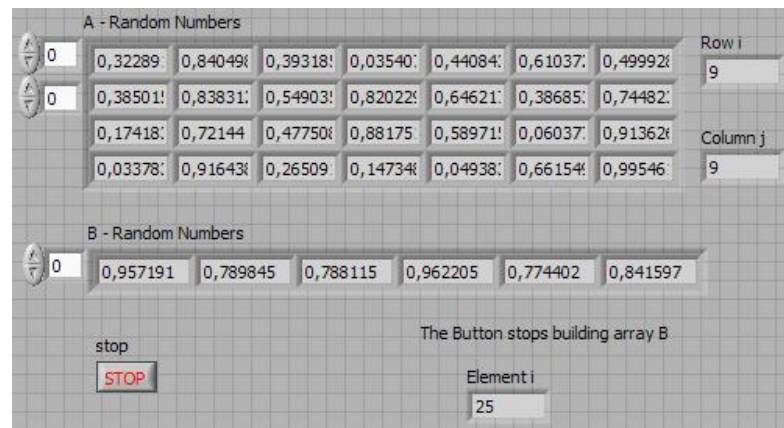


Рис. 6.12

4. Перш ніж отримати дані з циклу за умовою, клацніть правою кнопкою миші по точці вводу / виводу, що містить випадкове число, і виберіть опцію Включити індексування. Щоб побачити, як це працює, клацніть правою кнопкою миші по точці вводу / виводу і виберіть опцію Відключити індексування. Провідник, що виходить з циклу, буде розірваний. Знову клацніть правою кнопкою миші по точці вводу / виводу і виберіть опцію Включити індексування для відновлення провідника.

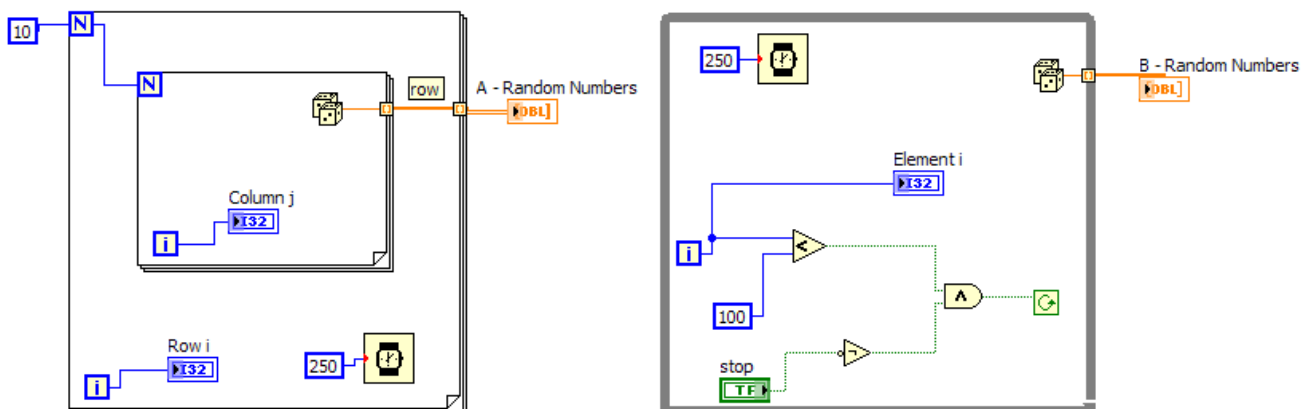


Рис. 6.13

У цьому циклі використовується логічний алгоритм, який гарантує зупинку циклу, якщо користувач не натиснув кнопку Стоп, після заданої кількості ітерацій (101 повторення).

Якщо користувач не натискає кнопку Стоп, а цикл виконався менше 101 рази, то він буде продовжувати виконуватися. Якщо будь-яке з цих умов змінюється, то цикл зупиняється.

Чому цикл повинен виконуватися 101 разів, а не 100? Згадайте, що цикл за умовою перевіряє термінал умови виходу в кінці кожного повторення. Наприкінці 100-го повторення $i = 99$ (оскільки починається з нуля), а не 100, і цикл продовжує виконуватися. Наприкінці 101-го повторення значення лічильника стає рівним 100, і цикл зупиняється (за умови, що він не перерваний натисканням кнопки Стоп).

Запустіть ВП. Не забудьте натиснути кнопку Стоп для зупинки циклу за умовою, оскільки елемент відображення на лицьовій панелі не оновиться до тих пір, поки не буде створений весь масив (значення елементів управління і відображення за межами циклу не зчитуються і не оновлюються під час його виконання).

Функції роботи з масивами

LabVIEW має багато функцій маніпулювання масивами (підпалітра Масив палітри Функції). Нагадуємо, що індекси масивів починають відраховуватися з нуля - перший елемент має індекс 0, другий - індекс 1 і т.д. У даному розділі розглядаються деякі загальні функції, але ви можете звернутися до підпалітри Масив, щоб побачити інші вбудовані функції:

- функція Ініціалізація масиву (InitializeArray) створює і заповнює всі елементи n-мірного масиву значенням за вашим вибором (рис. 6.14).

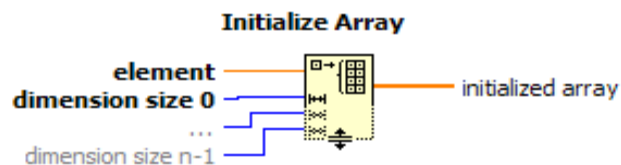


Рис. 6.14

Ви можете змінити розмірність масиву розтягуванням його меж інструментом переміщення («стрілка») для створення додаткових введів числа елементів розмірності (dimension size). Ця функція корисна для виділення пам'яті певного розміру або для ініціалізації зсувних регістрів даними з масиву; Рис. 6.15 показує, як створити за допомогою функції Ініціалізація масиву одновимірний масив із 10 елементів, в якому кожен елемент містить значення 0;

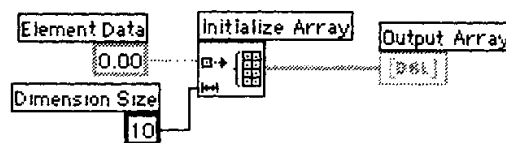


Рис. 6.15

- функція Число елементів масиву (ArraySize) повертає число елементів вхідного масиву (рис. 5.16). Якщо вхідний масив є n-мірним, то функція повертає одновимірний n-елементний масив, в якому кожен елемент містить число елементів однієї з розмірностей масиву (рис. 6.17);

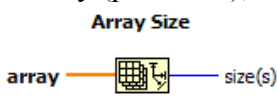


Рис. 6.16

Число елементів масиву

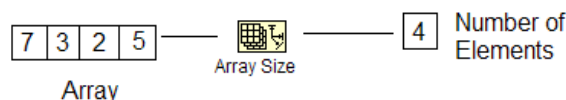


Рис. 6.17

В залежності від конфігурації, функція Створити масив (BuildArray) комбінує або об'єднує два масиви або додає в масив додаткові елементи (рис. 6.18). Будучи вперше поміщена на блок-діаграму, функція виглядає як іконка, зображена зліва. Ви можете розтягнути кордон цієї функції для збільшення кількості вводів. Функція Створити масив має два типи введення: для масиву і для елемента. Таким чином, з її допомогою легко створювати масив одночасно з масиву даних і з скалярних величин. Наприклад, функція Створити масив, зображена на рис. 6.19, об'єднує два масиви і одну скалярну величину в новий масив.

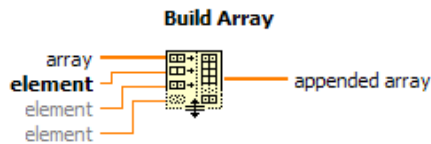


Рис. 6.18

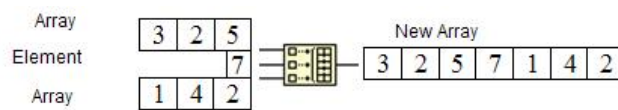


Рис. 6.19

Звертайте особливу увагу на входи функції Створити масив. Введення для масиву зображується у вигляді двох квадратиків з точками всередині, в той час як введення скалярною величини показаний порожнім квадратиком. Хоча LabVIEW адаптує введення в залежності від типу даних, які ви подаєте (масив або скаляр), ці типи даних не є взаємозамінними. Тому, не дивуйтеся, якщо у вас несподівано виникнуть несправні провідники.

Залежно від того, який тип даних підключається, введення цієї функції автоматично адаптується до вхідного елемента, будь це скаляр або масив.

За допомогою функції Створити масив можна створювати або додавати елементи в багатовимірні масиви. Елемент, що додається в багатовимірний масив, повинен бути на одну розмірність менше масиву (наприклад, допустимо додати 1D-масив в 2D-масив). Ви можете побудувати двовимірний масив, використовуючи функцію Створити масив і під'єднуючи одномерні масиви в якості елементів (кожен одновимірний масив стає рядком двовимірного масиву). Якщо ж необхідно об'єднати декілька одновимірних масивів замість створення одного двовимірного, в контекстному меню функції Створити масив виберіть опцію Об'єднати вводи (ConcatenateInputs);

- функція Підмножина масиву (ArraySubset) повертає частину масиву, який містить кількість елементів довжина (length), починаючи з індексу (index) - рис. 6.20. Не забувайте, що індексом третього елемента є 2, так як індексування починається з нуля (рис. 6.21);

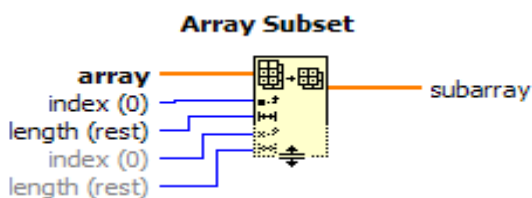


Рис. 6.20

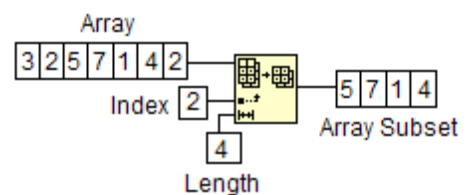


Рис. 6.21

- функція Вибірка з масиву (IndexArray) здійснює доступ до будь-якого елемента масиву. На рис. 6.22 показано, як за допомогою цієї функції здійснюється доступ до третього

елементу масиву. У даному випадку функція Вибірка з масиву витягує скалярний елемент з масиву. Щоб витягти окремий скалярний елемент, з'єднайте індекс рядка потрібного елемента з верхнім входом, а індекс стовпця - з нижнім входом функції.

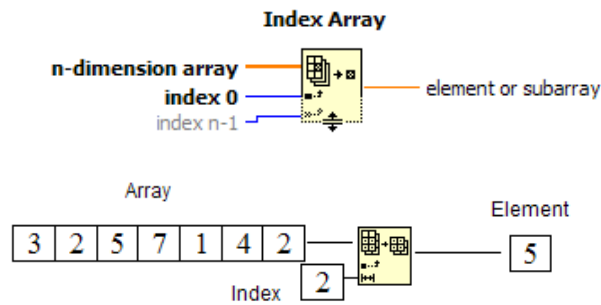


Рис. 6.22

Ви також можете використовувати цю функцію для «вирізування» з масиву рядки, стовпці або скалярного елемента. Щоб це зробити, залиште один з введів функції непідключеним. При вирізання рядки двовимірного масиву з'єднайте значення індексу вибраного рядка з першим індексним введенням функції. Щоб відокремити стовпець двовимірного масиву, залиште введення першого індексу непідключеним і з'єднайте значення індексу вибраного стовпця з другим індексним вводом.

Зверніть увагу, що, коли ви залишаєте введення непідключеним, символ відповідного індексу зображується у вигляді порожнього прямокутника. На рис. 6.23 показаний процес вирізання шпальти і рядки з двовимірного масиву.

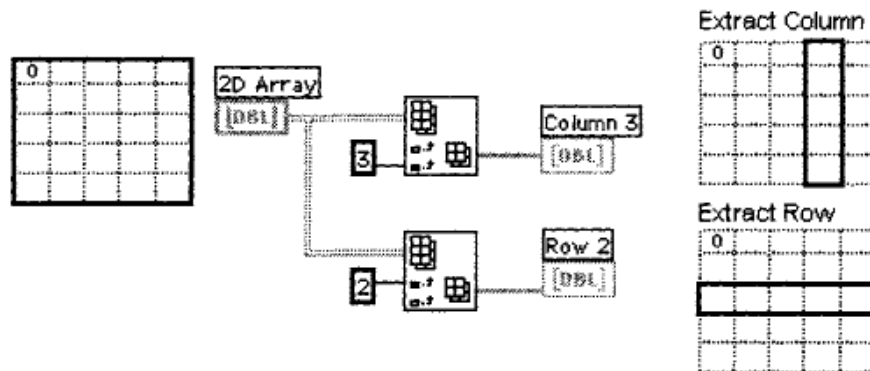


Рис. 6.23

Вправа 6.2: робота з масивами

У цій вправі ви створите ВП, що поєднає два масиви, а потім вирізує елемент з середини нового об'єднаного масиву.

1. Створіть наступний прилад. Лицьова панель цього приладу містить два вхідних масиви (кожен показує три елементи), два цифрових елементу управління і вихідний масив (показує вісім елементів).

Для створення нового масиву ВП об'єднає два масиви і значення елементів управління в такій послідовності:

Початковий масив + елемент 1 + елемент 2 + завершальний масив.

Лицьова панель вже створена. Закінчить побудову блок-діаграми. Елементи управління і відображення масивами матимуть сірий колір, поки ви або програму не введете в них які-небудь дані (рис. 6.24).

2. Побудуйте блок-діаграму, як показано на рис. 6.25. Використовуйте вікно контекстної допомоги для знаходження відповідних терміналів функцій

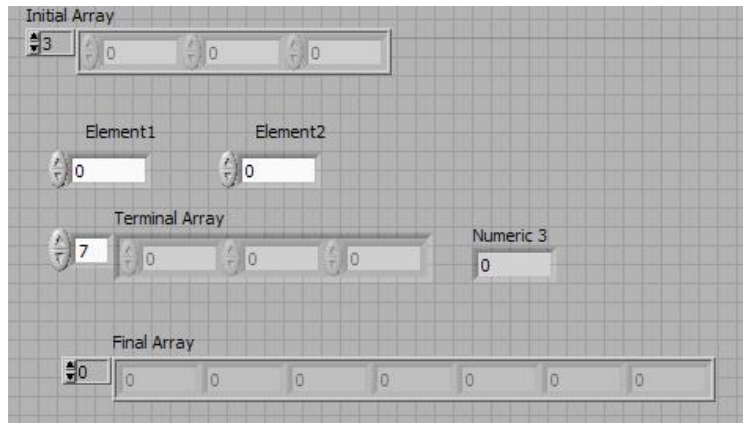


Рис. 6.24

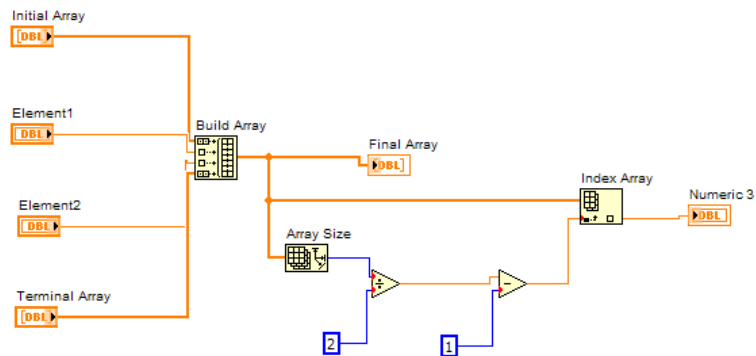


Рис. 6.25

Функція Створити масив (палітра Масив) в цій вправі об'єднує вхідні дані для створення нового масиву в такій послідовності:

Початковий масив + елемент 1 + елемент 2 + завершальний масив.

Встановіть інструмент переміщення в нижньому правому куті функції і змініть її розміри так, щоб вона включала чотири введення. Вводи автоматично визначають, який тип даних ви підключаєте: масив або скаляр.

Функція Число елементів масиву (палітра Масив) повертає число елементів в об'єднаному масиві.

Функція Вибірка з масиву повертає елемент з середини масиву.

LabVIEW створює масив за допомогою функції Створити масив. Потім обчислюється індекс середнього елемента масиву, рівний половині довжини масиву мінус одиниця (відлік починається з нуля). Оскільки масив має парну кількість елементів, то серединним буде один з двох середніх елементів.

Поверніться до лицьовій панелі і запустіть ВП. Спробуйте зробити кілька різних комбінацій вхідних даних. Збережіть ВП.

Поліморфізм

Іншим корисним якістю LabVIEW є поліморфізм (polymorphism) його арифметичних функцій: додавання (Add), Множення (Multiply), Поділу (Divide) і т.д. Поліморфізм - це довге слово для назви простого принципу: вхідні дані функцій можуть мати різні розмірності і уявлення. Наприклад, використовуючи одну і ту ж функцію, ви можете скласти скалярну величину і масив або скласти два масиви. На рис. 5.26 показані деякі поліморфічне комбінації функції Додавання.

Результатом першого комбінації є скалярна величина. У другій комбінації скалярна величина додається до кожного елементу масиву. У третій комбінації кожен елемент масиву

додається до відповідного елементу іншого масиву. У всіх випадках використовується та ж сама функція Додавання, але вона виконує різні види дій.

Як показано на рис. 6.27, на кожному повторенні циклу з фіксованим числом ітерацій виробляється одне випадкове число (в діапазоні від 0 до 1), яке зберігається в масиві на кордоні циклу.

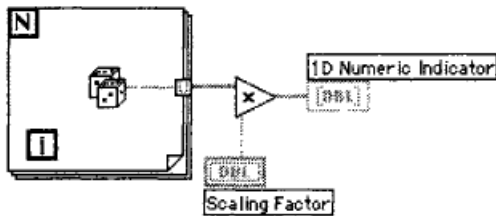


Рис. 6.26

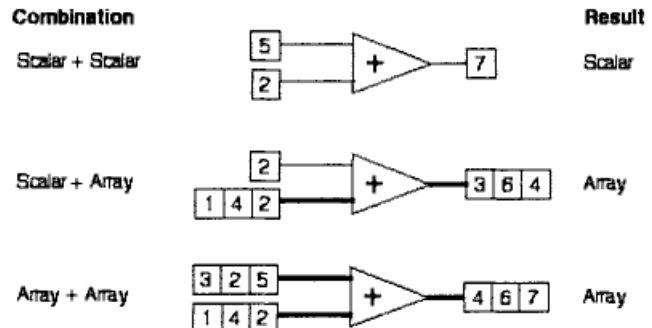


Рис. 6.27

Після того як цикл завершиться, функція Множення примножує кожен елемент масиву на заданий вами масштабний коефіцієнт. Елемент відображення масиву на лицьовій панелі покаже масштабований масив.

На рис. 6.28 представлені деякі можливі поліморфічне комбінації функції Додавання.

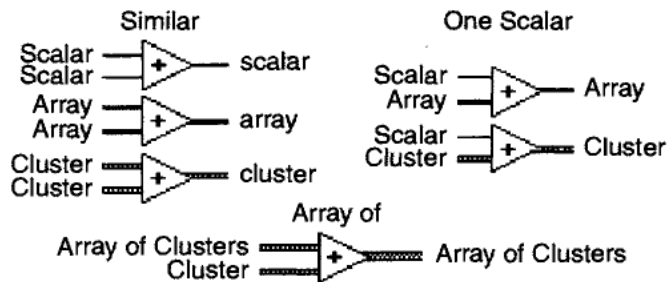


Рис. 6.28

Якщо ви виконаєте арифметичні дії з масивами з різною кількістю елементів, то результуючий масив буде мати розмірність найменшого з двох. Іншими словами, LabVIEW виконує дії над відповідними елементами двох масивів до тих пір, поки елементи одного з них не закінчаться. Залишилися елементи довшого масиву ігноруються.

Вправа 6.3: поліморфізм на прикладі масивів.

У цій вправі ви створите ВП, який демонструє поліморфізм на прикладі масивів.

1. Відкрийте нову панель і створіть ВП, показаний на рис. 6.29. Спочатку створіть два масиви, насамперед вибравши шаблон масиву в підпалітрі Масив і кластер палітри Функції. Потім помістіть числовий елемент відображення у вікно шаблону. Для того щоб побачити кілька елементів масиву, ви повинні пересунути кут вікна відображення елементів за допомогою сіткового курсору інструменту переміщення. Ви можете зробити видимими одночасно велику кількість елементів одновимірного масиву, витягаючи межу вікна в горизонтальному або у вертикальному напрямку.

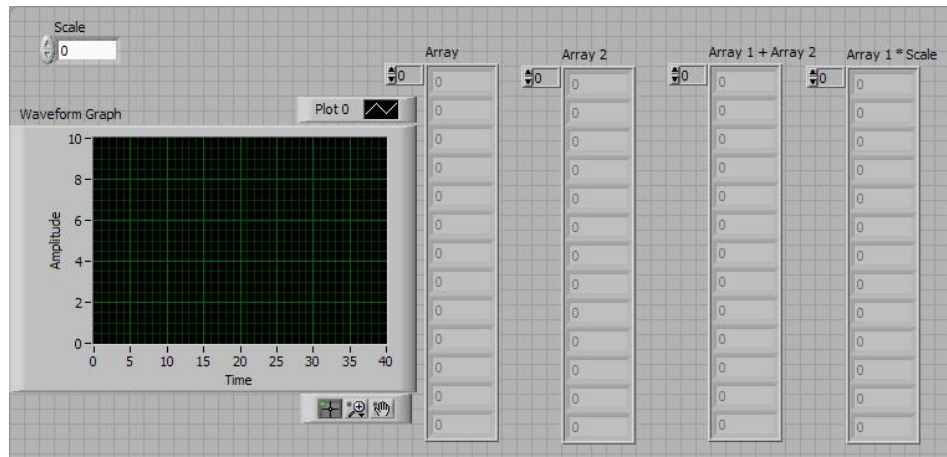


Рис. 6.29

Всі чотири масиви в цій вправі містять елементи відображення. Надайте їм особливі ярлики, щоб надалі їх не переплутати. Якщо ви все-таки забудете, який об'єкт лицьовій панелі відповідає якому терміналу блок-діаграми, то просто клацніть правою кнопкою миші по будь-якому з них і виберіть опцію Знайти термінал (FindTerminal) або Знайти індикатор (FindIndicator), і LabVIEW виділить відповідний об'єкт.

2. Після того як ви створили масиви, виберіть функцію Графік осцилограми, в підпалітрі Графік (Graph) палітри Елементи управління.

3. Не забудьте створити елемент управління масштабом.

4. Побудуйте блок-діаграму, як показано на рис. 6.30.

У циклі з фіксованим числом ітерацій автоіндексування задіяне за умовчанням, тому масиви будуть створюватися автоматично.

5. Функції Додавання, Множення і Випадкове число (0-1) знаходяться в палітрі Числові.

6. Використовуйте функцію Створити масив з палітри Масив. Збільште її за допомогою інструменту переміщення таким чином, щоб вона мала чотири введення. За замовчуванням вводи функції не об'єднуються. На виході функції Створити масив буде двовимірний масив. Кожен вхідний масив стає рядком, так що вихідний двовимірний масив складається з чотирьох рядків і десяти стовпців.

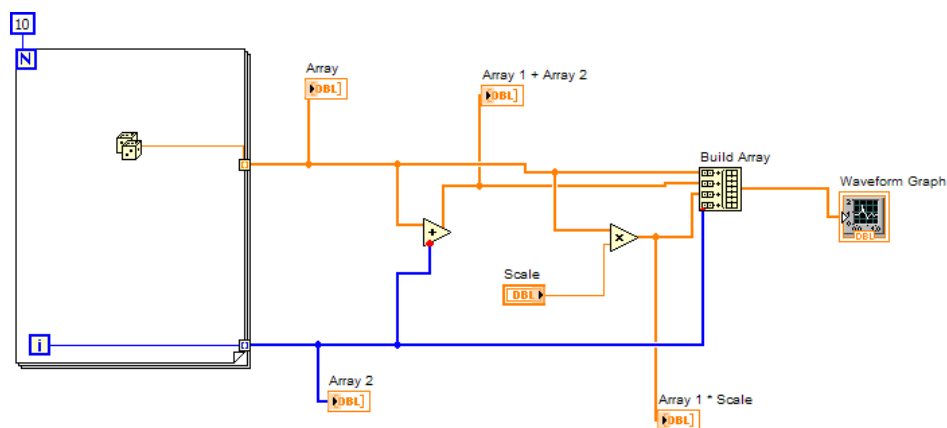


Рис. 6.30

7. Запустіть ВП. На графіку кожен елемент масиву розташовується навпроти свого індексу одночасно для всіх чотирьох масивів: дані масиву 1, дані масиву 2, дані масиву 1 * масштаб і масив1 + масив2.

Ці результати демонструють кілька видів використання поліморфізму в LabVIEW. Наприклад, масив 1 і масив 2 можуть бути вхідними осцилограмами, які потрібно масштабувати.

Варіанти індивідуальних завдань:

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 палітрі 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, що містить 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 тих пір, поки не натиснута кнопка на лицьовій 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 DecimateIDArray, розташовану в палітрі 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 в масиві слід використовувати функцію Sort 1DArray, розташовану на палітрі Functions »Array

7. Створіть ВП, який генерує одновимірний масив випадкових чисел і виводить максимальне значення отриманого масиву і його порядковий номер. Використовувати функцію ArrayMax&Min, розташовану на палітрі Functions »Array

8. Створіть ВП, який генерує одновимірний масив випадкових чисел і виводить мінімальне значення отриманого масиву і його порядковий номер. Використовувати функцію ArrayMax&Min, розташовану на палітрі Functions »Array

9. Створіть ВП, який генерує двовимірний масив випадкових чисел, що містить 4 рядки і 5 стовпців

10. Створіть ВП, який генерує два одновимірних 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

Контрольні питання

1. Що таке автоіндексування, для чого воно застосовується?
2. Як задати одновимірний і двовимірний масив в LabView?
3. Як додати дані в багатовимірний масив?
4. Як отримати доступ до будь-якого елемента масиву?
5. Опишіть принцип поліморфізму
6. Для чого використовується функція Складова арифметика?

Лабораторна робота №7

Складові дані LabView: кластери

Мета – вивчення складних складових типів даних -кластерів.

Завдання:

- Навчитися використовувати кластери, а також, розділяти і об'єднувати їх.
- Зрозуміти, чим кластери відрізняються від масивів

Усе про кластери

Як і масив, кластер (cluster) є структурою, яка групує дані. Однак, на відміну від масиву, кластер може групувати дані різних типів (числові, логічні і т.д.). Це поняття аналогічно struct в мові програмування C або об'єктам даних, визначеним як елементи класу, в C++ або Java. Кластер може бути подумки представлений у вигляді зв'язки проводів, як в телефонному кабелі. Кожен провід в кабелі представляє елемент кластера. Оскільки кластер має тільки один «провід» на блок-діаграмі (незважаючи на те, що по ньому проходить безліч даних різних типів), кластери зменшують нагромадження провідників і кількість терміналів підключення, необхідних для підпрограм (рис. 7.32). Надалі ви виявите, що дані у вигляді кластерів часто з'являються під час їх виведення на графіки та діаграми.

Доступ до елементів кластера можна отримати шляхом їх повного поділу (unbundling) або поділу за індексом елемента. Метод поділу залежить від обраної вами функції і має свою сферу застосування. Поділ елементів кластера можна уявити як розщеплення різнокольорових дротів в телефонному кабелі. На відміну від масивів, які можуть динамічно змінювати розмір, кластери мають фіксований розмір або фіксована кількість проводів (рис. 7.1).

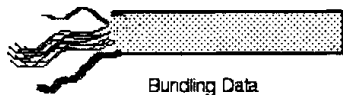


Рис. 7.1 Об'єднання даних в кластер

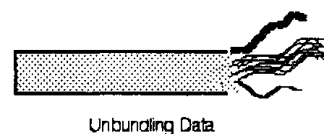


Рис. 7.2 Поділ кластера

Ви можете з'єднати термінали кластерів тільки в тому випадку, якщо вони мають однаковий тип; іншими словами, обидва кластера повинні мати однакову кількість елементів і відповідні елементи повинні збігатися як по типу даних, так і за їх порядку. Принцип поліморфізму може бути також застосований і до кластерів за умови збігу типу даних.

Кластери часто зустрічаються при обробці помилок. На рис. 7.3 показані кластери помилок, ErrorIn.ctl і ErrorOut.ctl, які використовуються LabVIEW для передачі введень про помилки серед безлічі віртуальних приладів на блок-діаграмі (наприклад, більшість ВП збору даних і ВПП введення / виводу в файл / з файлу мають вбудовані кластери обробки помилок). Ці кластери помилок застосовуються настільки часто, що їх спеціально виділили для більш легкого доступу до них - відповідні ВП є в підпалітрі Масив і кластер палітри Елементи управління.

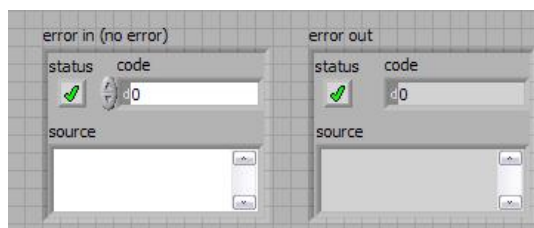


Рис. 7.3

Створення елементів управління і відображення для кластерів

Створіть кластер, помістивши шаблон Кластер (Cluster) з підпалітри Масив і Кластер на лицьову панель. Ви тепер можете розмістити будь-який об'єкт лицьовій панелі усередині кластера. Як у випадку з масивами, допустимо помістити об'єкти всередину кластера безпосередньо при витяганні їх з палітри Елементи управління або перетягнути існуючий об'єкт і помістити його в кластер. Об'єктами всередині кластера можуть бути тільки елементи управління або тільки індикатори. Не можна помістити елементи керування та індикатори в одному кластері, так як сам кластер повинен бути або першим, або другого типу.

Кластер стає елементом управління або індикатори в залежності від типу першого внесеного в нього об'єкта. У разі необхідності можна змінити розміри кластера за допомогою інструменту переміщення. На рис. 7.4 зображений кластер з чотирма елементами управління.

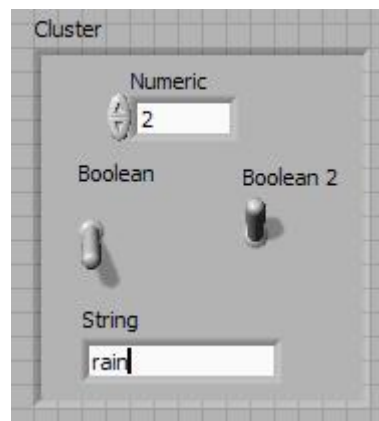


Рис. 7.4

Створити кластер з констант на блок-діаграмі ви можете в два етапи аналогічно масивам.

Якщо ви хочете, щоб візуальні розміри кластера відповідали об'єктам всередині нього, клацніть правою кнопкою миші по його кордоні (не всередині кластера) і виберіть опцію Автоматичне встановлення розміру (Autosizing).

Упорядкування елементів кластера

Елементи кластера мають логічний порядок незалежно від їх місця розташування в рамках шаблону. Перший об'єкт, поміщений в кластер, стає нульовим елементом, другий - першим і т.д. При видаленні одного елемента порядок автоматично змінюється. Якщо ви хочете з'єднати один кластер з іншим, то повинні чітко відстежувати порядок елементів вашого кластера, оскільки порядок і тип даних повинні бути ідентичними. Крім цього, при необхідності відразу розділити весь кластер потрібно знати, яке значення відповідає якому виходу у функції Розділити (Unbundle).

Змінити порядок в межах кластера можна, викликавши контекстне меню на його кордоні і вибравши опцію Змінити порядок елементів кластера (ReorderControlsinCluster). На інструментальній панелі з'являється новий набір кнопок, а зовнішній вигляд кластера змінюється (рис. 7.5).

Білі вікна елементів показують їх поточні порядкові номери, чорні вікна - нові. Клацання по елементу присвоює йому номер, відображений на інструментальній панелі. Ви можете ввести новий номер в це поле, перш ніж клацнути по об'єкту.

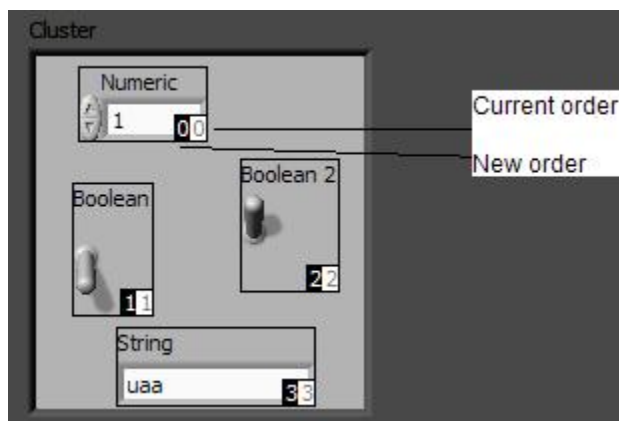


Рис. 7.5

Об'єднання даних

Функція Об'єднати (Bundle) з палітри Кластер об'єднує окремі компоненти в кластер або дозволяє замінити елементи в існуючому кластері (рис. 7.6). Коли ви розміщуєте функцію на блок-діаграму, вона має вигляд іконки зліва. Ви можете збільшити кількість входів, витягаючи кут функції інструментом переміщення. При підключенні до кожного вводу в його порожньому полі з'являється символ, що вказує на тип підключаються даних. Порядком елементів результуючого кластера буде порядок входів у функцію Об'єднати

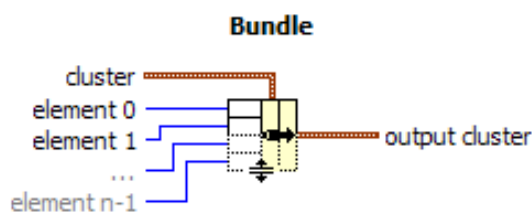


Рис. 7.6

Якщо ви хочете лише створити кластер, немає необхідності підключати будь-який провідник до центрального входу кластера (з назвою кластер - cluster) функції Об'єднати. Але цей вхід слід підключати, якщо ви замінюєте елемент кластера.

Будьте обережні при роботі з описаною функцією. Якщо ви додасте один елемент у кластер без попередньої підгонки розмірів функції Об'єднати на блок-діаграмі, програма не буде працювати.

Заміна елемента кластера

Якщо ви хочете замінити який-небудь елемент в кластері, то спочатку змініть розмір функції Об'єднати таким чином, щоб вона містила стільки ж вхідних терміналів, скільки елементів містить кластер (в іншому випадку ви отримаєте розірваний провідник). Потім з'єднайте кластер із середнім терміналом функції Об'єднати (на входах функції Об'єднати з'являться символи типів даних елементів усередині кластера). Після цього підключіть нові значення до відповідних входів функції. Потрібно підключитися тільки до тих терміналів, елементи яких в кластері підлягають зміні.

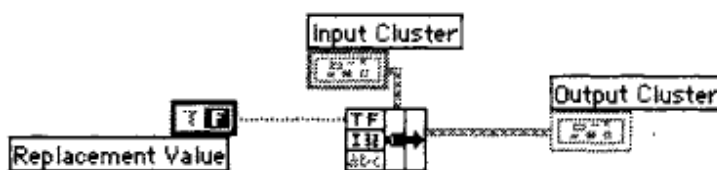


Рис. 7.6

Поділ кластерів

Функція Розділити (Unbundle) з палітри Кластер розділяє кластер на компоненти. Вихідні компоненти розташовані зверху вниз в тому ж порядку, що і в кластері. Якщо компоненти належать до одного типу, то порядок елементів в кластері є єдиним способом їх розрізнення. Коли ви ставите функцію на блок-діаграму, вона має вигляд іконки, зображеної ліворуч. Збільшити число виходів можна витягуванням кута функції інструментом переміщення. Вам необхідно змінити розмір функції Розділити, щоб вона містила стільки ж виходів, скільки є елементів у вхідному кластері, інакше з'явиться розірваний провідник. Коли ви підключаєте кластер до функції Розділити з правильними розмірами, то в раніше порожніх вихідних терміналах з'являться символи типів даних кластеру.

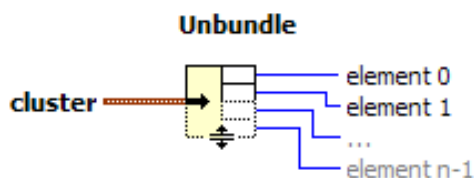


Рис. 7.7

Будьте обережні, користуючись цією функцією. Якщо ви додасте один елемент у кластер без попередньої підгонки розмірів функції Розділити на блок-діаграмі, то програма не працюватиме!

Вправа 7.1: робота з кластером

У цій вправі ви створите ВП для навчання роботі з кластерами. Ви створите кластер, розділіть його на елементи, потім знову з'єднайте елементи і відобразіть їх значення в новому кластері.

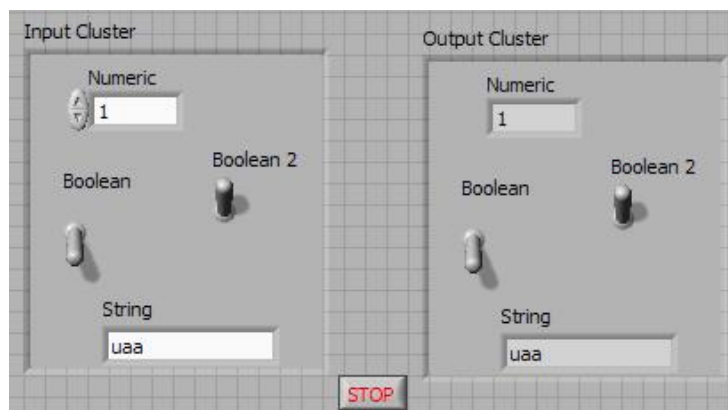


Рис. 7.8

1. Відкрийте нову панель і помістіть в неї шаблон Кластер (палітра Масив і кластер). Назвіть його Вхідний кластер. Збільште шаблон (переконайтеся, що ви захопили кордон кластера, - в іншому випадку нічого не станеться).

2. Помістіть всередину шаблону Вхідний кластер цифровий елемент управління, два логічних перемикача і елемент управління рядками.

3. Тепер створіть шаблон Вихідний кластер шляхом копіювання вхідного. Клацніть правою кнопкою миші по об'єкту в новому кластері (або на кордоні кластера) і виберіть опцію Замінити на індикатор (ChangeToIndicator). Змініть також ярлик кластеру. Ця методика дозволяє отримати правильний порядок в кластері і є досить ефективною.

Ви можете створити вихідний кластер точно таким же способом, як і вхідний, використовуючи елементи відображення замість елементів управління (уважно стежте за

порядком розміщення елементів в кластер - він повинен збігатися з порядком елементів вхідного кластера).

4. Переконайтеся, що елементи вхідного і вихідного кластерів мають однаковий порядок. Це можна зробити, викликавши контекстне меню на кордоні кожного кластера і вибравши опцію Змінити порядок елементів кластера. Якщо порядки елементів кластерів різні, то зробіть один з них аналогічним іншому.

5. Помістіть на лицьовій панелі прямокутну кнопку Стоп (StopButton) з палітри Логічні. Зверніть увагу, що за замовчуванням стан цієї кнопки - БРЕХНЯ. Не міняйте цього стану.

6. Побудуйте блок-діаграму, як показано на рис. 7.9. Відзначте той факт, що, хоча кожен кластер містить чотири об'єкти, на блок-діаграмі ви побачите лише по одному терміналу на кластер. Не забудьте змінити функцію терміналу умови виходу з циклу за умовою так, щоб вона була Зупинити, якщо Істина (це можна зробити викликом контекстного меню терміналу умови виходу).

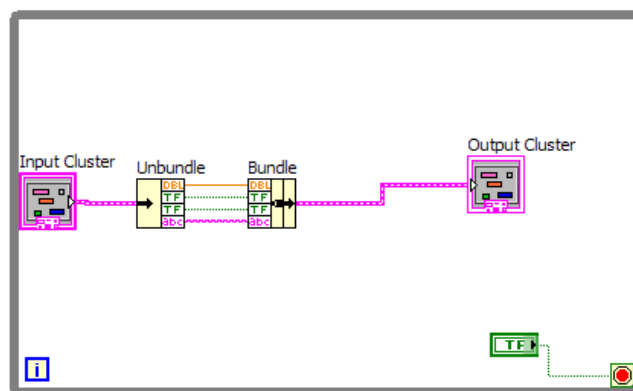


Рис. 7.9

Функція **Розділити** розділяє кластер. Після цього ви отримуєте доступ до окремих елементів кластера. Змініть її розмір таким чином, щоб вона мала чотири виходи. Як тільки ви підключите до функції **Розділити** вхідний кластер, в її терміналах з'являться символи типів даних.

Функція **Об'єднати** відтворює кластер. Змініть його розмір, щоб вона мала чотири входи.

7. Поверніться до лицьовій панелі і запустіть ВП. Введіть різні значення елементів кластера управління і простежте, як кластер відображення повторює ці значення.

Натисніть кнопку Стоп, щоб зупинити виконання.

8. Закрийте і збережіть ВП як ClusterExercise.vi

Об'єднання і поділ на ім'я

Часто не потрібно розділяти весь кластер - потрібен лише один або два елементи. Для цього можна скористатися функціями **Об'єднати за ім'ям** (BundlebyName) і **Розділити по імені** (UnbundlebyName).

Функція Об'єднати за ім'ям (рис. 7.10), яка також знаходиться в палітрі Кластер, повертає елементи, імена яких ви задали. Вам не потрібно думати про порядок елементів кластера або коригувати розмір функції Розділити один елемент в існуючому кластері. На відміну від функції Об'єднати, середньому вхідному терміналу функції Об'єднати по імені завжди необхідне підключення, щоб повідомити функції, в якому кластері потрібно замінити елемент.



Рис. 7.10 Функція Об'єднати за ім'ям Рис. 7.11 Функція Розділити за ім'ям

Функція Розділити за ім'ям(рис. 7.11), яка також знаходиться в палітрі Кластер, повертає елементи, імена яких ви задали. Вам не потрібно думати про порядок елементів кластера або коригувати розмір функції Розділити

Коли ви використовуєте функції поділу чи об'єднання по імені, кожен елемент кластера повинен мати ярлик. Інакше ви не отримаєте доступу до потрібного елементу: LabVIEW ж не знає, який елемент ви хочете вибрати!

Наприклад, якщо вам захочеться замінити значення boolean2 в останній вправі, можете скористатися функцією Об'єднати за ім'ям, не турбуючись про порядок елементів кластера або розмір функції.

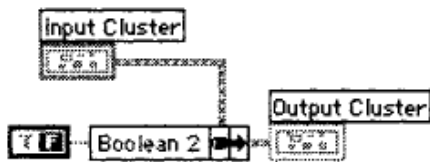


Рис. 7.12



Рис. 7.13

Точно так само при необхідності отримати доступ до значення рядка слід скористатися функцією **Розділити по імені**.

Як тільки ви підключите вхідний кластер до функції **Об'єднати по імені** або **Розділити на ім'я**, ім'я першого елемента кластера з'явиться у вікні введення або виведення функції.

Щоб отримати доступ до іншого елемента, клацніть інструментом управління («палець») або введення тексту (A) по вводу або висновку. Ви побачите перелік імен всіх елементів кластера, що мають ярлики. Виберіть потрібний, елемент зі списку, і його ім'я з'явиться в терміналі. Ви також можете отримати доступ до цього переліку, клацнувши правою кнопкою миші по імені і вказавши функцію Вибрати об'єкт (виберіть елемент).

Дозволяється змінити розміри обох функцій, щоб вони відображали стільки елементів, скільки вам необхідно. Після цього можете вибрати кожен компонент для індивідуального доступу. Не потрібно турбуватися про те, що програма перестане працювати при зміні елементів кластера: функції об'єднання і розділення по імені не втрачають працездатності, поки ви не видалите елемент, на який вони посилаються.

Вправа 7.2: ще раз про кластери

У цій вправі ви створите ВП, який перевіряє, є значення числового елемента керування Число 1 у вхідному кластері більшим чи рівним нулю. Якщо воно менше 0, то ВП вираховує абсолютні величини всіх елементів управління. Якщо значення Число 1 більше або дорівнює 0, то ВП залишить значення всіх елементів управління без зміни. Незалежно від значення Число 1 віртуальний прилад примножує всі величини на 0,5 і відображає результати у вихідному кластері, демонструючи використання поліморфізму при роботі з кластерами.

1. Відкрийте нову лицьову панель і помістіть на неї шаблон Кластер (палітра Масив і кластер). Назвіть його Вхідний кластер.

2. Створіть елементи управління Число 1, Число2 і Повзун з палітри Числові, відразу поміщаючи їх усередині шаблону кластеру. Створювати їх потрібно в певному порядку (оскільки вам доведеться з'єднати вхідний кластер з вихідним) і відразу присвоювати їм імена.

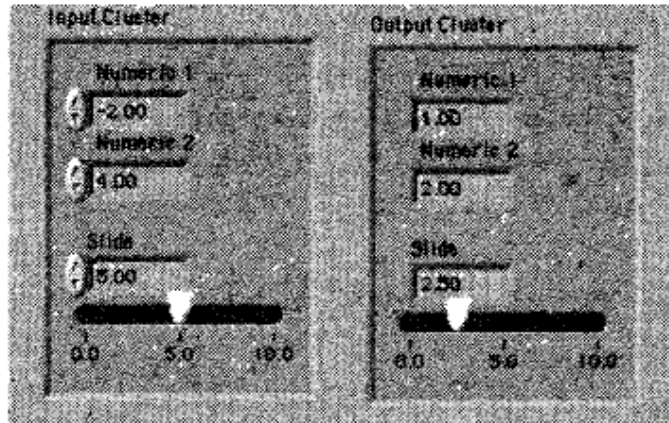


Рис. 7.14

3. Створіть таким же чином шаблон Вихідний кластер, використовуючи елементи відображення (всі елементи повинні бути введені в кластер в тому ж порядку). Ви можете створити цей кластер за допомогою копіювання вхідного кластера з подальшою зміною його ярлика.

5. Побудуйте блок-діаграму, як показано на рис. 7.15. У ній повинні бути задіяні обидва варіанти БРЕХНЯ і ІСТИНА структури варіанту.

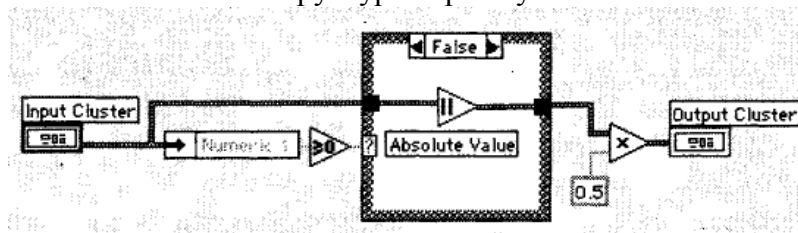


Рис. 7.15

Функція **Розділити за ім'ям** виділяє значення елемента управління Число 1 з вхідного кластера, і тепер ви можете порівняти його з нулем. Якщо Число 1 не з'явилось на виході функції, клацніть інструментом управління по цьому полю і виберіть його зі списку елементів кластера.

Функція **Більше або дорівнює 0?** (Більше або дорівнює 0?) Повертає значення ІСТИНА, якщо вхідне значення більше або дорівнює нулю.

Функція **Абсолютна величина** (абсолютне значення) повертає вхідне значення, якщо воно більше або дорівнює 0, або значення, протилежне вхідного, якщо воно менше 0. У цій праві функція бере абсолютну величину значень елементів всього кластера.

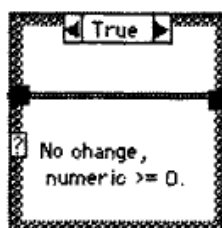


Рис. 7.16

5. Запустіть ВП. Спробуйте ввести позитивне і негативне значення елемента Число 1. Зверніть увагу на використання поліморфізму для одночасного множення всіх значень в кластері на 0,5 і знаходження абсолютної величини всього кластера.

6. Збережіть ВП як ClusterComparison.vi

Взаємозамінні масиви і кластери

Іноді зручно поміняти масиви на кластери і навпаки. Наприклад, є кластер кнопок на лицьовій панелі, і ви хочете поміняти порядок кнопок на протилежний. Функція Звернення індексів одновимірного масиву (Reverse1DArray) виконала б цю операцію, але вона працює тільки з масивами. Ви можете використовувати функцію Кластер в масив (кластерів Array) для перетворення кластера в масив. Зворотна операція здійснюється за допомогою функції Масив в кластер (Array кластера) - рис. 7.17



Рис. 7.17. Функції Масив в кластер і Кластер в масив

Функція Кластер в масив конвертує кластер з кількістю елементів N одного типу даних в масив з кількістю елементів N того ж типу даних. Індекс масиву відповідає порядковому номеру в кластері (тобто нульовий елемент кластера стає значенням масиву з індексом 0). Не можна застосовувати цю функцію в кластері, що містить масиви в якості елементів, так як LabVIEW не дозволяє створювати масиви, що складаються з масивів. Також зверніть увагу, що при використанні цієї функції всі елементи в кластері повинні бути одного типу.

Функція Масив в кластер перетворює одновимірний масив з числом елементів N в кластер з числом елементів N того ж типу даних. Для включення цієї функції ви повинні натиснути правою кнопкою миші по терміналу Масив в кластер і вибрати опцію Розмір кластера (ClusterSize) для встановлення розміру вихідного кластера, оскільки кластери, на відміну від масивів, не встановлюють свій розмір автоматично. Розмір кластера за замовчуванням дорівнює 9. Якщо ваш масив має меншу кількість елементів, ніж це визначено розміром кластера, LabVIEW автоматично створить додаткові елементи кластера 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 ви не відрегулюєте його розмір.

Обидві функції зручні у використанні, особливо коли потрібно показати елементи кластера на лицьовій панелі, але при цьому необхідно управляти індексами елементів на блок-діаграмі. Названі функції знаходяться в підпалітрах Масив і кластер палітри Функції.

Вправа 7.1. ВП Робота з кластерами

Лицьова панель

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

а) помістіть на лицьову панель кнопку «Стоп» та круглий світлодіод;

б) з палітри Controls »масиву і кластера виберіть шаблон кластера;

в) об'єкти лицьовій панелі, показані на ілюстрації, помістіть в шаблон кластера;

г) створіть і перейменуйте копію елемента Кластер в Модифікований Кластер.

Після цього клацніть правою кнопкою миші по межі шаблону кластера Модифікований Кластер і виберіть з контекстного меню пункт Змінити індикатору;

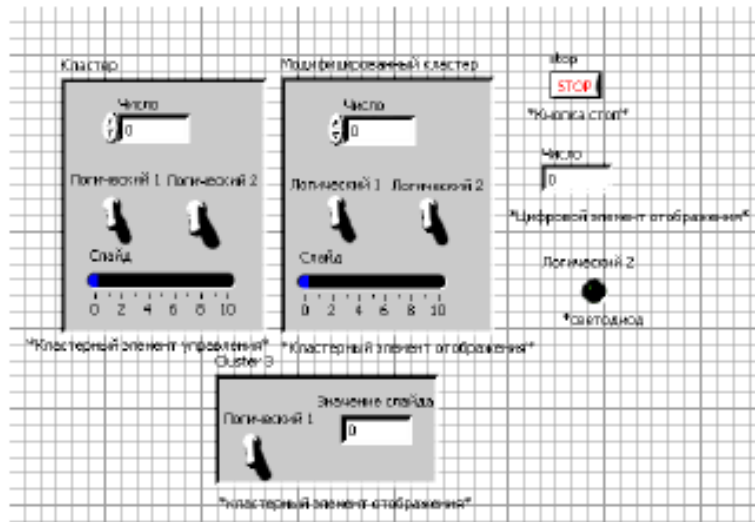


Рис. 7.18

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

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

- а) клацніть правою кнопкою миші по межі шаблону кожного кластера, з контекстного меню виберіть пункт Reorder управління в кластері;
- б) порядкові номери елементів встановіть, як показано нижче на малюнку.

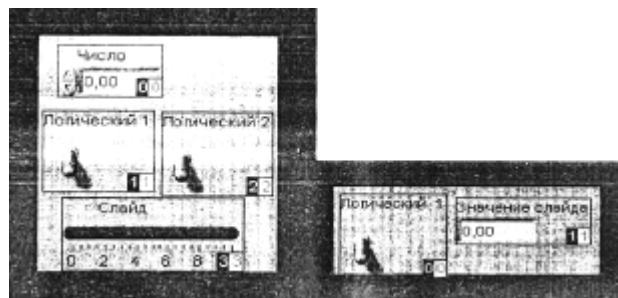


Рис. 7.19

3. Створіть блок-діаграму, як показано нижче.

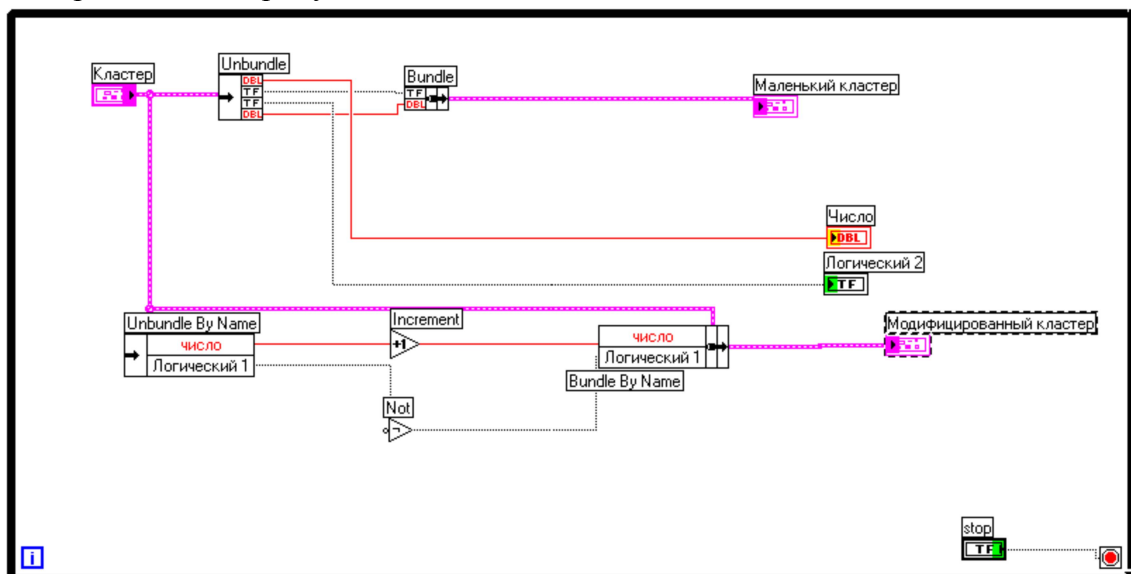


Рис. 7.20

З палітри Функції ->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 Функції ->Cluster виберіть функцію Bundle. Ця функція об'єднає елементи в кластер Маленький кластер.

З палітри Функції ->Cluster виберіть функцію Unbundle по імені. Ця функція виділить два елементи з кластера Кластер. Змініть розмір функції до двох полів виведення даних. Якщо імена в полях виведення даних відрізняються від показаних на ілюстрації, слід натиснути правою кнопкою миші по імені елемента і в контекстному меню увійти в розділ SelectItem.

З палітри Функції ->Цифрові виберіть функцію приросту. Ця функція додасть 1 до значення елемента Число.

З палітри Функції ->Булеві виберіть функцію Not. Ця функція видасть логічне заперечення елемента Логічний 1.

З палітри Функції ->Cluster виберіть функцію Bundle по імені. Ця функція змінить значення елементів Число і Логічний в кластері Кластер і створить кластер Модифікований кластер. Змініть розмір цієї функції на два поля введення даних. Якщо імена в полях виведення даних відрізняються від показаних на ілюстрації, слід натиснути правою кнопкою миші по імені елемента і в контекстному меню увійти в розділ SelectItem.

4. Збережіть ВП під ім'ям Робота з кластерами, VI.

5. Перейдіть на лицьову панель і запустіть ВП.

6. Поміняйте значення елементів в кластері Кластер і запустіть 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 лицьовій панелі?
6. Як збираються і демонтуються кластери?
7. Які функції обробки кластерів Ви знаєте?
8. Що таке поліморфізм в кластерах?
9. Як створити модифікований кластер?

Література

1. Дж. Трэвис, Дж. Кринг. LabVIEW для всех. Издательство ДМК, 2005 г.
2. Федосов В. П., Нестеренко А. К. Цифровая обработка сигналов в LabVIEW. Издательство: ДМК Пресс. 2007, 256 с.
3. Кехтарнаваз Н., Ким Н.Цифровая обработка сигналов на системном уровне с использованием LabVIEW. Издательство: Додэка-XXI. 2007, 288 с.