

Міністерство освіти і науки України
Національний університет водного господарства
та природокористування
Навчально-науковий інститут автоматики,
кібернетики та обчислювальної техніки

Кафедра обчислювальної техніки

04-04-261М

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з навчальної дисципліни
«Інформаційні системи реального часу» (частина 2)

для здобувачів вищої освіти першого
(бакалаврського) рівня за освітньо-
професійною програмою «Комп'ютерна
інженерія» спеціальності 123
«Комп'ютерна інженерія» денної та заочної
форм навчання

Рекомендовано науково-
методичною радою з
якості ННІАКОТ
Протокол № 1 від 09.10.2023 р.

Рівне – 2023

Методичні вказівки до виконання лабораторних робіт з навчальної дисципліни «Інформаційні системи реального часу» (частина 2) для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Комп'ютерна інженерія» спеціальності 123 «Комп'ютерна інженерія» денної та заочної форм навчання [Електронне видання] / Шатна А. В., Шатний С. В., Бойчура М. В. – Рівне : НУВГП, 2023 – 48 с.

Укладачі: Шатна А. В., старший викладач кафедри обчислювальної техніки; Шатний С. В., к.т.н., доцент кафедри обчислювальної техніки; Бойчура М. В., к.т.н., старший викладач кафедри обчислювальної техніки.

Відповідальний за випуск: Круліковський Б. Б., к.т.н., доцент., зав. кафедри обчислювальної техніки.

Керівник (гарант) освітньої програми
«Комп'ютерна інженерія»
спеціальності 123 «Комп'ютерна інженерія» Сидор А. І.

© А. В. Шатна,
С. В. Шатний,
М. В. Бойчура, 2023
© НУВГП, 2023

ЗМІСТ

Вступ	4
Лабораторна робота №6. Робота з масивами даних та передача значень в текстовий файл в системи LabView в режимі реального часу.....	5
Лабораторна робота №7. Основні прийоми обробки зображень в LabView.....	12
Лабораторна робота №8. Робота з тривимірними зображеннями та визначення координат в LabVIEW.....	20
Лабораторна робота №9. Аналіз табличних даних для інформаційних систем в середовищі LabView.....	27
Лабораторна робота №10. Робота зі звуковими потоковими даними в режимі реального часу в середовищі LabView.....	34
Список використаної літератури.....	41
Додаток 1.....	42
Додаток 2.....	47

Вступ

Дисципліна «Інформаційні системи реального часу» є складовою частиною підготовки студентів за спеціальністю 123 «Комп'ютерна інженерія». Метою викладання дисципліни «Інформаційні системи реального часу» є підготовка спеціалістів за освітньо-кваліфікаційним рівнем бакалавр відповідно до державних стандартів, встановлених освітньо-кваліфікаційною характеристикою (ОКХ) та освітньо-професійною програмою (ОПП) підготовки магістрів вищезазначеного фахового спрямування. Мета навчальної дисципліни – формування системи теоретичних і практичних знань студентами базових складових щодо побудови і функціонування систем реального часу та розробки програмного забезпечення у режимі реального часу.

Згідно з вимогами освітньо-професійної програми студенти після засвоєння дисципліни «Інформаційні системи реального часу» мають продемонструвати такі результати навчання як методи оцінки продуктивності обчислювальних систем, визначення та основні вимоги, що пред'являються до систем реального часу, особливостей планування задач та типів планувальників, що використовують у системах реального часу, проблем, що виникають при обміні інформацією між процесами і задачами, проблеми паралельної обробки та основних шляхів їх вирішення, методів оцінки часу виконання задач як у автономному режимі виконання, так і при роботі в конкретній програмній системі на різних стадіях проектування програмно-технічного комплексу реального часу.

Лабораторна робота №6

Робота з масивами даних та передача значень в текстовий файл в системі LabView в режимі реального часу

Мета: навчитись створювати масиви та передавати в них дані, навчитись та освоїти методику підключення текстових документів та передавання в них числових та символічних даних в режимі реального часу.

Короткі теоретичні відомості

Массив LabVIEW - це набір індексованих даних одного типу. Він може мати будь-який розмір і містити до $2^{31}-1$ елементів розміру. Елементом масиву може бути будь-який тип даних, для складу масиву, таблиці чи графіки. Доступ до елементів здійснюється за допомогою індексів. Значення індексів лежать в діапазоні від 0 до N-1, де N - кількість елементів масиву.

Для створення масиву елементів управління або індикації даних необхідно вибрати шаблон масиву з палітри Controls => Array & Cluster і помістити його на особову панель.

Масив містить дві області: зона індексу та зона видимості елементів. Одразу після створення масиву видно тільки один елемент. Для того, щоб побачити кілька елементів масиву, необхідно за допомогою інструменту переміщення розтягнути зону видимості елементів у горизонтальному або вертикальному напрямку.

Двовимірний (2D) масив являє собою прямокутну таблицю (матрицю). Кожен елемент двовимірного масиву характеризується двома індексами. Для збільшення розмірності масиву необхідно клацнути правою кнопкою миші по елементу індексу і вибрати в контекстному меню пункт Додати розмір.

Порядок виконання роботи:

1. Створити новий проєкт VI
2. Створити структуру While Loop
3. До циклу додати наступні елементи:
 - 3.1. Меню File I/O – Open / Create / replace File;
 - 3.2. Меню File I/O – Write to Text File;
 - 3.3. Меню File I/O – close File

Результат розташування елементів зображений на рис.6.1.



Рис.6.1. Розташування елементів на фронтальній панелі

4. Для роботи з рядками додати наступні елементи:
 - 4.1. Меню String – Concatenate String (встановити кількість входів 3);
 - 4.2. Меню String – End of line Constant;

4.3. Меню String – Number String Conversion – Number To Fractional String;

4.4. Меню String – string Constant;

4.5. Меню String – End of Line Constant

Результат розташування елементів зображено на рис.6.2.

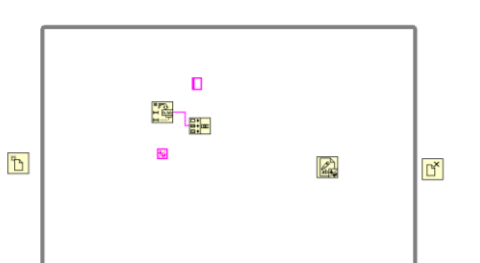


Рис.6.2. Розташування рядкових елементів в основному циклі

5. Додати підбір випадкових чисел, для цього виконати наступні дії в меню Numeric – Random Number (0,1). Підписати константу та підключити все як показано на рис.6.3.

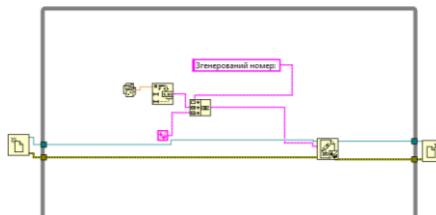


Рис.6.3. Підключення основних елементів до циклу

6. Для елемента «Close File» створити індикатор на виході «error out»

7. Для елемента «Open» для входу «file path» створити контроль, для входу «operation» створити константу а для входу «access» також константу та змінити її на «write-only». Приклад створення зображений на рис.6.4.

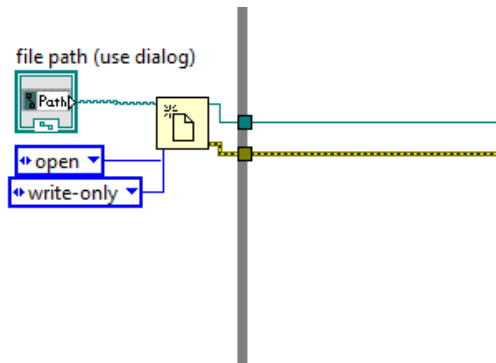


Рис.6.4. Опрацювання елемента «Open»

8. Додати для циклу аварійну зупинку.

9. Створити на своєму ПК текстовий файл з назвою test.txt

10. Додати в цикл часову затримку із меню Timing – Wait Until Next ms Multiple, та вхід затримки додати константу 100

11. На блок діаграмі відкрити створений файл test.txt. запустити програму та почекати декілька секунд, зупинити кнопкою аварійної зупинки циклу та перевірити чи відбувся запис до текстового фалу, приклад файлу з записом зображений на рис.6.5.

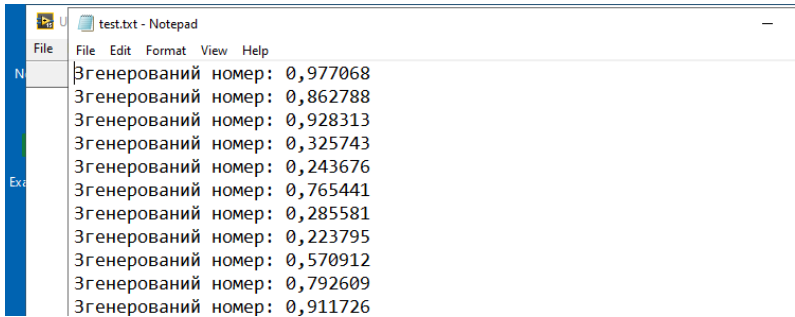


Рис.6.5. Текстовий файл з даними

12. Для перевірки себе, нижче зображені кінцеві результати панелей блок діаграми та фронтальної. Рис.6.6.

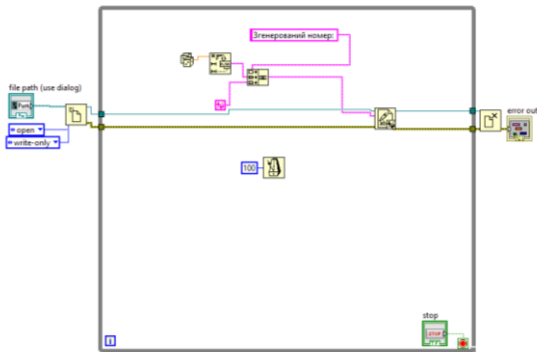
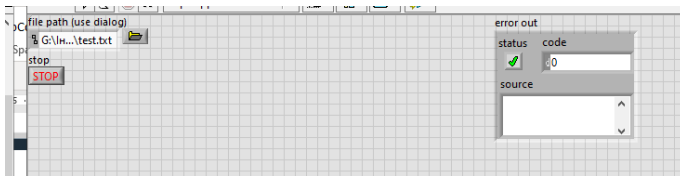


Рис.6.6. Результат побудови панелей системи

13. Для того, щоб заповнити таблицю в системі LabVIEW необхідно додати масив та номерний індикатор (індикатор розтягти донизу) рис.6.7.

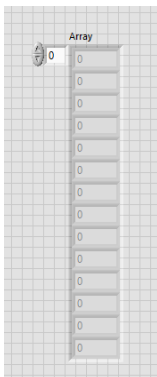


Рис.6.7. Створення полей для отримання значень

14. Додати цикл for, до нього додати наступні елементи:

14.1. Із меню Numeric – random Number

14.2. Array – Build Array (2 input)

14.3. Для «N» створити контроль

14.4. ПКЛ на циклі for – Add Shift Register. Приклад вище сказаного зображений на рис.6.8.

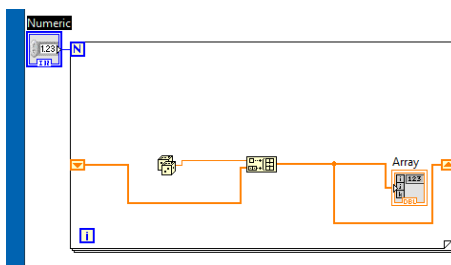


Рис.6.8. Налаштування циклу For

15. Додати головну затримку Wait (ms) з константою 500, на кнопку вхід поставити значення 10. Запустити програму. Результат на рис.6.9.

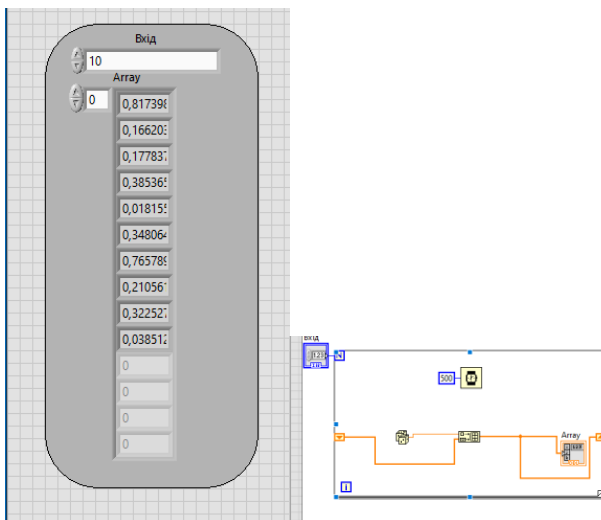


Рис.6.9. Результат налаштування блок-діаграми та фронтальної панелі

Варіанти до виконання лабораторної роботи:

Додати дизайн, отримати значення та отримавши результат програми зробити знімок вікна фронтального та блок-діаграми, оформити звіт.

Вміст звіту

1. Тема, мета роботи.
2. Хід роботи.
3. Результати розв'язання наведених задач в порядку виконання згідно варіанта (скрінки блок-діаграм та фронтальної панелі).
4. Висновки про результат виконаної роботи.

Контрольні питання:

1. Які у LabVIEW є елементи графічного відображення інформації?
2. Що являє собою розгорнення осцилограми?
3. Які є режими відображення розгорнення осцилограми?
4. Що таке цикл?
5. Як працює цикл із фіксованим числом ітерацій For Loop?

Лабораторна робота №7

Основні прийоми обробки зображень в LabView

Мета: Навчитись працювати з зображеннями через 1-вимірні та 2-вимірні масиви. Навчитись обробляти зображення, змінювати та зберігати їх в режимі реального часу.

Короткі теоретичні відомості

Для графічного відображення даних у LabVIEW використовують спеціальні засоби візуалізації:

- Графік Діаграм (Waveform Chart) – це спеціальний цифровий елемент

відображення, призначений для відображення однієї і більше діаграм.

- Графік Діаграм має три режими відображення даних:

- strip chart являє собою екран, що прокручується зліва направо

подібно паперовій стрічці.

- score chart , після досягнення правої межі вікно діаграми

очищується, і заповнення діаграми починається з лівої границі.

- sweep chart, на відміну від режиму scope chart, вікно діаграми не

очищується, а нові дані відокремлюються від старих вертикальної лінією – маркером.

- Графік осцилограм (Waveform Graph) і Двохкоординатний графік Осцилограм (XY graph) відображають дані з масивів.

- Для редагування графіку або зміни налаштувань графіку Діаграм необхідно клацнути по ньому правою кнопкою миші.

- Для візуалізації тривимірних даних можна скористатися графіками і таблицями інтенсивності. Третя координата відображається кольором відповідно до певної колірної схеми. Графіки і таблиці інтенсивності часто застосовуються в задачах спектрального аналізу, візуалізації температури і обробки зображень.

- При виведенні даних на діаграму або графік зручно використовувати вікно контекстної довідки Context Help .

Порядок виконання:

1. Створити новий VI
2. Завантажити на свій ПК будь-яке чорно-біле зображення формату PNG, додати наступну панель FP-String & Path – File Path Control, назвати дану панель «Вхідне зображення» додати скачане зображення.

3. BD – Graphics & Sound – Graphics Formats – Read PNG File.vi. Вхід даного блоку “Path to PNG file” підключити до виходу блоку з назвою «Вхідне зображення»

4. В блоці «Read PNG File.vi» на виході «image data» створити індикатор. Підключити до лінії перетину або згідно рисунка 7.1. Запустити та отримати в сусідній панелі інформацію про додане зображення.

5. BD - Graphics & Sound – Picture Functions – Draw Flattened Pixmap.vi, створити на виході даного блоку індикатор. В фронтальній панелі має з’явитись дане зображення рис.7.1.

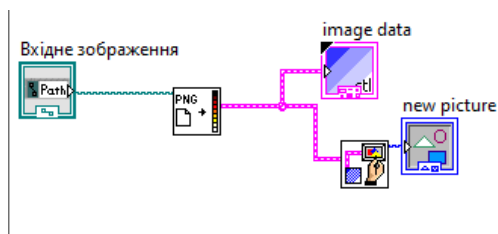


Рис.7.1. Підключення зображення

6. Виконаємо деякі примітивні процеси з даним зображенням (перевернути, змінити кольори і тд.), BD – Cluster – Unbundle By Name, підключити аналогічно до п.4.(лінії перетину), та змінити налаштування доданого блок ЛКЛ “image”.

7. Далі необхідно додати 1 вимірний масив. BD – Arrays – Reverse 1D Array, підключити його як показано на рис.7.2.

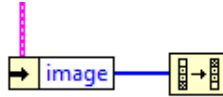


Рис.7.2. Додавання масиву

8. Тепер необхідно виконати зворотній процес, для цього BD – Cluster – Bundle By Name, та змінити налаштування доданого блок ЛКЛ “image” рис.7.3.

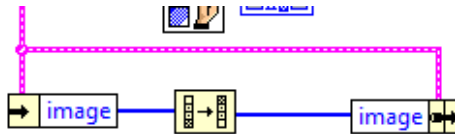


Рис.7.3. Зворотній процес

9. До щойно створеного блоку п.8. додати блок Draw Flattened та індикатор (можна скопіювати в схемі вище), підключити як показано на рис.7.4. З'явиться ще одне зображення в сусідній панелі. Запустити, при вірному виконанні зображення повинне перевернутись.

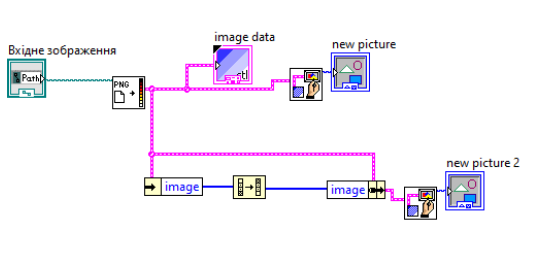


Рис.7.4. Налаштування додаткового зображення

10. Для того, щоб зберегти даний процес до файлу виконати п.2, та назвати нову панель «Вихідне зображення» рис.7.5.

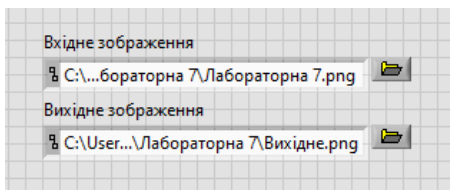


Рис.7.5. Підключення зображень

11. BD – Graphics – Graphics Formats – Write PNG File.vi, на його вхід “path to png file” підключити блок «Вихідне зображення» та виконати підключення як показано на рис.7.6. Запустити та переглянути чи зберіглося вихідне зображення до папки на вашому пристрої

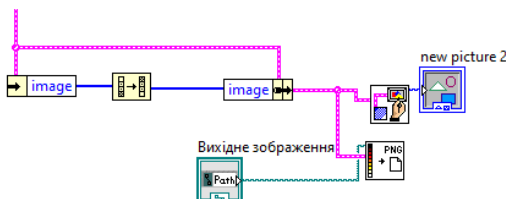


Рис.7.6. Підключення блоку

12. Ускладнимо процес, спочатку необхідно додати BD – Graphics – Graphics formats – Unflatten Pixmap.vi. Його вхід підключити до перетину png та Draw, як вже робили вище.

13. Для виконання більш складних процесів необхідно використовувати багатовимірні масиви. Вони працюють

через блок Convolution. Додаймо даний блок BD – Signal Processing – Sig Operation – Convolution. Для того щоб підключити до нього 2Вимірний масив необхідно також додати блок TdF, BD – Numeric – Conversion – To Double Decision Float, вхід підключити до вих. Попереднього блоку 8-bit.

14. Налаштування блоку Convolution:

14.1. Вхід X підключити до блоку BDL

14.2. Для входу Y створити константу та налаштувати її як показано на рис.7.7.

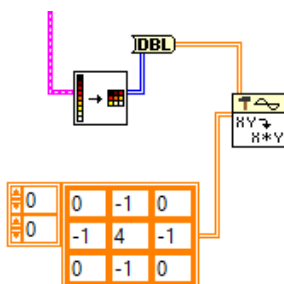


Рис.7.7. Створення масиву

15. Тепер додамо наступні елементи:

15.1. Numeric – Conversions – To Unsigned By Integer (U8)

15.2. Numeric – Add – на Y додати константу 128, а X підключити до виходу X*Y, сам вихід підключити до U8

16. Додамо відображення для результату:

BD – Graphics – Picture Function – Draw Unflattened Pixmap.vi, вхід даного блоку “data” підключити до виходу U8

17. Щоб підключити масив зображень необхідно створити індикатор рисунку для блоку Draw (п.16), для цього: На виході “New picture” створити індикатор або скопіювати вже готовий та налаштований індикатор вище, запустити та зафіксувати зміни.

18. Зробити останні налаштування блоків Зробити підключення блоків між собою, рис.7.8, а саме вхід “Colors” підключити до виходу “Color table”, в останньому блоці Draw натиснувши ПКЛ змінити тип на 8біт, Select Type – Draw 8-bit Pixmap. Запустити та отримати результат зміненого зображення. Оформити звіт. Кінцева схема представлена в Додатку 1 Приклад Б.

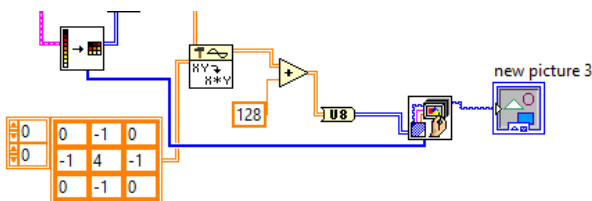


Рис.7.8. Підключення входів

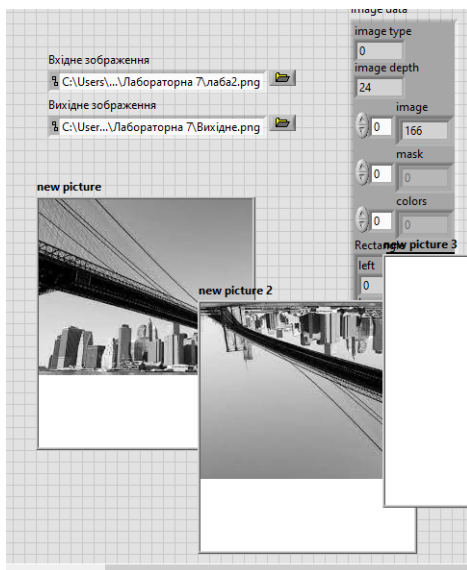


Рис.7.9. Кінцевий результат фронтальної панелі

Варіанти до виконання лабораторної роботи:

Додати дизайн, отримати значення та отримавши результат програми зробити знімок вікна фронтального та блок-діаграми, оформити звіт.

Вміст звіту

1. Тема, мета роботи.
2. Хід роботи.
3. Результати розв'язання наведених задач в порядку виконання згідно варіанта (скріни блок-діаграм та фронтальної панелі).
4. Висновки про результат виконаної роботи.

Контрольні питання:

1. Які засоби візуалізації використовуються в для графічного відображення даних?

2. Які режими відображення даних використовуються в графіку Діаграм Waveform Chart?

3. Яка функція використовується для об'єднання даних, що відображаються при виведенні кілька графіків на графік Діаграм Waveform Chart?

4. У чому відмінність одиночного графіку осцилограм і графіку множини осцилограм?

5. В чому відмінність графіку осцилограм (Waveform Graph) і двохкоординатного графіку осцилограм (XY Graph)?

Лабораторна робота №8

Робота з тривимірними зображеннями та визначення координат в LabVIEW

Мета: навчитись будувати 3Д зображення в програмному середовищі LabVIEW, навчитись визначати координати деталей в режимі реального часу.

Короткі теоретичні відомості

Створення тестових послідовностей

Після створення тестів у LabVIEW інженери використовують TestStand для створення тестової послідовності. Коли складність продукту та кількість пристроїв зростає, TestStand усуває власну розробку секвенсора з функціями для інтеграції кількох вимірювань, виконання послідовності та паралельного тестування.

TestStand для автоматизованого тестування:

Зменшити витрати на розробку



Не будуйте його самостійно. TestStand має необхідні можливості секвенування, створення звітів і розгортання, заощаджуючи значний час і ресурси на розробку.



Отримайте автоматичну статистику

Користувачі мають миттєвий доступ до своїх даних за допомогою TestStand. Він має готові звіти для відстеження результатів на кожному кроці, перегляду статусу тесту та зберігання в базах даних.



Збільшення пропускнуої здатності

Продуктивність стрімко зростає з TestStand. Він включає в себе моделі паралельного виконання, які скорочують загальний час тестування.



Скоротіть час простою

Користувачі швидше виявляють і вирішують проблеми в TestStand. Він має інтуїтивно зрозумілі функції налагодження, тому оператори можуть поетапно виконувати виконання, щоб швидко знаходити помилки.

Порядок виконання роботи:

1. Створити новий VI
2. FP – Graph – 3D Picture

3. BD – Structures – While Loop (додати кнопку стоп)
4. BD – timing – Wait (ms) встановити на вхід константу 1000
5. BD – Graphics – 3D Picture – Geometries– Create Box та Create 3D Axis
6. BD – graphics – 3D Picture – Object – Add Object та Create Object
7. В блок Create Object встановити наступні налаштування: ПКЛ – Visible items – Polymorphic VI Selector – Object drawable
8. Виконати підключення на рис.8.1. Встановити константу «Create 3D Axis» входу “Name”, входу “Length” константу 2, “Length” константу 1 підключити до Radius. Встановити константу Create Object входу “Name”, ввести “box”
- 8.1. Character Size – константа 0,5
- 8.2. Half plane? – контроль
- 8.3. Axis label? – контроль. Запустити на екрані FP “3D Picture” червоний

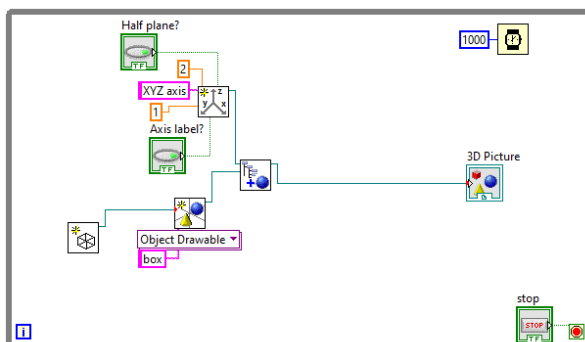


Рис.8.1. Схема підключення

9. ПКЛ по “3D Picture” – Invoke Node – Auto focus, запустити та отримати зображення рис.8.2.

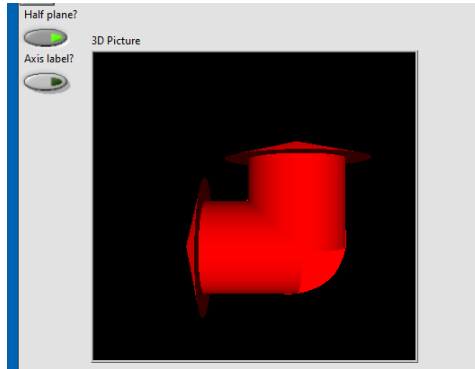


Рис.8.2. Отримане зображення

10. FP – ПКЛ по рисунку – Camera Controller – Spherical. Запустити та перевірити чи виконується 3Д

11. В BD константу входу Radius змінити з «1» на «0,3»

12. Запустити. Натиснути на кнопку Axis label?
Зображення 8.3. Перевірити чи присутні всі 3 вісі X, Y, Z.

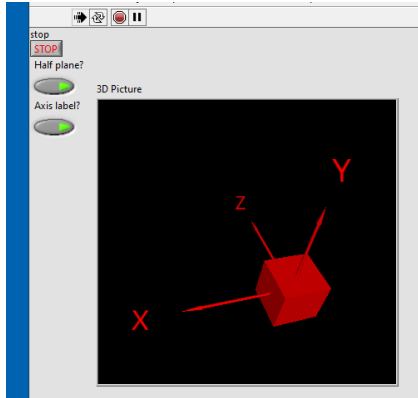


Рис.8.3. Перевернуте зображення

13. Виключити перемикач Half plane?, перевірити що відбулося та зафіксувати.
14. BD – видалити блок “Auto Focus” замість нього додати блок : ПКЛ 3D Picture – Create – Invoke Node – Setup Camera. Налаштувати даний блок:
 - 14.1. Camera position – контроль
 - 14.2. Target – контроль
 - 14.3. Up Direction – контроль. Запустити та перевірити рис.8.4.

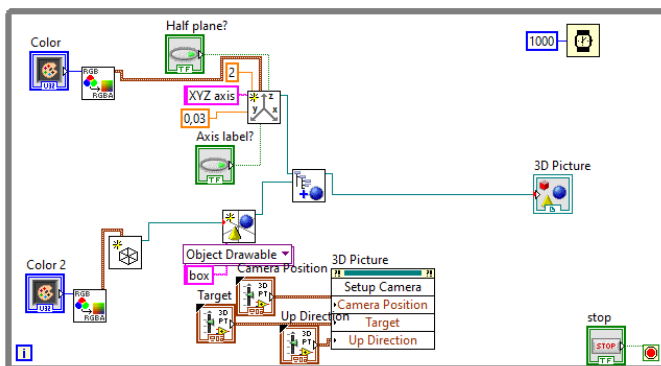


Рис.8.5. Підключення блоків

16. Кінцеві схеми зображенні на наступних рисунках
 Додаток 1 Приклад С1 та Додаток 2 Приклад С2 .

Варіанти до виконання лабораторної роботи:

Згідно варіанту оформити 3Д зображення в відповідних кольорах. Оформити звіт, зробити скрінш та висновки.

Варіант	01	02	03	04	05
Сигнал	Червний Жовтий	Зелени й, Синій	Зелений, Синій	Жовтий, Зелений	Жовтий, Червони й
Варіант	06	07	08	09	10
Сигнал	Червний Жовтий	Зелени й, Синій	Зелений, Синій	Жовтий, Зелений	Жовтий, Червони й

Вміст звіту

1. Тема, мета роботи.
2. Хід роботи.

3. Результати розв'язання наведених задач в порядку виконання згідно варіанта (скрині блок-діаграм та фронтальної панелі).
4. Висновки про результат виконаної роботи.

Контрольні питання:

1. Які засоби візуалізації використовуються в для графічного відображення даних?
2. Які режими відображення даних використовуються в графіку Діаграм Waveform Chart?
3. Яка функція використовується для об'єднання даних, що відображаються при виведенні кілька графіків на графік Діаграм Waveform Chart?
4. У чому відмінність одиночного графіку осцилограм і графіку множини осцилограм?
5. В чому відмінність графіку осцилограм (Waveform Graph) і двохкоординатного графіку осцилограм (XY Graph)?

Лабораторна робота №9

Аналіз табличних даних для інформаційних систем в середовищі LabView

Мета: навчитись створювати таблиці в системі LabView, навчитись та освоїти методику передачі та зчитування даних в режимі реального часу. Проаналізувати отримані результати.

Короткі теоретичні відомості

Основні функції для розробки FPGA

Графічне програмування. Пропустіть VHDL і Verilog. LabVIEW забезпечує інтуїтивно зрозумілий спосіб проектування систем і краще візуально представляє потік даних і паралельні процеси, які відбуваються в FPGA.

Апаратна інтеграція. LabVIEW FPGA створено для обладнання NI. Традиційно складні завдання, як-от конфігурація вводу-виводу, передача даних і буферів, легко виконуються з LabVIEW FPGA.

Система компіляції FPGA. Інженери швидше впроваджують проекти на апаратному забезпеченні за допомогою LabVIEW FPGA. Програмне забезпечення спрощує процес компіляції від перекладу до оптимізації та синтезу маршрутизації.

LabVIEW для програмування FPGA



Процес у реальному часі

Апаратне забезпечення NI FPGA адаптовано до потреб систем, які виробляють великі обсяги даних, наприклад радіочастотних систем. За допомогою справжнього апаратного паралелізму можна обробити десятки вимірювань.



Інтеграція наявного коду HDL

Інженери економлять час розробки, повторно використовуючи наявний код. LabVIEW FPGA може інтегрувати HDL або список мереж IP, включаючи файли синтезу VHDL і Verilog.



Впровадити користувальницьке спілкування

Коли комунікаційні потреби перевищують можливості стандартних протоколів, таких як SPI або I2C, LabVIEW FPGA дозволяє реалізувати спеціальні протоколи для зв'язку з тестовими системами та пристроями.



Налаштувати відповідно до ваших потреб

LabVIEW FPGA забезпечує розширений контроль апаратного забезпечення. Він має функціональні можливості для реалізації спеціального часу, запуску та синхронізації на пристроях NI FPGA.

Порядок виконання роботи:

1. Створити новий VI
2. Створити таблицю, FP – List, Table – Table.

Налаштувати її натиснувши на ній ПКЛ – Visible Items – Columb Headers.

3. Заповнити таблицю як показано на рис.9.1.

Time	Data	P1	P2	P3				

Рис.9.1. Створення таблиці

4. BD – створити для Table - local variable

5. Додати структуру While
6. Додати масив “Insert Into Array”
7. Натиснути ПКЛІ на лінію While та вибрати “Add shaft register”
8. Додати рядкову константу із меню String та масив констант з меню Array, з’єднати їх перетягнувши рядкову константу мишкою в масив констант
9. Виконати підключення рис.9.2

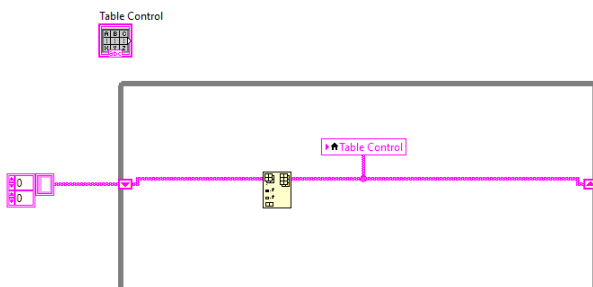


Рис.9.2. Схема підключення

10. BD – додати “Random Number 0-1” з меню Numeric, додати Multiply
 - 10.1. На вході «У» створити константу 100
 - 10.2. Вхід «X» підключити до виходу Random
11. Додати Format Into String з меню String. Для входу “Format string” створити константу “%2f”
12. Додати “Build Array” з меню Array. Зробити копію елементів п.10-11 та виконати підключення рис.9.3.

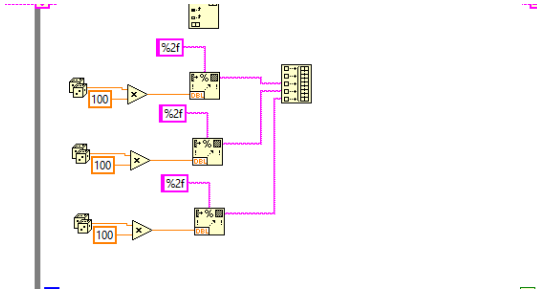


Рис.9.3. Додавання масиву

13. Додати «Get Date/Time In Seconds» та «Get Date/Time String», виконати підключення рис.9.4.

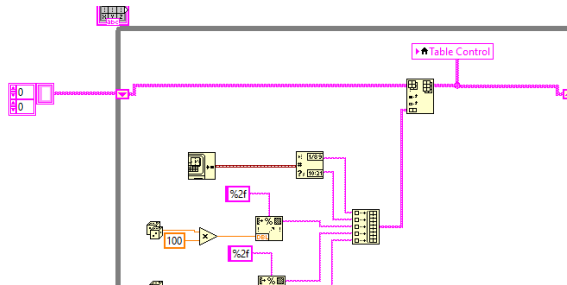


Рис.9.4. Додавання «Get Date/Time In Seconds»

14. Додати «Wait (ms)» з меню Timing, створити для нього константу 500

15. Перейти на фронтальну панель та запустити програму. Зафіксувати дані, що надійшли в таблицю для звіту.

16. BD – додати Array Size з меню array та Index Array

16.1. Для «Index Array» для входу Index створити константу та додати блок віднімання “Subtract” для даного

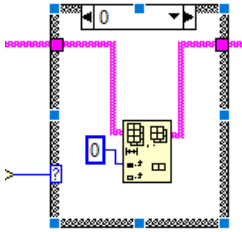


Рис.9.8. Створення константи

19. Перейти в фронтальну панель, запустити програму та зафіксувати зміни для звіту. Кінцевий результат FP та VD зображено в Додатку 1 Приклад Д1 та Додаток 2 Приклад Д2.

Варіанти до виконання лабораторної роботи:

Додати дизайн, отримати значення та отримавши результат програми зробити знімок вікна фронтального та блок-діаграми, оформити звіт.

Вміст звіту

1. Тема, мета роботи.
2. Хід роботи.
3. Результати розв'язання наведених задач в порядку виконання згідно варіанта (скріни блок-діаграм та фронтальної панелі).
4. Висновки про результат виконаної роботи.

Контрольні питання:

1. Які команди використовуються в середовищі LabVIEW для

налагодження програм?

2. Чи має якесь значення порядок підключення провідників до

елементів підсумовування і віднімання, множення і ділення?

3. Яка форма графічного представлення результатів роботи програми в найбільшій мірі відображає дискретний принцип роботи ПК?

У яких випадках доцільніше використовувати інші графіки?

Лабораторна робота №10

Робота зі звуковими потоковими даними в режимі реального часу в середовищі LabVIEW

Мета: навчитись читати та писати аудіо файли, імпортувати аудіоінформацію як масив . Створити елемент керування шляхом до файлу та прочитати звукову точкову хвилю в режимі реального часу.

Короткі теоретичні відомості

Основні функції для підключення інструментів

7000+ інструментальних драйверів. Отримайте неперевершене підключення інструментів із LabVIEW. Будь-який пристрій доступний і програмується за допомогою драйверів, які абстрагують численні команди низького рівня у функції високого рівня.

Приклади Open-and-Run. Ніколи не починайте з нуля. LabVIEW містить приклади, які допомагають користувачам зрозуміти основи та повторно використовувати код для швидшого завершення розробки.

Графічне програмування. Програмування в LabVIEW здійснюється за допомогою перетягування. LabVIEW містить усі функції для швидкого створення тестової системи, включаючи інтерфейс користувача та аналіз даних.

LabVIEW для керування приладами



Тест збірки швидше

LabVIEW скорочує час до першого вимірювання. Доступні драйвери та узгоджена структура програмування LabVIEW спрощують розробку.



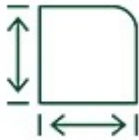
Налаштувати відповідно до ваших потреб

LabVIEW надає доступ до розширених можливостей інструменту. Включено команди для операцій низького рівня для таких функцій, як синхронізація між пристроями.



Підключіться до всіх своїх інструментів

LabVIEW — єдиний інструмент для всіх ваших інструментів. Інженери можуть додавати численні пристрої та спілкуватися між ними, усуваючи необхідність використання кількох програм.



Робіть більше з вашими вимірюваннями

LabVIEW створено для інженерів з випробувань і вимірювань. Він містить функції візуалізації даних, обробки сигналів, підключення до баз даних і

створення звітів.

Порядок виконання роботи:

1. Створити новий VI
2. Завантажити на свій ПК будь-яке аудіо формату .wav
3. BD – Graphocs – Sound – Files – Simple Read
4. FP – String & Path – File path Control. Додати до нього завантажене аудіо.
5. BD – додати Index Array. Виконати підключення рис.10.1

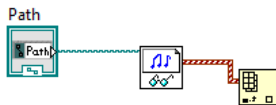


Рис.10.1. Підключення Index Array

6. FP – Graph – Waveform Graph, підключити його до схеми, запустити та зафіксувати зміни аудіо графіка.

7. BD – Sound – Output – Play Waveform – та натиснути Test Device, потім натиснути ОК. Запустити та перевірити чи почалось відтворення аудіо звуку. Схема підключення компонентів рис.10.2.

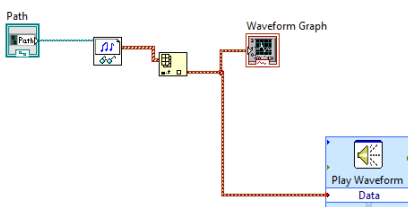


Рис.10.2. Схема підключення

8. Додати 1Вимірний масив Reverse 1D Array, підключити його через блок Get Wfm із меню waveform. Змінити налаштування Get Wfm на dt натиснувши на ньому ЛКМ. На його вході dt створити індикатор. Запустити та зафіксувати зміни в FP.

9. BD – Numeric – reciprocal. На виході доданого блоку створити індикатор, та виконати підключення рис.10.3. Запустити та зафіксувати зміни.

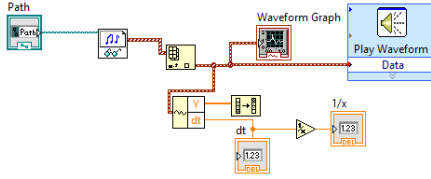


Рис.10.3. Схема підключення

10. Для того щоб отримати інформацію про наше аудіо виконаємо наступні дії: BD – Sound – Files – Info. На виході “Sound format” створити індикатор. Верхній вхід підключити до схеми на лінію між блоками Path та Sound File Read. Запустити та зафіксувати зміни.

11. Видалити блок reciprocal та індикатор, на його місце встановити блок Build Waveform, розтягти його вниз на +1 рядок. Змінити на dt, як це робили в п.8. Схема підключення зображена на рис.10.4

12. Назву блоку Waveform змінити на «Вхідне аудіо». Скопіювати даний блок, та підключити рис.10.4 та назвати скопійований блок «Вихідне аудіо». Запустити та зафіксувати зміни, якщо все виконано вірно то в графіках буде відображено обернене аудіо.

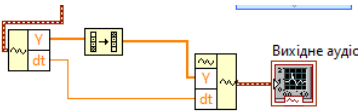


Рис.10.4. Підключення блоку

13. Скопіювати блок Play Waveform та підключити його до першого блоку, вхід data підключити до «вихідне аудіо»

14. BD – Sound – Files – Simple Write

15. Додати Build Array. Зробити копію path в FP та назвати її «Вихідний файл» у вікні назви задати іншу назву як це вже виконувалось на лабораторній роботі по обробці зображень. Щоб файл зберігся натиснути по створеному блоці ПКЛ – browse options та поставити галочку біля New or existing та перевірити чи збереглись перетворене аудіо. Кінцева схема зображена на рис.10.5 та Додаток 1 Приклад K1.

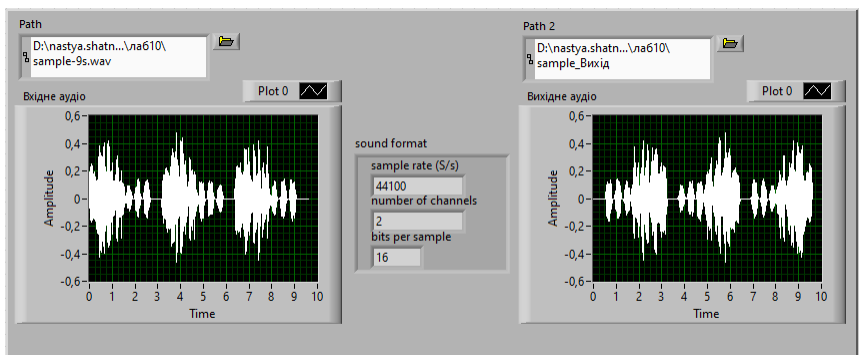


Рис.10.5. Графічна панель

Варіанти до виконання лабораторної роботи:

Додати дизайн, отримати значення та отримавши результат програми зробити знімок вікна фронтального та блок-діаграми, оформити звіт.

Вміст звіту

1. Тема, мета роботи.
2. Хід роботи.

3. Результати розв'язання наведених задач в порядку виконання згідно варіанта (скріни блок-діаграм та фронтальної панелі).

4. Висновки про результат виконаної роботи.

Контрольні питання:

1. Які варіанти конструкції «цикл» використовується в LabVIEW?

2. В яких випадках використовується цикл While?

3. В яких випадках використовується цикл For?

4. В яких випадках використовуються регістри зсуву?

Список використаної літератури

1. Harel D., Politi M. Modeling Reactive Systems with Statecharts. New York : McGraw Hill, 1998.

2. Jacobson I., Booch G., Rumbaugh J. The Unified Software Development Process. Reading. Mass. : Addison-Wesley, 1999.

3. Karampogias K. An In-Memory RDMA-Based Architecture for the Distributed Filesystem Computer Engineering and Networks Laboratory. Zurich : Computer Engineering and Networks Laboratory; ETII Zurich; IBM Research, 2012.

4. <https://www.ni.com/en-us/support/downloads/software/products/download.labview.html#487445>

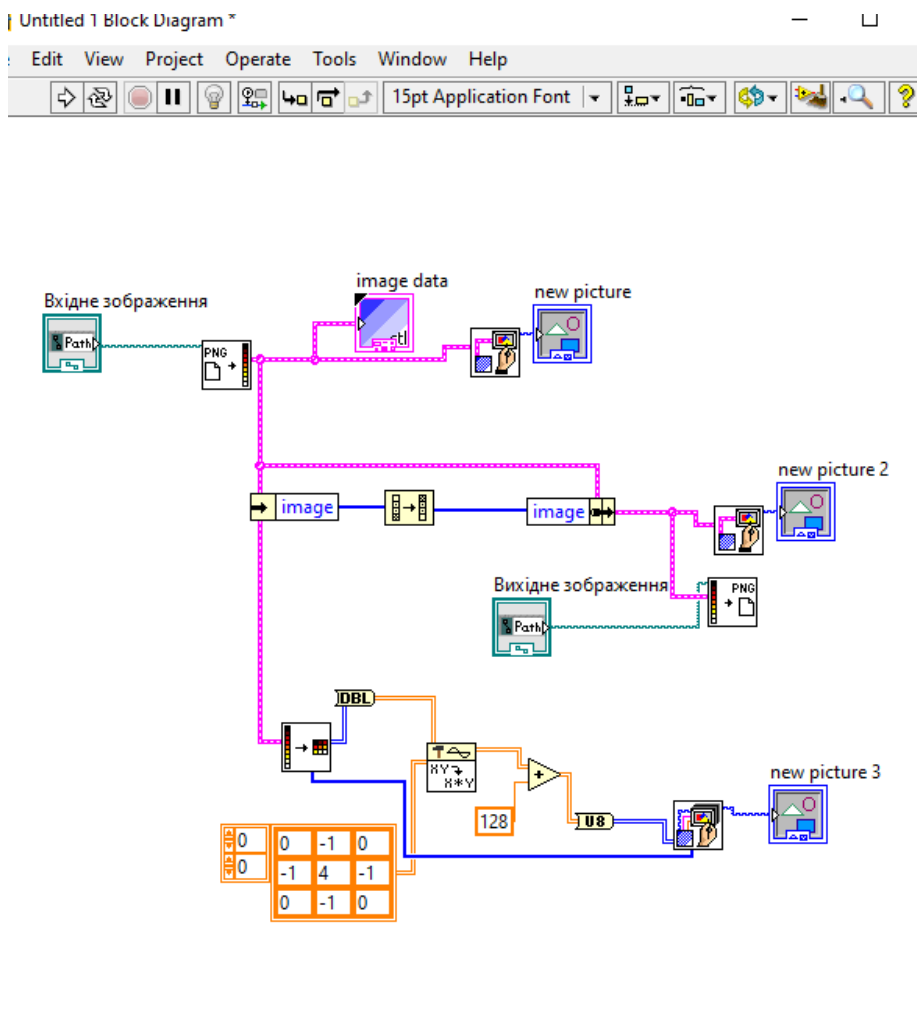
5. Hard Real-Time Computing Systems Predictable Scheduling Algorithms and Applications. Authors: Giorgio C Buttazzo. Copyright: 2011.

6. Real-Time Systems Design Principles for Distributed Embedded Applications Authors: Hermann Kopetz. Copyright: 2011.

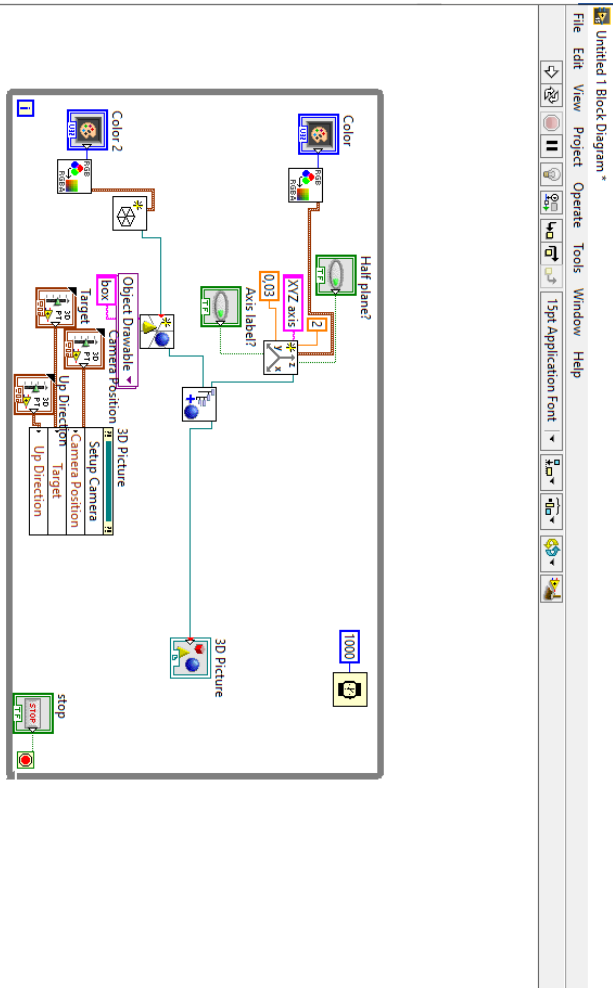
7. Hard Real-Time Computing Systems Predictable Scheduling Algorithms and Applications Authors: Giorgio C Buttazzo. Copyright: 2005.

Event-Triggered and Time-Triggered Control Paradigms. Authors: Roman Obermaisser. Copyright: 2005.

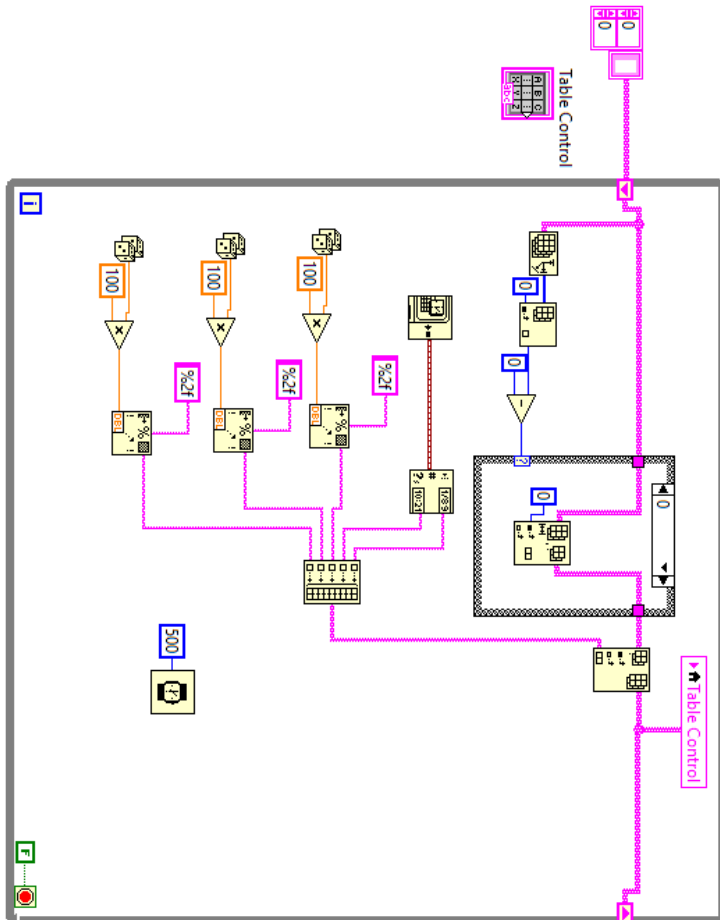
Приклад Б. Панель блок-діаграми обробки зображень в LabView.



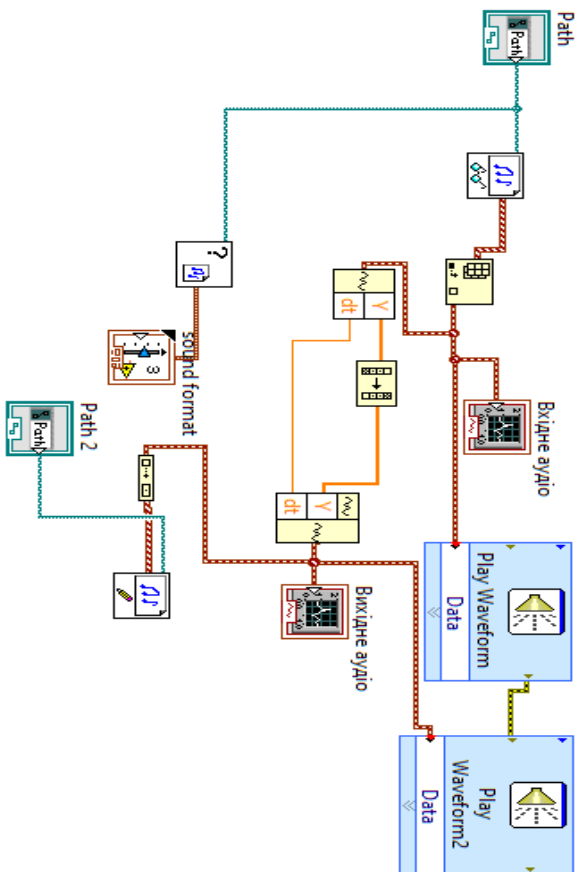
Приклад С1. Панель блок-діаграма «Робота з тривимірними зображенні та визначення координат в LabVIEW»



Приклад Д1. Панель блок-діаграма «Аналіз табличних даних для інформаційних систем в середовищі LabView»



Приклад К1. Панель блок-діаграма «Робота зі звуковими потоковими даними в режимі реального часу в середовищі LabView»



Додаток 2

Приклад С2. Фронтальна панель «Робота з тривимірними зображеннями та визначення координат в LabVIEW»

