



**Innovative Teaching Approaches
in development of Software Designed
Instrumentation and its application
in real-time systems**

Podstawy Projektowania Przyrządów Wirtualnych

Wykład 4: Wykresy, rejestr przesuwany, tworzenie subVI oraz zapis do pliku.

Co-funded by the
Erasmus+ Programme
of the European Union



Powtórka z wykładu 3

Co może być przyczyną przekreślonej strzałki *run*?

- a) subVI nie działa
- b) Operacja dzielenia przez zero
- c) Wymagane wejścia subVI nie są podłączone
- d) Kontrolka boolean jest podłączona do numeric indicator.

Powtórka z wykładu 3

Co może być przyczyną przekreślonej strzałki *run*?

- a) **subVI nie działa**
- b) Operacja dzielenia przez zero
- c) **Wymagane wejścia subVI nie są podłączone**
- d) **Kontrolka boolean jest podłączona do numeric indicator.**

Powtórka z wykładu 3

Które elementy z poniższych wchodzi w skład error cluster?

- a) Status: Boolean
- b) Error: String
- c) Code: 32-bit integer
- d) Source: String

Powtórka z wykładu 3

Które elementy z poniższych wchodzą w skład error cluster?

- a) **Status: Boolean**
- b) Error: String
- c) **Code: 32-bit integer**
- d) **Source: String**

Powtórka z wykładu 3

Wszystkie błędy mają ujemny kod błędu i wszystkie ostrzeżenia mają dodatni kod błędu.

- a) Prawda
- b) Fałsz

Powtórka z wykładu 3

Wszystkie błędy mają ujemny kod błędu i wszystkie ostrzeżenia mają dodatni kod błędu.

- a) Prawda
- b) **Fałsz**

Powtórka z wykładu 3

Funkcja Merge Errors łączy informacje o błędach z różnych źródeł.

- a) Prawda
- b) Fałsz

Powtórka z wykładu 3

Funkcja Merge Errors łączy informacje o błędach z różnych źródeł.

- a) Prawda
- b) **Fałsz**

Powtórka z wykładu 3

Która z poniższych struktur musi się wykonać chociaż raz?

- a) While Loop
- b) For Loop

Powtórka z wykładu 3

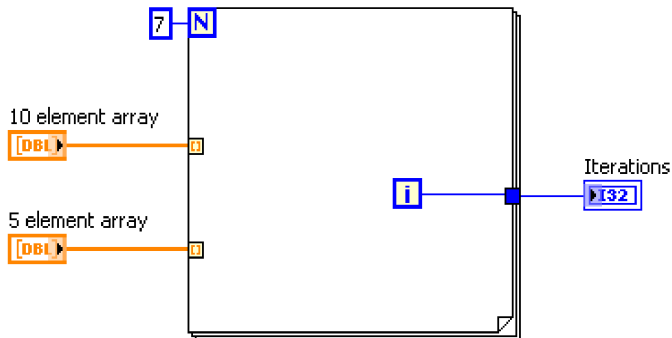
Która z poniższych struktur musi się wykonać chociaż raz?

- a) **While Loop**
- b) For Loop

Powtórka z wykładu 3

Jaką wartość zwróci *Iterations* po wykonaniu pętli?

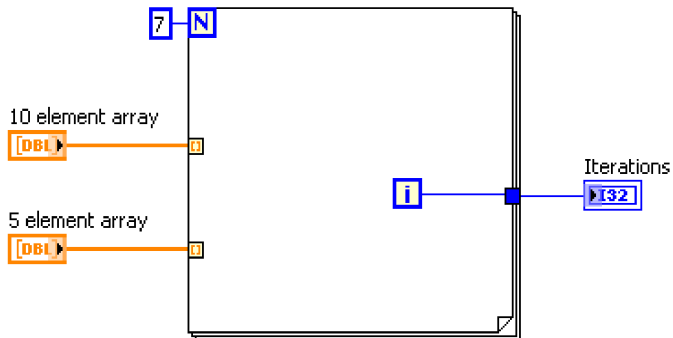
- a) 4
- b) 6
- c) 7
- d) 9



Powtórka z wykładu 3

Jaką wartość zwróci *Iterations* po wykonaniu pętli?

- a) 4
- b) 6
- c) 7
- d) 9



Powtórka z wykładu 3

Które zdania są prawdziwe?

- a) Domyślnym ustawieniem tunelowania dla pętli For jest autoindeksowanie.
- b) Domyślnym ustawieniem tunelowania dla pętli While nie jest autoindeksowanie.
- c) Zdanie A i B jest prawdziwe.
- d) Żadne z powyższych zdań nie jest prawdziwe.

Powtórka z wykładu 3

Które zdania są prawdziwe?

- a) Domyślnym ustawieniem tunelowania dla pętli For jest autoindeksowanie.
- b) Domyślnym ustawieniem tunelowania dla pętli While nie jest autoindeksowanie.
- c) **Zdanie A i B jest prawdziwe.**
- d) Żadne z powyższych zdań nie jest prawdziwe.

Powtórka z wykładu 3

Możesz używać wszystkich poniższych typów danych jako wejście do terminala *case selector* za wyjątkiem:

- a) Double
- b) Enum
- c) String
- d) Integer

Powtórka z wykładu 3

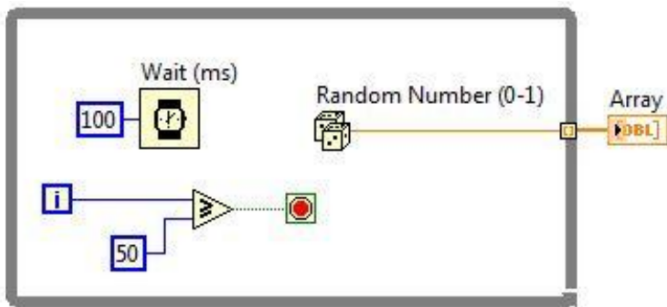
Możesz używać wszystkich poniższych typów danych jako wejście do terminala *case selector* za wyjątkiem:

- a) **Double**
- b) Enum
- c) String
- d) Integer

Powtórka z wykładu 3

Które zdanie jest prawdziwe?

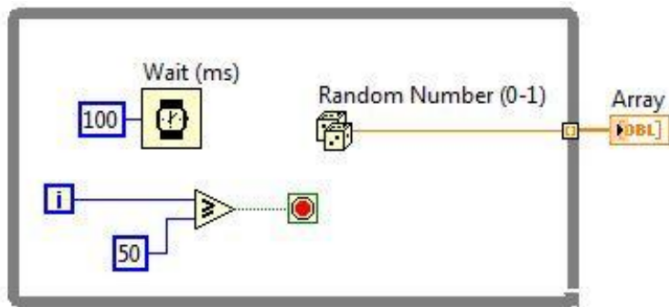
- a) Pętla się wykona 51 razy.
- b) Pętla się wykona 50 razy.
- c) Pętla się wykona 49 razy.
- d) Pętla się nie wykona.



Powtórka z wykładu 3

Które zdanie jest prawdziwe?

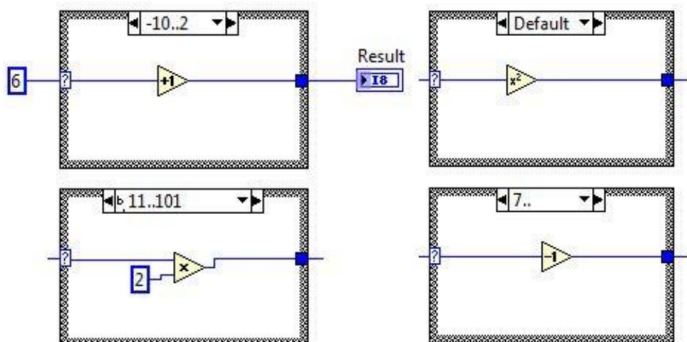
- a) Pętla się wykona 51 razy.
- b) Pętla się wykona 50 razy.
- c) Pętla się wykona 49 razy.
- d) Pętla się nie wykona.



Powtórka z wykładu 3

Poniżej przedstawiono wszystkie stany struktury Case. Jaka wartość zostanie wyświetlona w zmiennej *Result*?

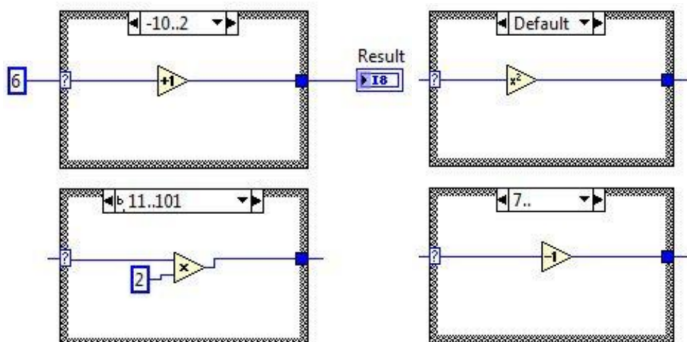
- a) 5
- b) 7
- c) 12
- d) 36



Powtórka z wykładu 3

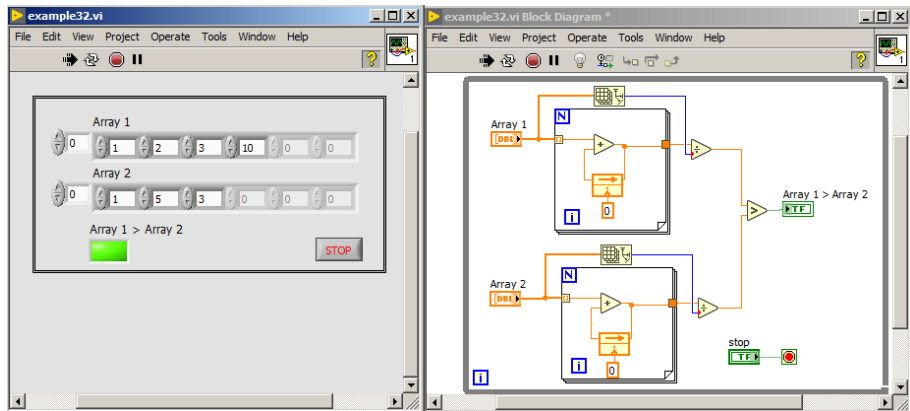
Poniżej przedstawiono wszystkie stany struktury Case. Jaka wartość zostanie wyświetlona w zmiennej *Result*?

- a) 5
- b) 7
- c) 12
- d) 36



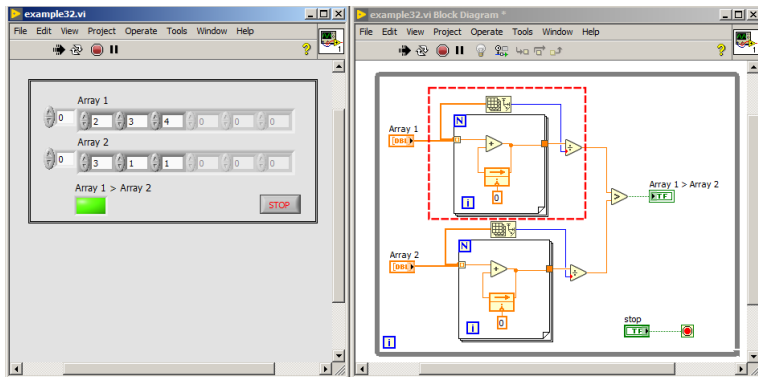
Po co są potrzebne subVI?

- Powtarzające się fragmenty kodu można zastąpić funkcją.
- W LabVIEW odpowiednikiem funkcji jest **subVI**.



Tworzenie subVI

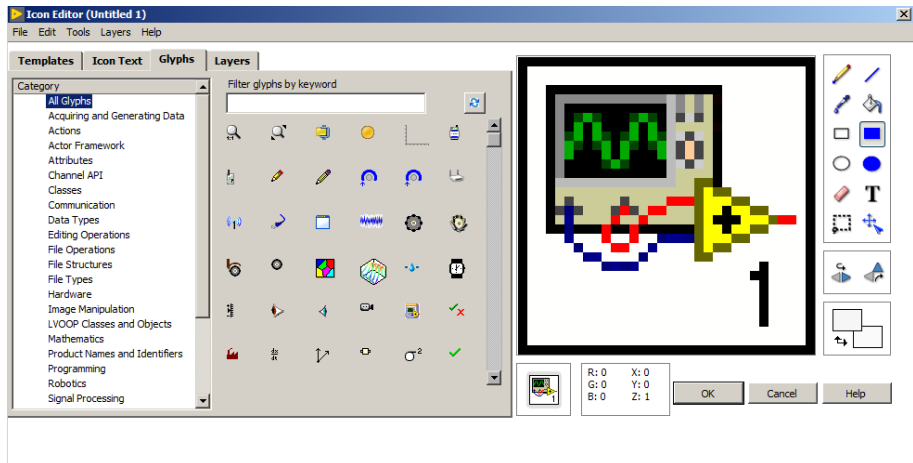
- Jednym ze sposobów utworzenia subVI jest zaznaczenie wybranego fragmentu kodu a potem wybranie: **Edit** → **Create SubVI**
- Fragment kodu zostanie zastąpiony przez subVI, który będzie widoczny jako ikona.



Ikona

- W przypadku wielu różnych subVI domyślne ikony nie reprezentują funkcjonalności danego subVI. Kod staje się nie czytelny.
- Dobrą praktyką jest zmiana wyglądu ikony subVI, tak aby przedstawiała funkcjonalność danego subVI.
- W celu modyfikacji ikony, kliknij na nią dwa razy a pojawi się okno do edycji.

Ikona



Zmodyfikowany przykład wykorzystujący subVI

The image displays two windows from the LabVIEW software interface, illustrating a modified example using a subVI.

Left Window: example33.vi Front Panel

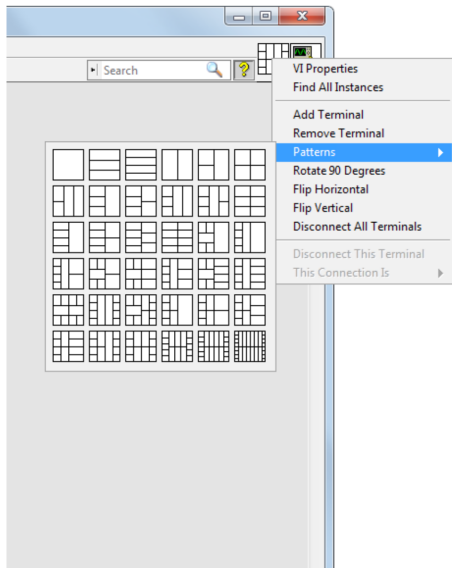
- Array 1:** A numeric array control with values [0, 1, 2, 3, 10, 0, 0].
- Array 2:** A numeric array control with values [0, 1, 5, 3, 0, 0, 0].
- Comparison:** A green indicator light labeled "Array 1 > Array 2".
- Control:** A red "STOP" button.

Right Window: example33.vi Block Diagram

- Array 1:** A numeric array control with a value of 4 (labeled "0#4").
- Array 2:** A numeric array control with a value of 3 (labeled "0#3").
- Mean Calculation:** Two "Mean" (x̄) blocks. The first block takes the value 4 and outputs 6,25. The second block takes the value 3 and outputs 3,00.
- Comparison:** A greater-than (>) comparison block that compares the two mean values (6,25 and 3,00).
- Output:** A green "TF" (True/False) indicator block that is set to "True" (TF).
- Stop:** A green "TF" (True/False) indicator block that is set to "False" (TF), which is connected to a red "stop" button.

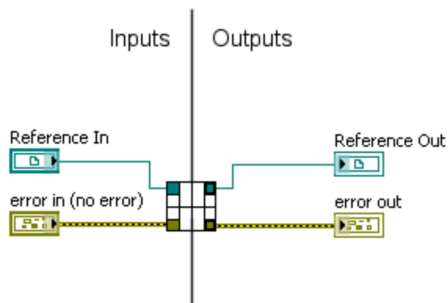
Połączenia

- Connector pane jest wyświetlany obok ikony w prawym górnym rogu na front panel-u.
- Reprezentuje wszystkie możliwe połączenia.
- Tworząc subVI możesz wybrać wzór rozłożenia terminali. Zalecany wzór to: 4x2x2x4 (lewy bok, góra, dół, prawy bok).
- Każdy terminal może być wymagany do podłączenia, rekomendowany lub opcjonalny. Jedną z powyższych opcji wybiera się poprzez kliknięcie na terminal na connector pane i wybranie **This Connection Is....**



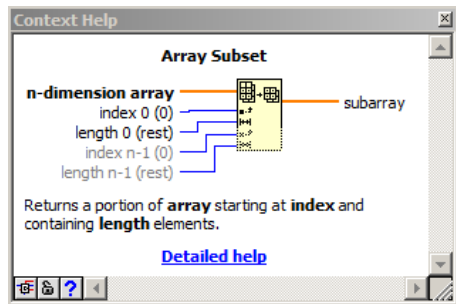
Standard podłączenia terminali do connector pane

- Umownie terminale po lewej stronie są przeznaczone do podłączenia danych wejściowych a terminale po prawej stronie do wyjściowe.
- Do górnych terminali zazwyczaj podłącza się: referencje czy główny element, na którym będzie wykonywana operacja.
- Dolne terminale są zazwyczaj używane do przekazywania klastra błędów.



Terminale w pomocy kontekstowej

- Pogrubione (Bold) - połączenia wymagane
- Standardowa czcionka (Plain) - połączenia rekomendowane
- Poszarzały tekst (Dimmed) - połączenia opcjonalne.



Dokumentacja programu

VI

- Name
- Description

Front Panel

- Label Names
- Tip Strips
- Descriptions
- Free Labels

Block Diagram

- Label Names
- Free Labels
- Owned Labels
- SubVI Descriptions

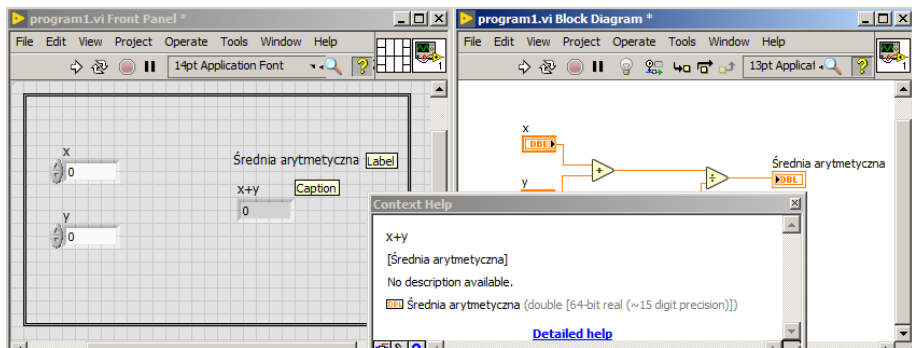
Labels

- opisują funkcjonalność kontroltek/indykatorów
- są wyświetlane w pomocy kontekstowej
- nazwa powinna zawierać wartość domyślną oraz jednostkę
- jeśli kontrolka ma ograniczony zakres dodaj go jako komentarz a nie jako część label.

The screenshot displays the LabVIEW interface for a program named 'program1.vi'. It is split into two main views: the 'Front Panel' and the 'Block Diagram'.
In the 'Front Panel', there are two numeric input controls labeled 'x' and 'y', both showing the value '0'. To the right, there is a numeric indicator control labeled 'Średnia arytmetyczna' (Arithmetic Mean) also showing '0'.
The 'Block Diagram' shows the logic behind the calculation. It features two input terminals labeled 'x' and 'y', both with a 'DBL' (double) data type. The 'x' terminal is connected to an addition block (+), and the 'y' terminal is also connected to the same addition block. The output of this addition block is connected to a division block (÷), which also has a 'DBL' data type. The output of the division block is connected to an output terminal labeled 'Średnia arytmetyczna' with a 'DBL' data type.
A 'Context Help' window is open in the foreground, displaying the text: 'Średnia arytmetyczna', 'No description available.', and 'DBL Średnia arytmetyczna (double [64-bit real (~15 digit precision)])'. Below this text is a blue link labeled 'Detailed help'.

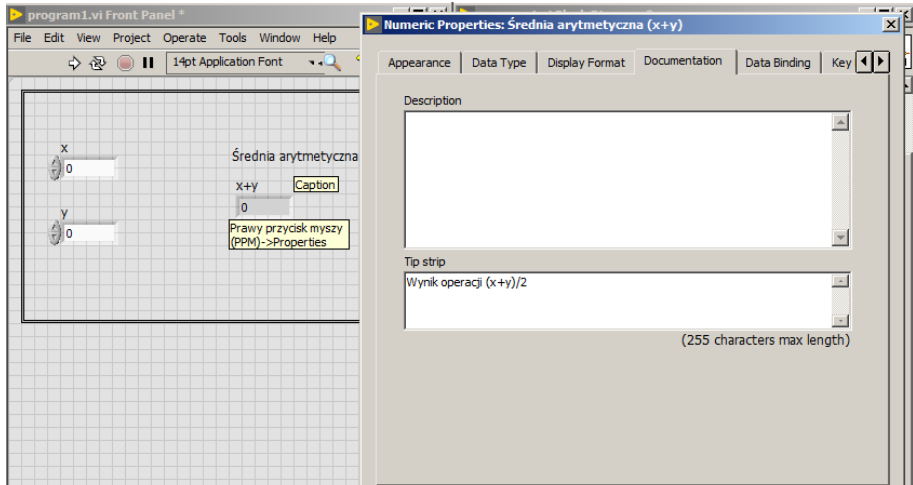
Captions

- stanowi dodatkowy opis kontrolki/indykatora wyświetlany na front panel-u.
- captions nie są wyświetlane na block diagramie zatem jeśli label byłaby zbyt długa to można ją podzielić i część informacji przenieść do caption oszczędzając miejsce na block diagramie.
- captions są wyświetlane jako tip strip w pomocy kontekstowej.



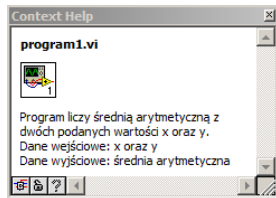
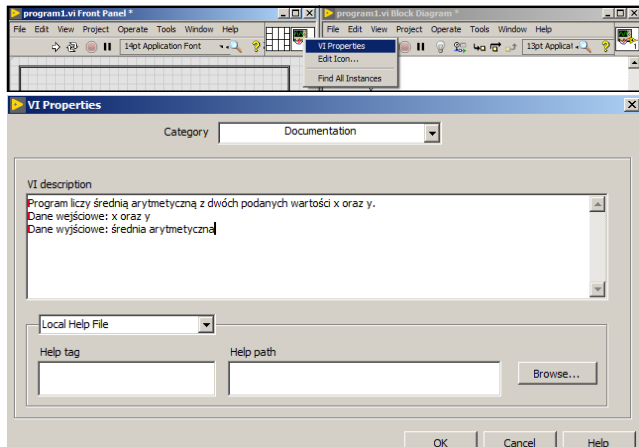
Tip strip

- Tip strip to dodatkowy opis pojawiający się na żółtym polu po najechaniu na kontrolkę/indykator.



VI/subVI description powinno zawierać:

- opis przeznaczenia VI
- instrukcja używania VI
- opis danych wejściowych (kontrolki) oraz wyjściowych (indykatorów)



Komentarze (*Free labels*)

The image displays two windows from the LabVIEW environment: the Front Panel and the Block Diagram.

Front Panel (Left Window): Titled "program1.vi Front Panel *". It contains a main label "Program liczy średnią arytmetyczną z dwóch podanych wartości x+y" with a "Free Label" property. Below this are two numeric input controls labeled "x" and "y", both with a value of "0". To the right, there is a numeric output control labeled "Średnia arytmetyczna" with a "Caption" property, also showing "0".

Block Diagram (Right Window): Titled "program1.vi Block Diagram *". It shows the logic for the program. Two double-precision floating-point (DBL) inputs labeled "x" and "y" are connected to an addition (+) block. The output of the addition block is connected to a division (÷) block. A constant value of "2" is connected to the denominator input of the division block. The output of the division block is connected to a DBL output control labeled "Średnia arytmetyczna".

Opis połączeń (*Wiring description*)

The image displays four screenshots of the LabVIEW interface, illustrating the wiring process for a program that calculates the arithmetic mean of two numbers.

Top Row (Initial State):

- Left Panel (Front Panel):** Shows the user interface with two input controls labeled 'x' and 'y', both set to 0. A label 'Średnia arytmetyczna' is present, along with a 'Caption' and a 'Label'.
- Right Panel (Block Diagram):** Shows the initial wiring. Two 'DBL' (Double) inputs labeled 'x' and 'y' are connected to a sum block (represented by a triangle with a plus sign). A context menu is open over the sum block, with 'Visible Items' selected, showing a 'Label' option.

Bottom Row (Final State):

- Left Panel (Front Panel):** Identical to the top-left panel, showing the user interface.
- Right Panel (Block Diagram):** Shows the final wiring. The sum block is now labeled 'x+y'. Its output is connected to a divide block (represented by a triangle with a division sign). The divide block also receives a constant value '2' as a second input. The output of the divide block is connected to a 'DBL' control labeled 'Średnia arytmetyczna'.

Timing VI

Do czego są nam potrzebne funkcje kontrolujące czas w VI?

- Do kontroli częstotliwości z jaką wykonuje się pętla Loop.
- Do zapewnienia czasu procesorowi na kontrolę innych zadań np. przetwarzania interfejsu użytkownika.

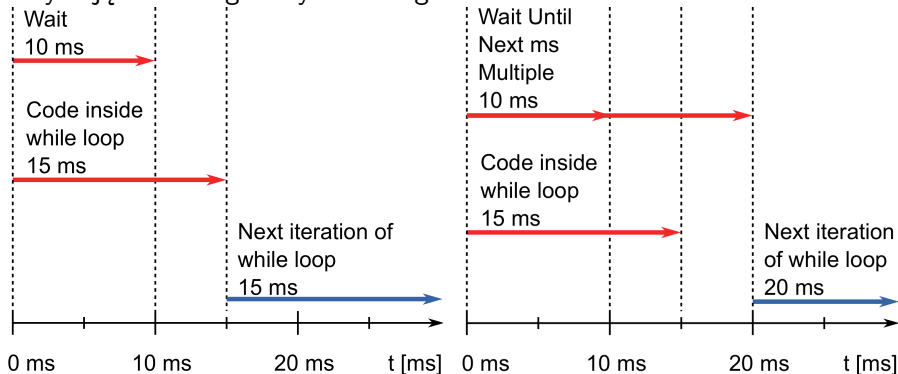


Wait Functions

Użycie wait function pozwala na:

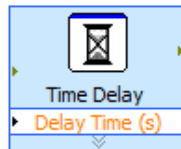
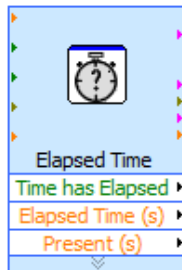
- uśpienie programu na określoną ilość czasu
- wykonanie innych zadań przez procesor

Korzystają one z zegara systemowego.

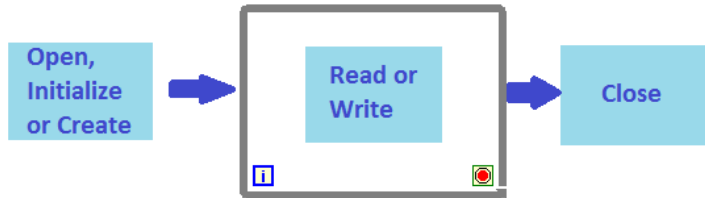


Express VI

- Nie zapewniają czasu na wykonanie innych zadań procesorowi.
- Elapsed Time określa ile czasu upłynęło po określonym punkcie w VI.
- Time Delay działa analogicznie jak funkcja Wait (ms).



Schemat operacji na pliku



Formaty plików obsługiwanych przez LabVIEW

- **Binary** - wydajny format ale trudny do przeczytania przez ludzi. Powszechnie używany do wielokanałowych DAQ.
- **ASCII** - dane są zapisane w postaci łańcuchów znaków. Powszechnie używany przez low-speed DAQ.
- **LVM** - format budowany z wykorzystaniem ASCII. Dane z LabVIEW są rozdzielone tabulatorami co ułatwia ich wczytywanie przez arkusze kalkulacyjne.
- **TDMS** - format stworzony przez NI, który jest specyficzną wersją formatu binarnego. Przechowuje on informacje zarówno liczbowe jak i właściwości zapisanych obiektów.

Format ASCII

Kiedy zaleca się używanie formatu ASCII?:

- Jeśli chcesz mieć dostęp do plików z poziomu innych aplikacji.
- Nie ma dla Ciebie dużego znaczenia szybkość odczytu/zapisu danych.
- Brak ograniczeń na rozmiar pliku.
- Precyzja numeryczna nie jest ważna.

High-Level File I/O vs. Low-Level File I/O

1 High-Level File I/O

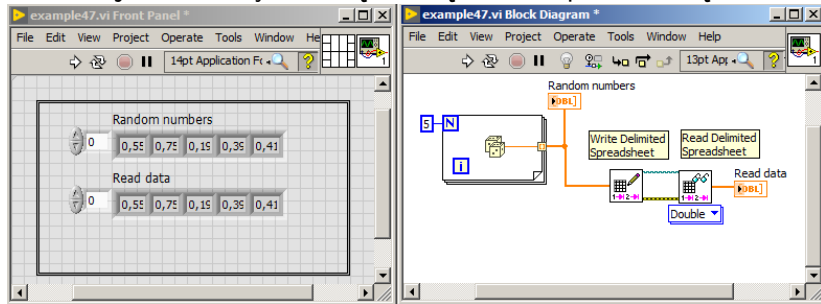
- zawiera funkcje, które realizują od razu trzy kroki obsługi plików.
- funkcje High-Level są bardzo wygodne jednak mało efektywne.

2 Low-Level File I/O

- zawiera osobne funkcje do realizacji poszczególnych kroków operacji na plikach.
- funkcje te zapewniają znacznie efektywniejszą obsługę plików i są polecane w sytuacjach gdy prędkość odczytu/zapisu danych ma duże znaczenie.

High-Level File I/O

- **Write Delimited Spreadsheet** - konwertuje tablicę z wartościami zmiennoprzecinkowymi na łańcuch znaków i zapisuje do pliku ASCII
- **Read Delimited Spreadsheet** - odczytuje określoną liczbę wierszy i konwertuje na dwuwymiarową tablicę zmiennoprzecinkową.



- **Write To/Read From Measurement File** - funkcje do zapisu i odczytu danych z formatu LVM/TDMS.

Low-Level File I/O

The image shows the LabVIEW interface for 'example48.vi', divided into two main windows: the Front Panel and the Block Diagram.

example48.vi Front Panel:

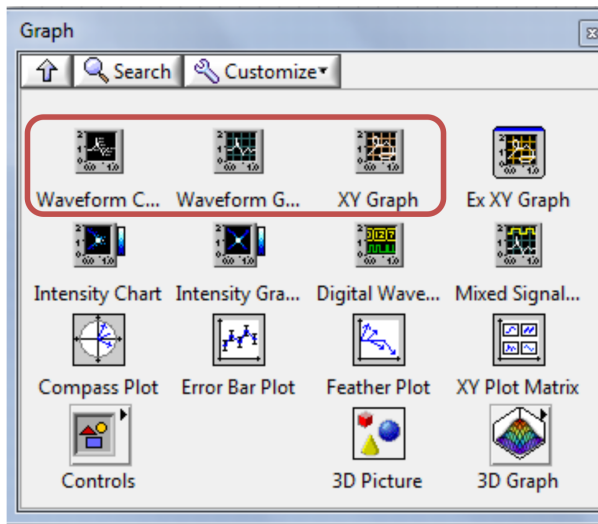
- Random numbers:** A control knob set to 0, with a display showing a row of six random numbers: 0,0804, 0,7277, 0,8452, 0,362E, 0,618C.
- Read data:** A control knob set to 0, with a display showing a row of six numbers: 0,08, 0,73, 0,85, 0,36, 0,62.
- results — Notatnik:** A text area displaying the same six numbers as the 'Read data' control: 0,08, 0,73, 0,85, 0,36, 0,62.

example48.vi Block Diagram:

- 1. Build Path:** A file path is constructed from a constant 'results.txt'.
- 2. Open/Create/Replace File:** The file is opened with the mode 'replace or create'.
- 3. Conversion double to string and adding tab:** The random numbers are converted to strings using a '%.2f' format string, and a tab character is added between them.
- 4. Write to text file:** The formatted string is written to the file.
- 5. Close file:** The file is closed.
- 6. Open/Create/Replace File:** The file is opened again to read the data.
- 7. Read from Text File:** The data is read from the file.
- 8. Spreadsheet String To Array:** The read string is converted into a string array.

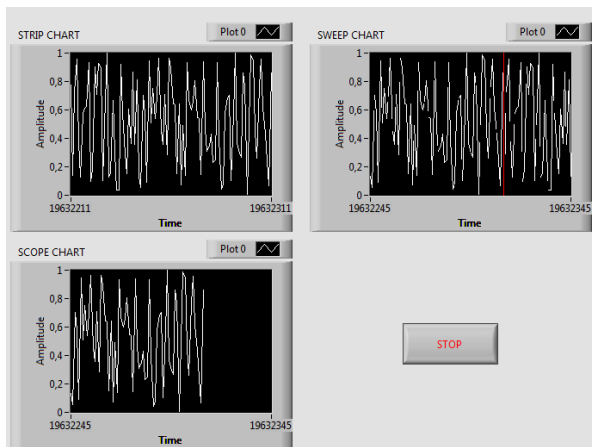
The Block Diagram also includes an error handling mechanism (Error In) and a 'Random numbers' control that feeds into the conversion step.

Wykresy

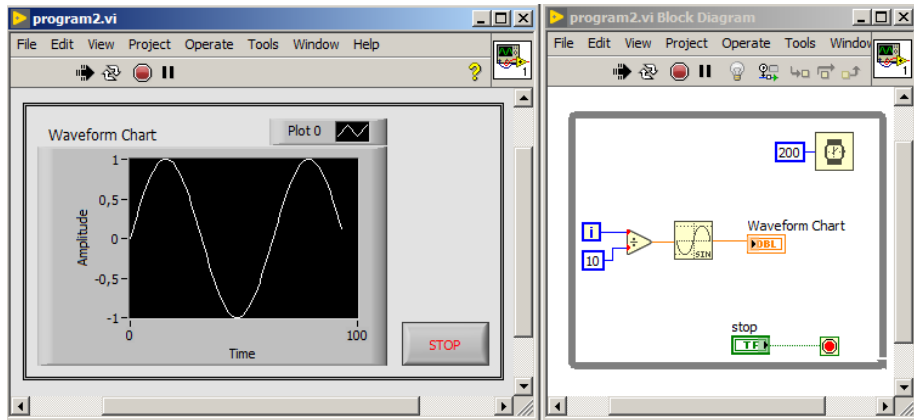


Waveform Chart

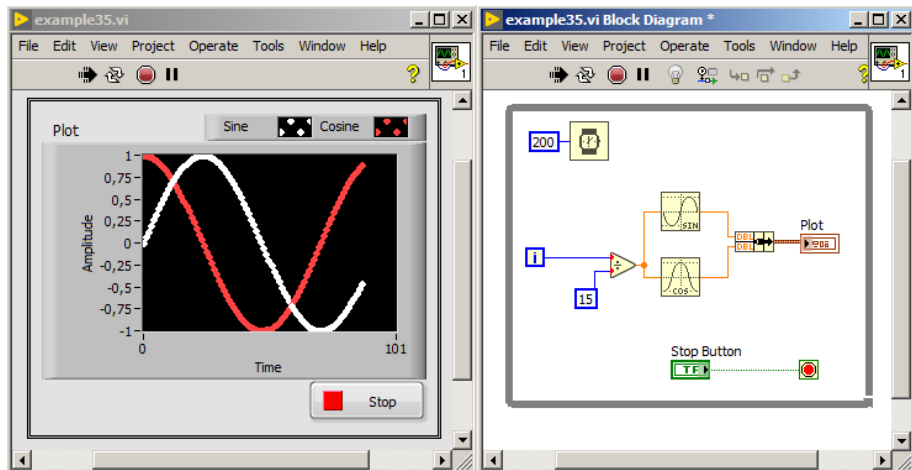
- Wyświetla na bieżąco punkt po punkcie przy czym punkty są oddalone o równą odległość na osi X.
- Wszystkie punkty są przechowywane w pamięci przez Waveform Chart.
- Na poniższym rysunku przedstawiono trzy możliwe sposoby wizualizacji danych.



Wyświetlanie jednej serii pomiarowej na Waveform Chart

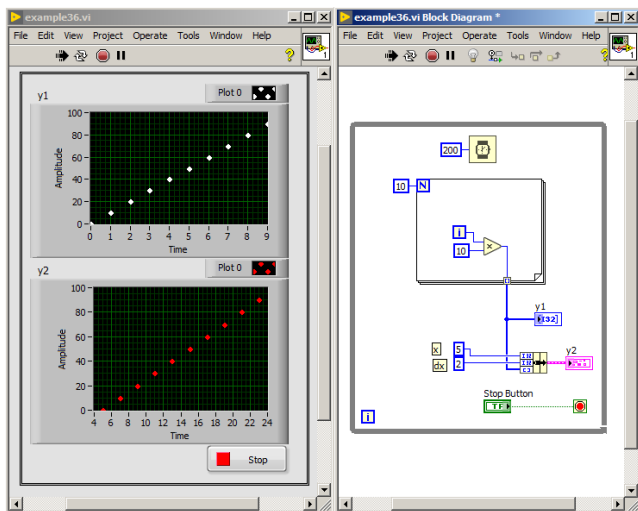


Wyświetlanie kilku serii pomiarowej na Waveform Chart



Waveform Graph

- Wyświetla całą serię punktów zamiast dodawać punkt po punkcie.

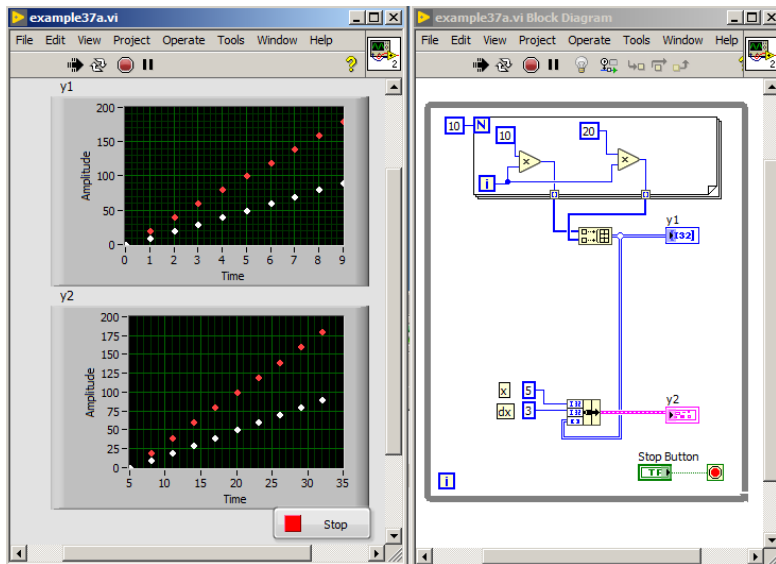


Wyświetlenie kilku serii pomiarowych na Waveform Graph

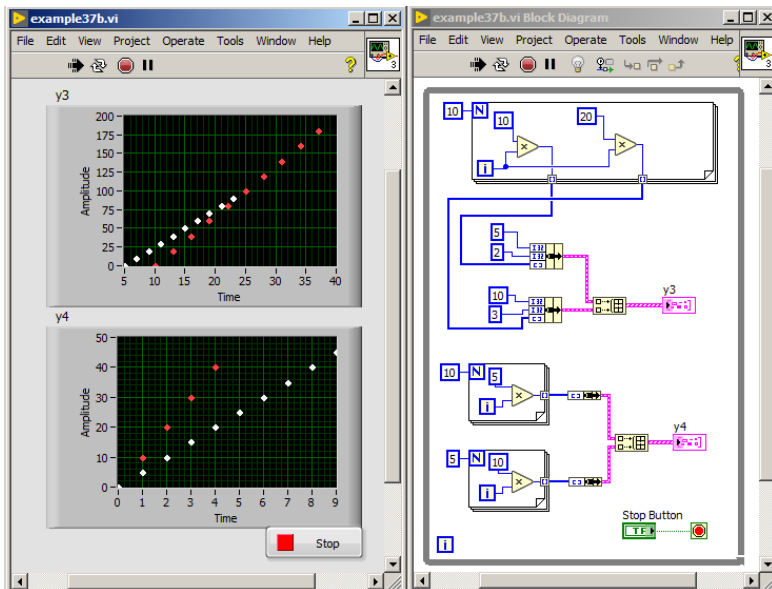
Do wykresu typu waveform graph można również przekazać kilka serii pomiarowych na wiele sposobów:

- 1 **poprzez przekazanie tablicy 2D gdzie pierwszy rząd to punkty y dla pierwszej serii, drugi rząd to punkty y dla drugiej serii itd.**
- 2 poprzez przekazanie klastra, który zawiera w podanej kolejności: wartość x, zmiana wartości x, tablicę 2D o strukturze opisanej powyżej.
- 3 poprzez przekazanie tablicy klastrów, których poszczególne elementy to wartość x, zmiana x, tablica 1D wartości y.
- 4 poprzez przekazanie tablicy 1D klastrów gdzie każdy klaster zawiera tablicę wartości y. Taka metoda jest stosowana głównie dla przypadków gdy kilka serii pomiarowych różni się ilością punktów.

Wyświetlenie kilku serii pomiarowych na Waveform Graph

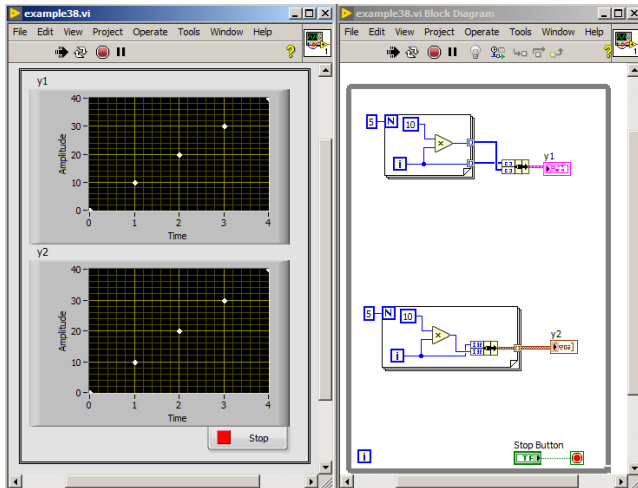


Wyświetlenie kilku serii pomiarowych na Waveform Graph

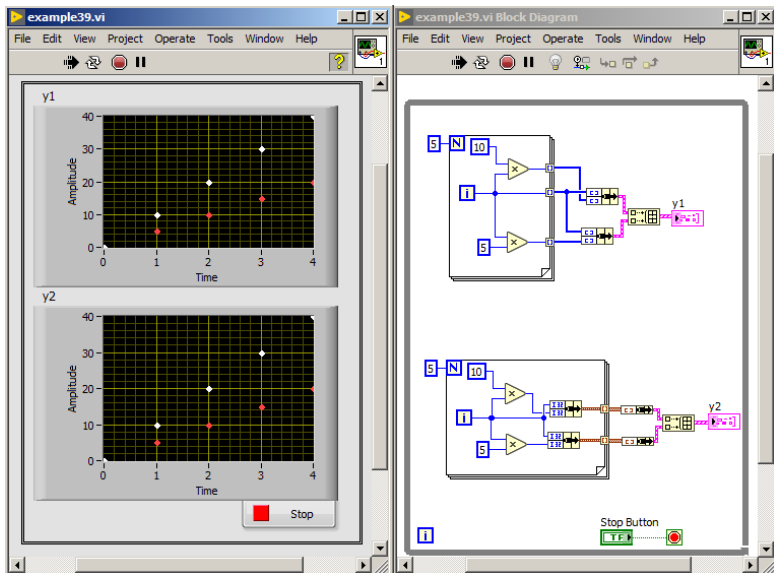


XY Graph

- Umożliwia umieszczenie punktów nierównomiernie rozłożonych



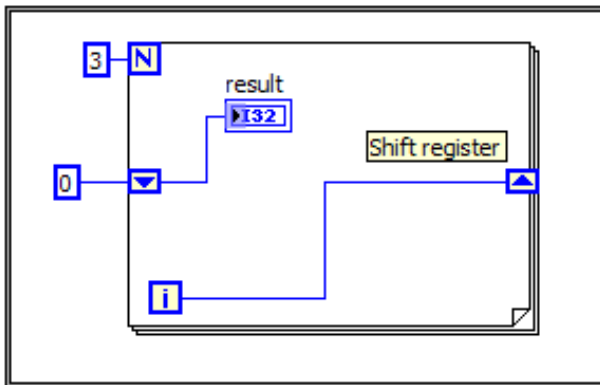
Wyświetlenie kilku serii pomiarowych na XY Graph



Rejestr przesuwny

- Często potrzebujemy informacji o wartości danej zmiennej z poprzedniej iteracji pętli Loop.
- Rejestr przesuwny przekazuje dane z jednej iteracji pętli do kolejnej.
- W celu stworzenia rejestru przesuwanego, kliknij na granicę pętli i wybierz: **Add Shift Register**.
- Lewy rejestr przekazuje dane z poprzedniej iteracji. Do prawego rejestru są przekazywane dane z obecnej iteracji.

Rejestr przesuwny z wartością początkową



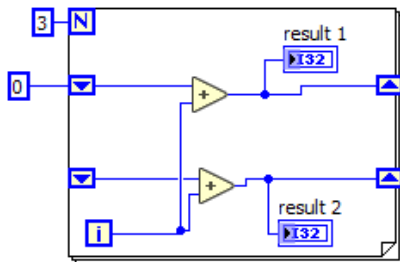
i	left terminal	result indicator	right terminal
0	0	0	0
1	0	0	1
2	1	1	2

Wartości domyślne

Jeśli nie zadeklarowaliśmy wartości początkowej rejestru przesuwnego to zostanie użyta wartość domyślna dla danego typu:

- Numeric - 0
- Boolean - FALSE
- String - Pusty znak

Rejestr przesuwny bez wartości początkowej



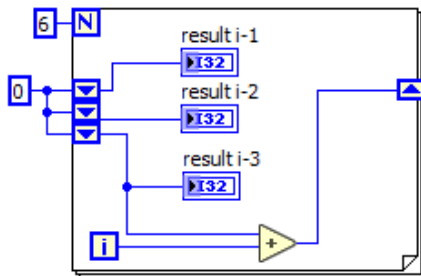
Pierwsze uruchomienie programu

i	left terminal	result 1 indicator	right terminal	left terminal	result 2 indicator	right terminal
0	0	0 (0+0)	0	0	0 (0+0)	0
1	0	1 (0+1)	1	0	1 (0+1)	1
2	1	3 (1+2)	2	1	3 (1+2)	3

Drugie uruchomienie programu

0	0	0 (0+0)	0	3	3 (3+0)	3
1	0	1 (0+1)	1	3	4 (3+1)	4
2	1	3 (1+2)	3	4	6 (4+2)	6

Rejestr przesuwny z kilkoma lewymi terminalami



i	result i-1	result i-2	result i-3	right terminal
0	0	0	0	0 (0+0)
1	0	0	0	1 (0+1)
2	1	0	0	2 (0+2)
3	2	1	0	3 (0+3)
4	3	2	1	5 (1+4)
5	5	3	2	7 (2+5)



Itasdi

Dziękuję za uwagę!

Wsparcie Komisji Europejskiej dla produkcji tej publikacji nie stanowi poparcia dla treści, które odzwierciedlają jedynie poglądy autorów, a Komisja nie może zostać pociągnięta do odpowiedzialności za jakiegokolwiek wykorzystanie informacji w niej zawartych.

Wykład został opracowany w oparciu o materiały: "LabVIEW Core 1 Course Manual", "LabVIEW Core 2 Course Manual", pierwotną wersję wykładu: mgr. inż. Marcina Biedy oraz przykładowe egzaminy CLAD opublikowane na stronie www.ni.com.