

Wykład 1
6.10.2026 r.

Wprowadzenie, wstęp
teoretyczny dotyczący obsługi
Labview

Katarzyna Gwóźdź



Prowadzący

dr inż. Katarzyna Gwóźdź

Katedra Fizyki Doświadczalnej

Wydział Podstawowych Problemów Techniki

p. 231A bud. A1 Wybrzeże Wyspiańskiego 27

Konsultacje: pn 13-14, wt. 9-11 pt. 9-10

katarzyna.r.gwozdz@pwr.edu.pl

<https://gwozdz.wppt.pwr.edu.pl/>



Plan zajęć

1. Wprowadzenie (2h)
- 2 i 3. Funkcje i struktury (4h)
4. Dokumentacja i wizualizacja danych (2h)
5. Macierze, tablice, klastry (2h)
6. Funkcje graficzne (2h)
7. Struktura event (2h)



Plan zajęć

- 8. Obsługa joysticka (2h)
- 9. Komunikacja z komputerem (2h)
- 10. Wyświetlacz 7-segmentowy (2h)
- 11. Potencjometr, przełącznik DIP (2h)
- 12. Silniczek servo (2h)
- 13. Wstęp do projektu (2h)
- 14 - 15. Projekt, zaliczenie (4h)



Zaliczenie

Na podstawie:

- Projektu - ostatnie trzy zajęcia + praca poza zajęciami
- Na ostatnich zajęciach krótka prezentacja projektu
- Aktywności
- Pracy na zajęciach
- Obecności - dopuszczalne dwie nieobecności



Materiały

- Dokumentacja labview ->
<http://www.ni.com/getting-started/labview-basics/online-help>
- Wykłady
- „Labview w praktyce” Marcin Chruściel, Wydawnictwo BTC, 2008.
- „Środowisko LabView™ w eksperymencie wspomaganym komputerowo” Wiesław Tłaczała, Wydawnictwo WNT, 2014.



Labview dla studentów

Opis instalacji:

<https://di.pwr.edu.pl/oprogramowanie/labview>





NATIONAL INSTRUMENTS

LabVIEW™

Laboratory Virtual Instrument Engineering Workbench

„Środowisko to od początku prezentowało **odmienne podejście do sposobu tworzenia programu**. Tworzenie programu polega w nim na łączeniu różnych graficznych **ikon**, z których każda odpowiada za wykonywanie różnych funkcji. Wykonywanie programu nie jest w tym przypadku realizowane poprzez wykonywanie poszczególnych **linii kodu, ponieważ takowe nie występują**. (...)

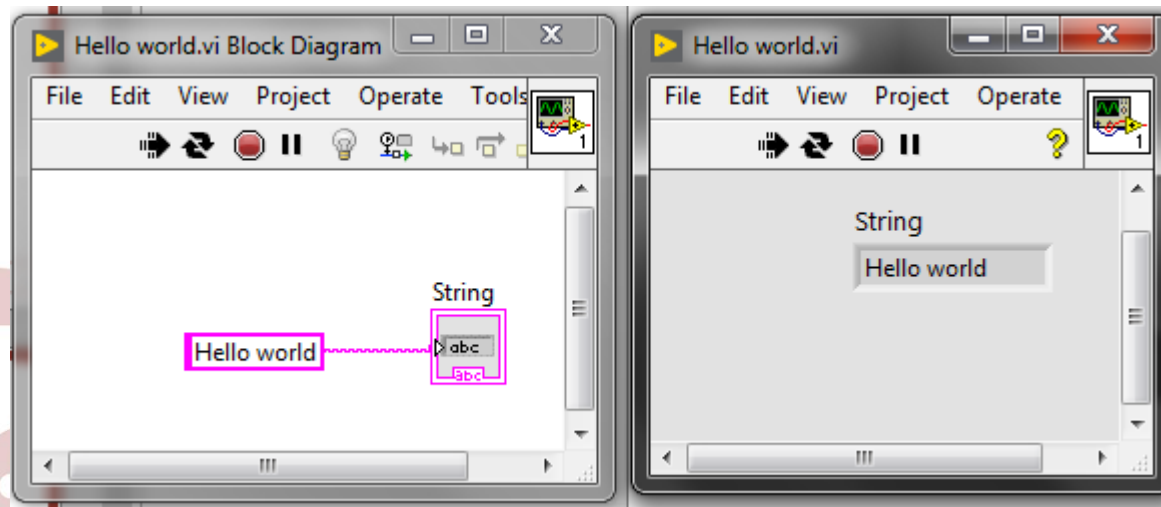
Autorzy środowiska przeznaczyli je od samego początku dla **inżynierów i naukowców**, którzy właśnie w ten sposób rozwiązują problemy. Dlatego też uznali, że właśnie takie podejście do programowania będzie najbardziej naturalne dla osób, które będą używały LabVIEW.”

Wikipedia



```
Hello.cpp* x
(Global Scope)
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     //say hello
7     cout << "Hello C++" << endl;
8
9     system("PAUSE");
10    return 0;
11 }
12
```

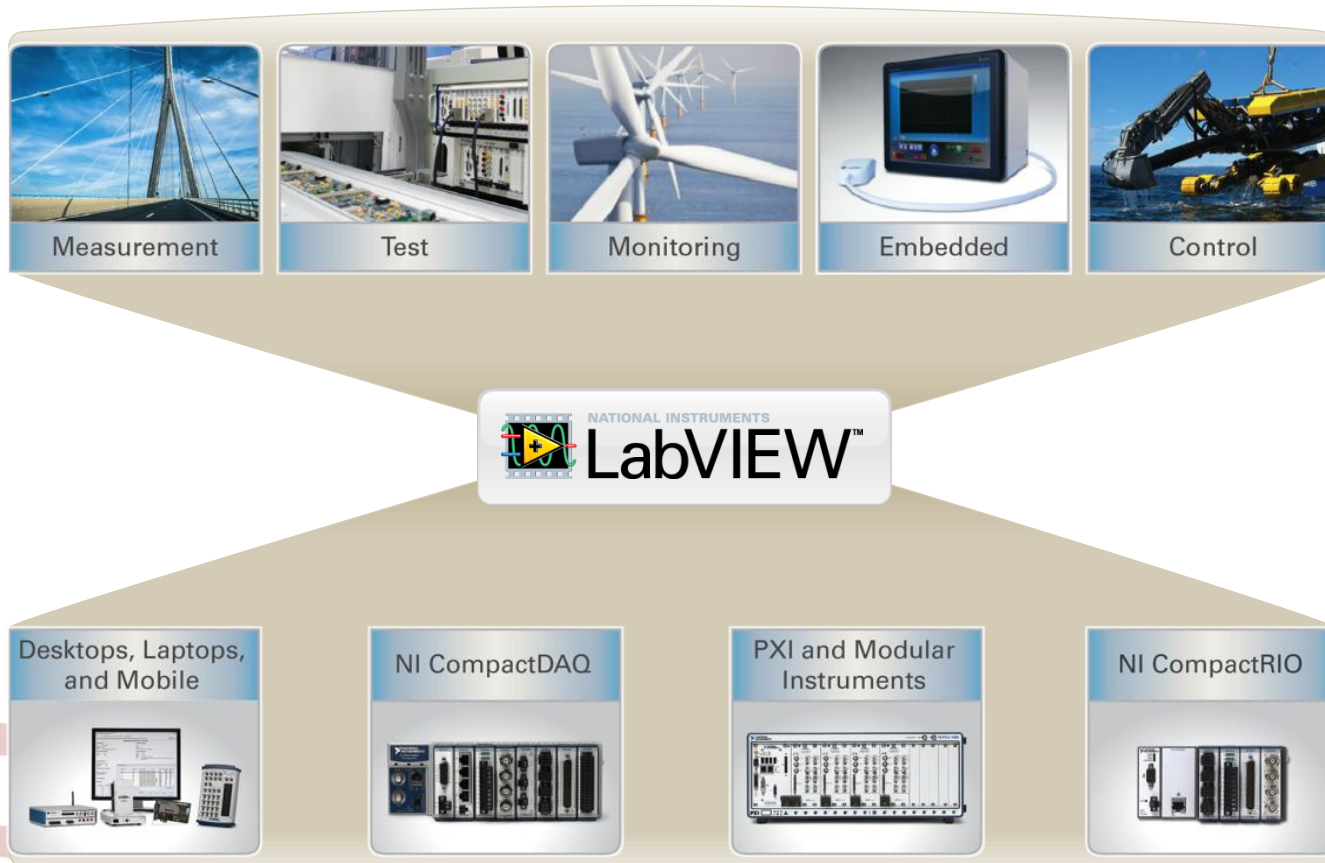
Język C++



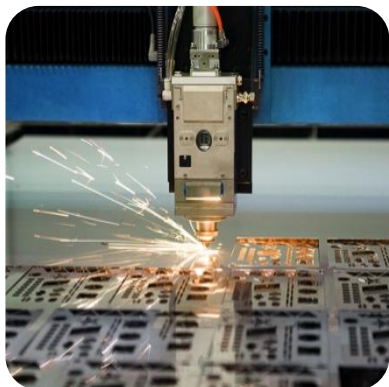
Język G



Zastosowania



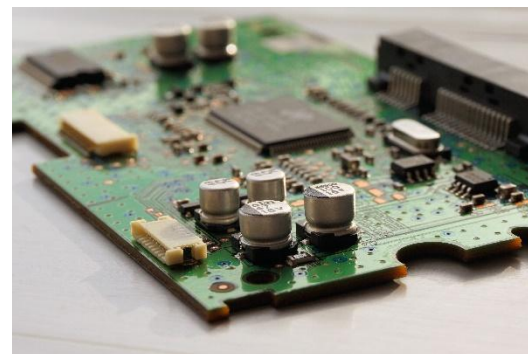
Zastosowania w przemyśle



Technologie precyzyjne



Lotnictwo



Elektronika



Telekomunikacja



Energetyka

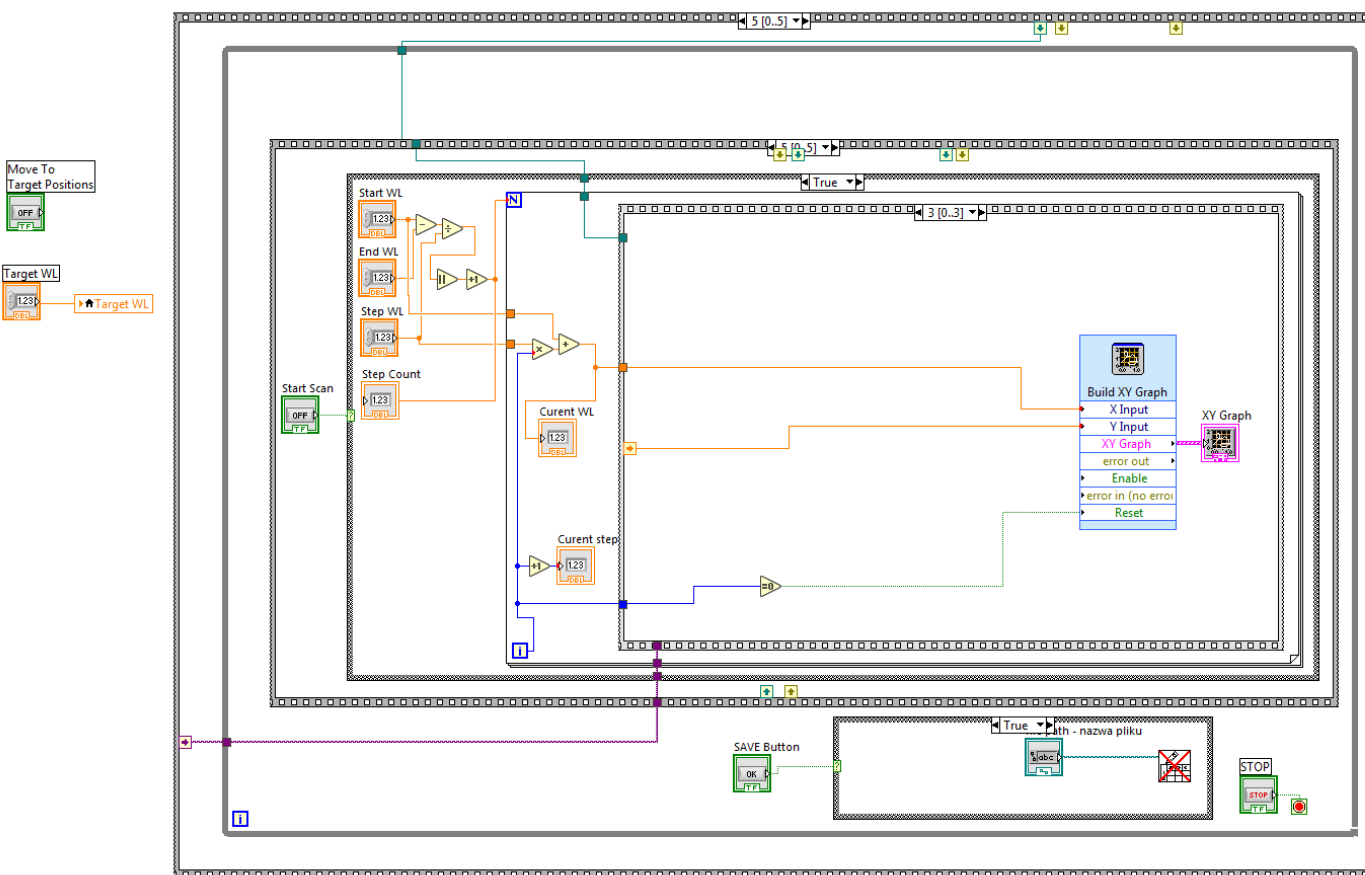


Transport



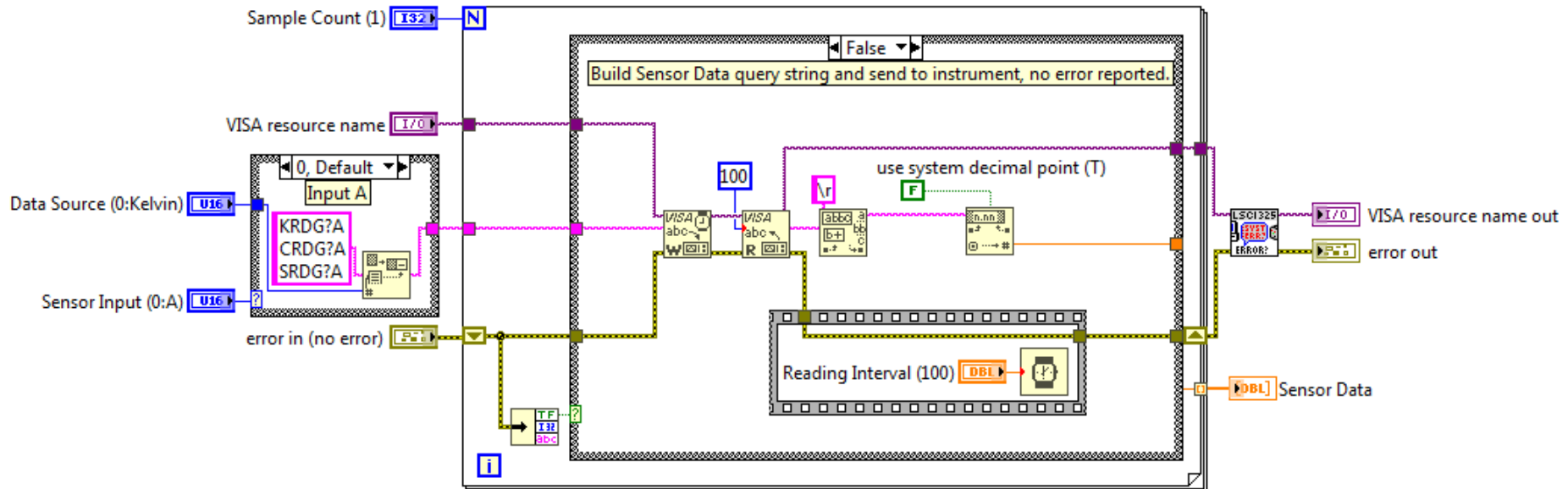
Zastosowania w K-63

Kamera CCD Symphony II do pomiaru fotoluminescencji



Zastosowania w K-63

Kontroler temperatury LakeShore 325

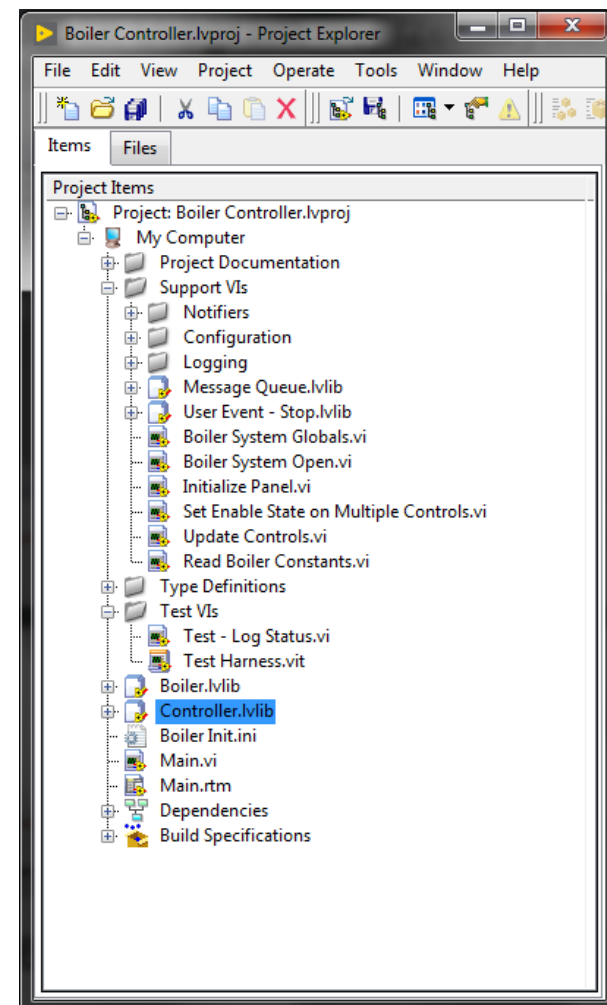
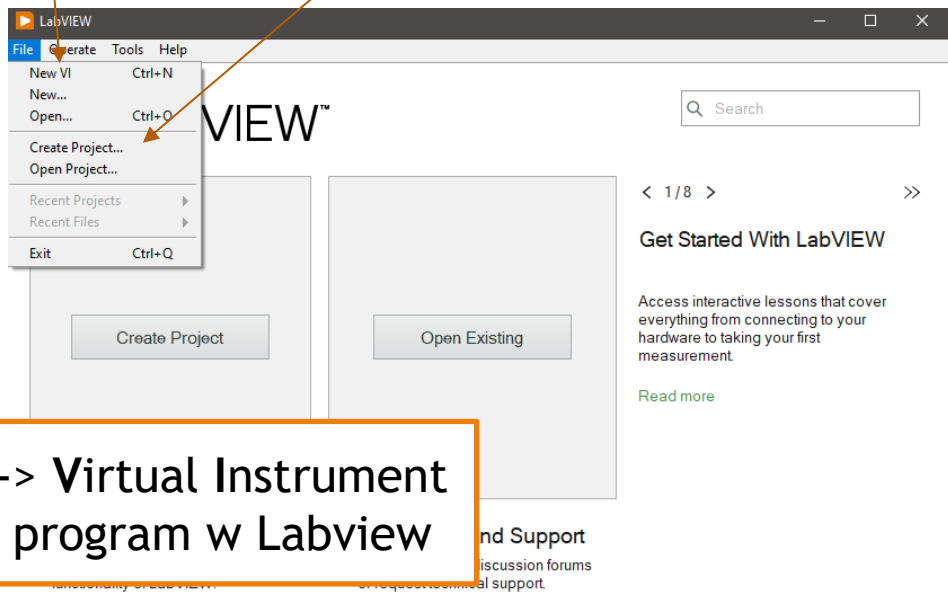


Ctrl+N

Nowy program

Nowy projekt

Okno startowe

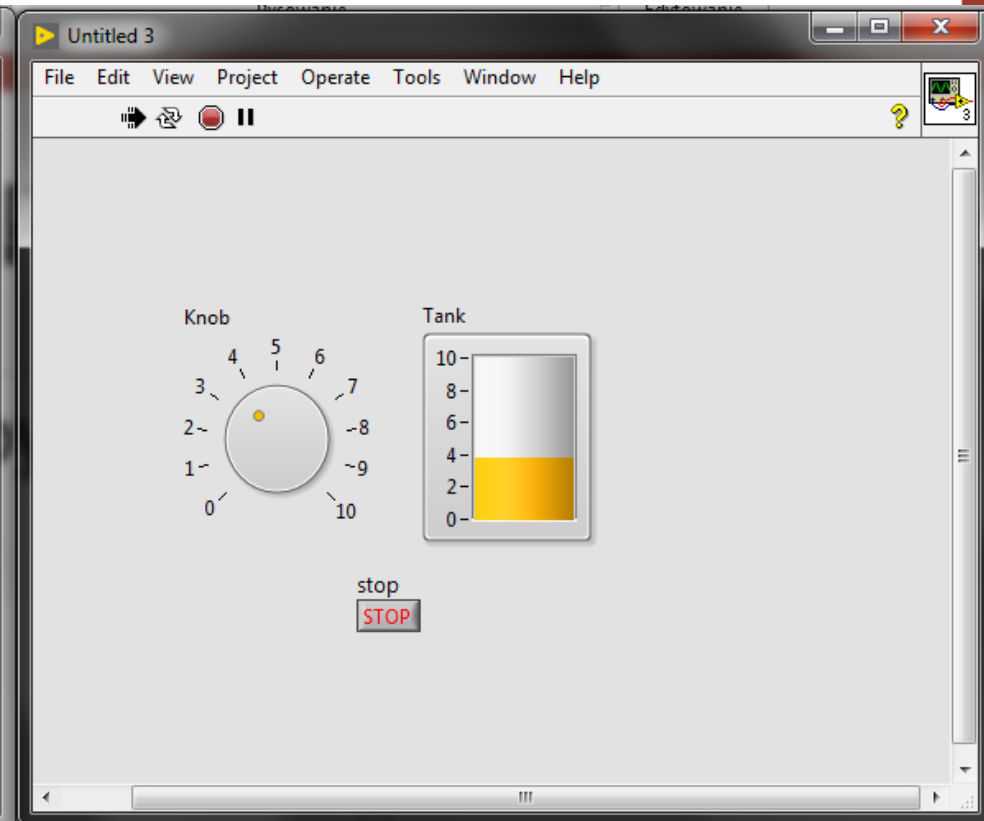
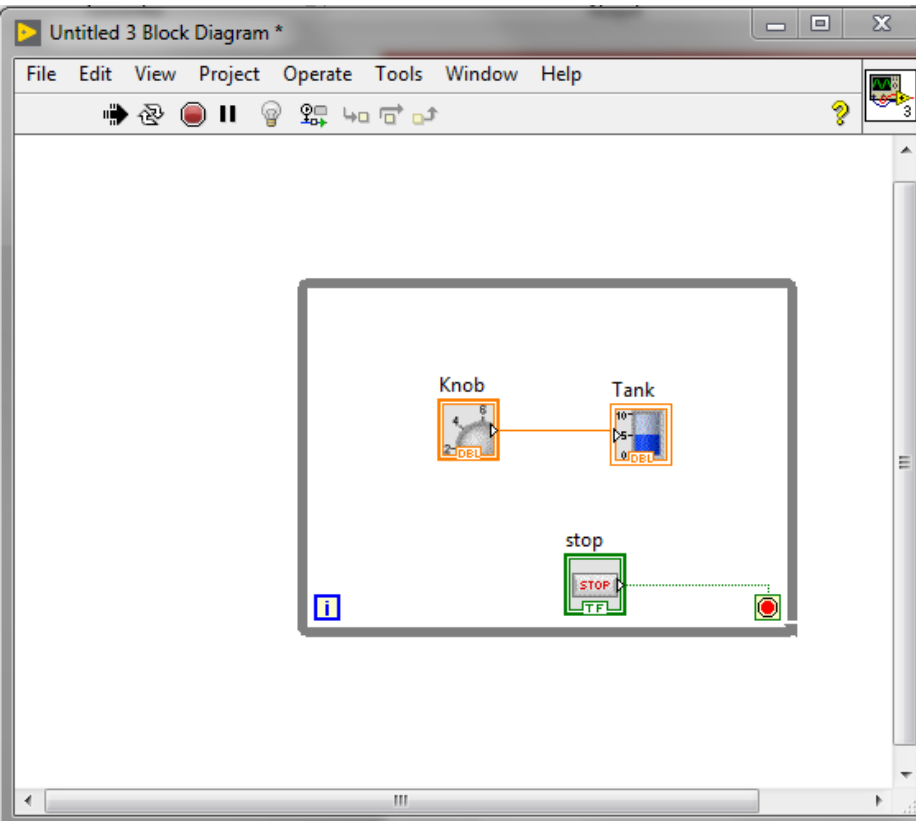


VI -> Virtual Instrument
-> program w Labview



Block diagram

Front panel

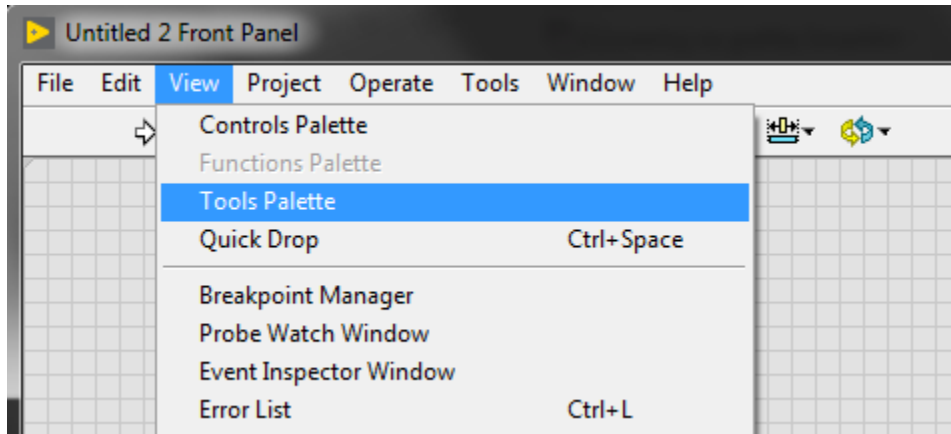


Funkcje, przewody, subvi, struktury...

Kontrolki, indykatory, opisy, interfejs użytkownika...



Tools palette



Auto

Zmiana wartości

Łączenie
przewodów

Zmiana koloru

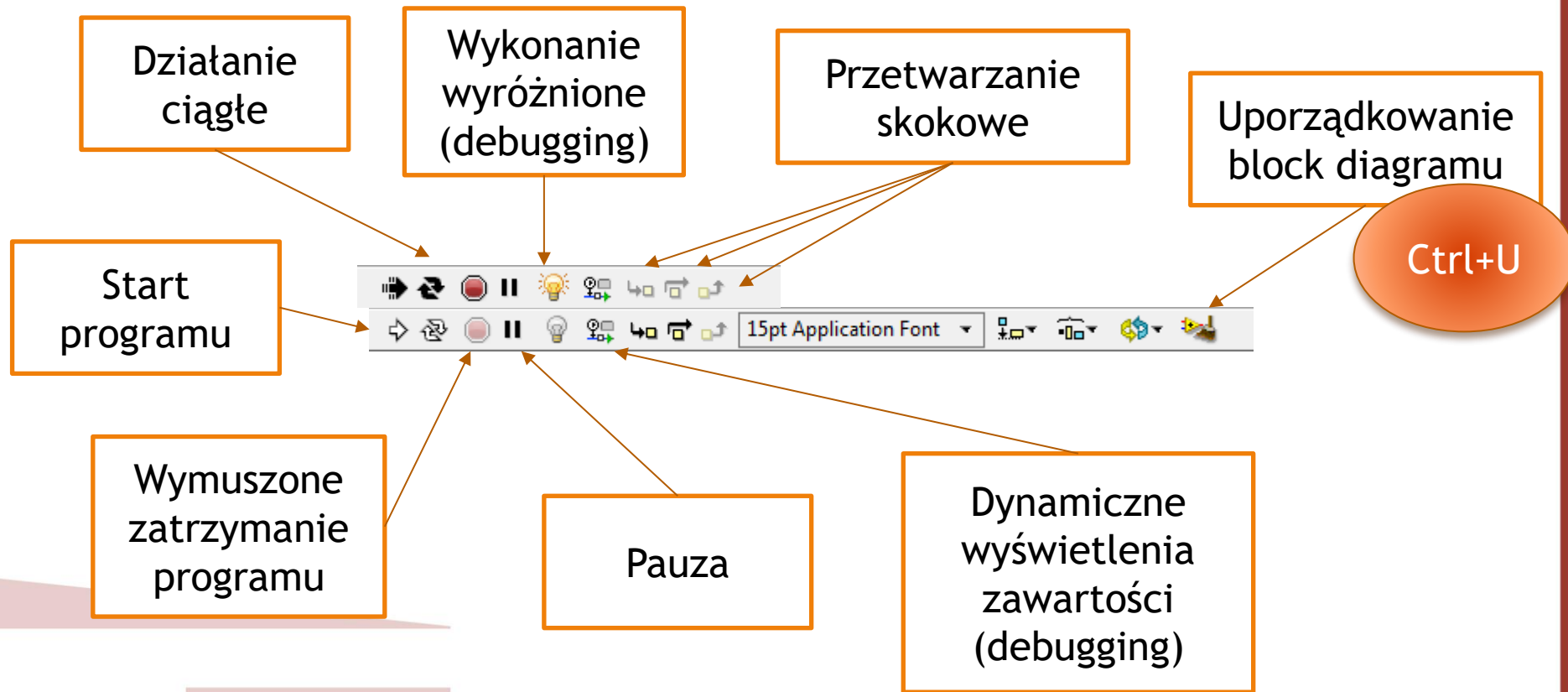


Edycja tekstu

Przewijanie



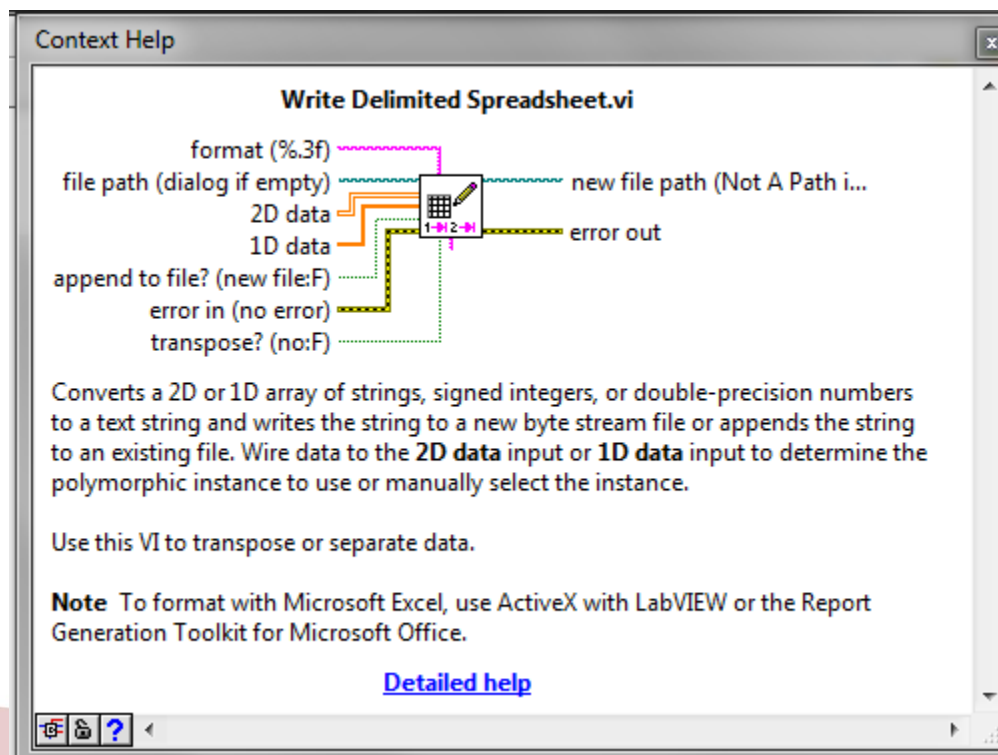
Status toolbar





Ctrl+H

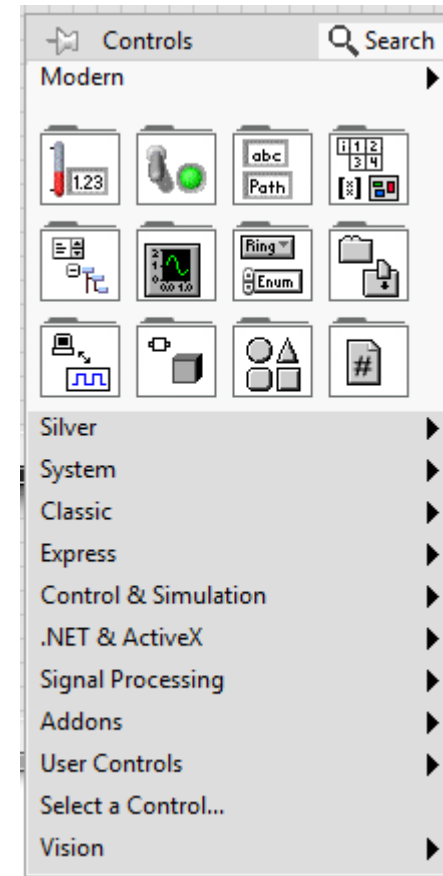
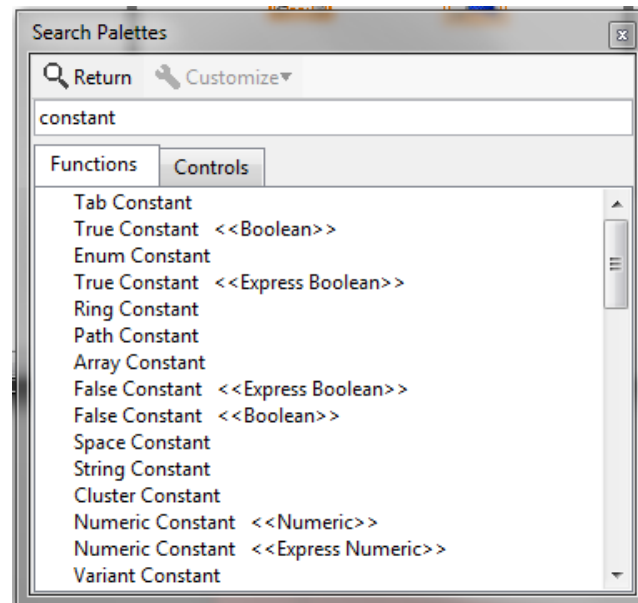
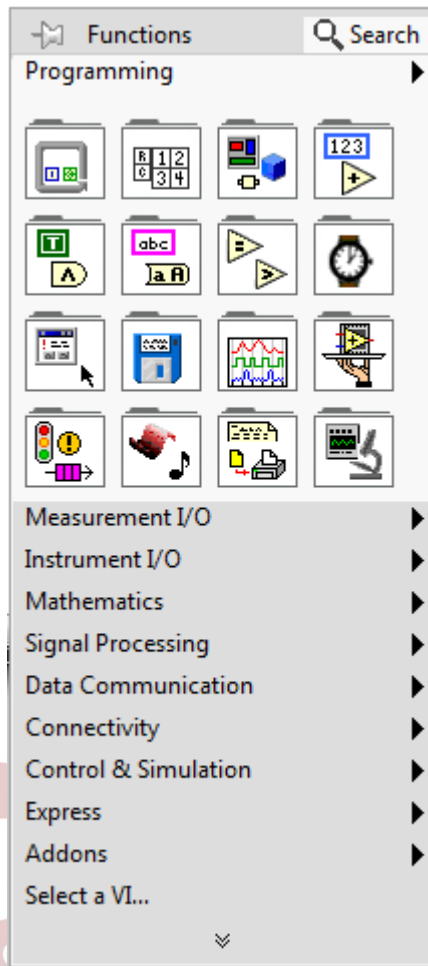
Context help



Wyświetla
podstawowe
informacje o
wybranej funkcji,
przewodzie,
kontrolce... po
najechnaniu kursorem
myszki.

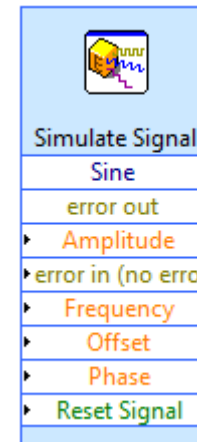


Wybór funkcji

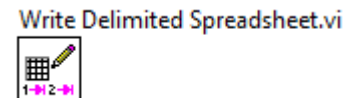


Nodes (węzły)

- Express Vi - konfigurowalne w osobnym okienku



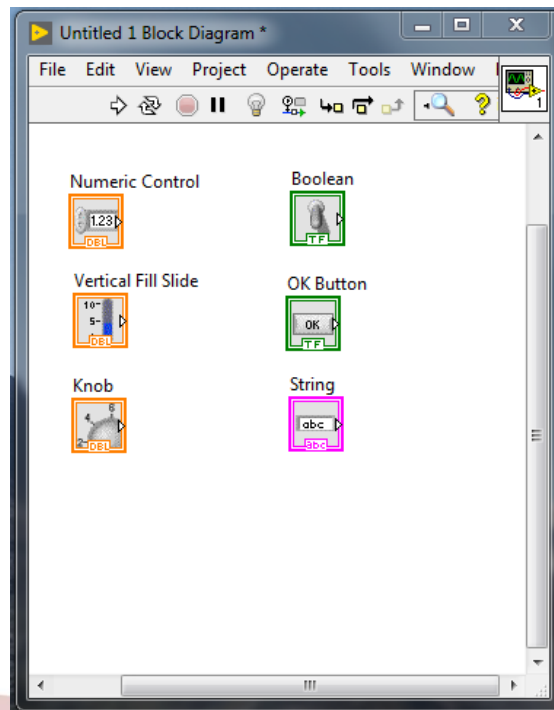
- Standard Vi - ograniczona konfiguracja, zależna od danych wejściowych
- Proste funkcje - niekonfigurowalne



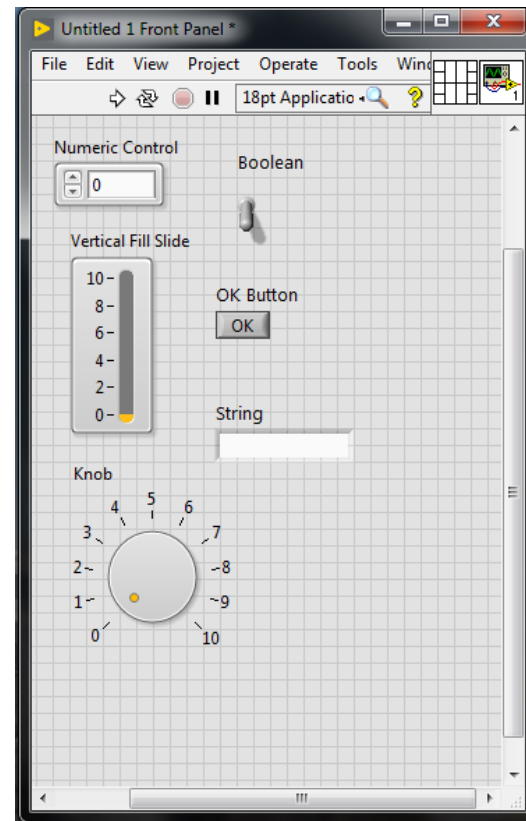
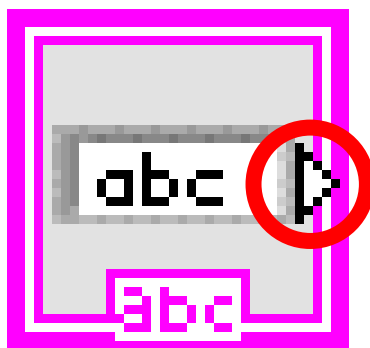
Kontrolki

Dane wejściowe do programu:

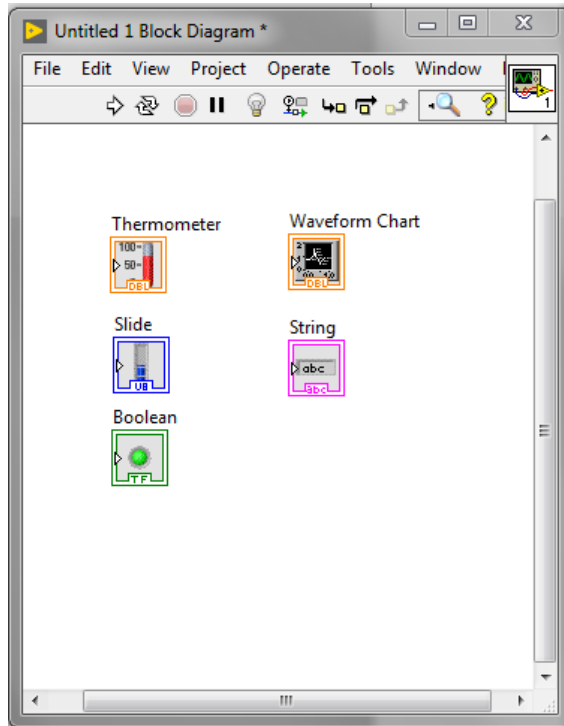
- pokrętła,
- suwaki,
- przyciski,
- przełączniki...



String



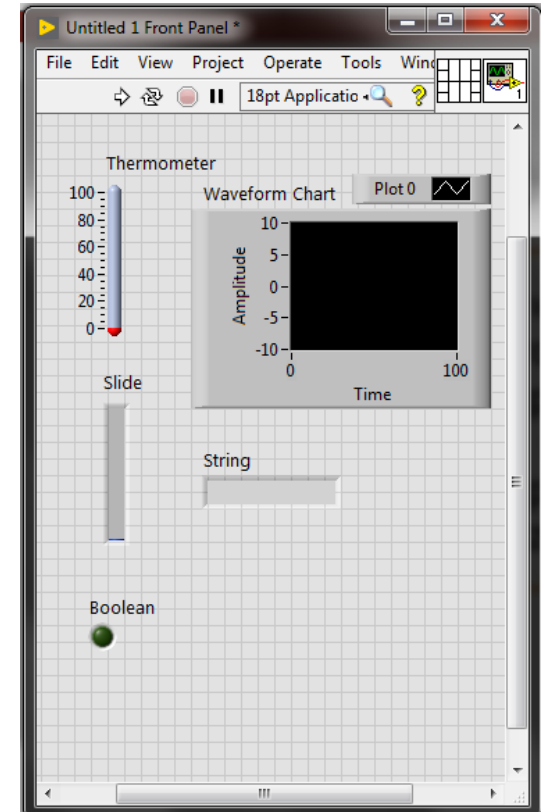
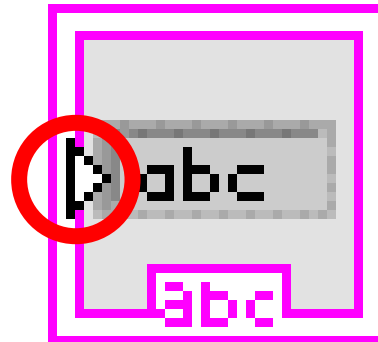
Indykatory



Dane wyjściowe z programu:

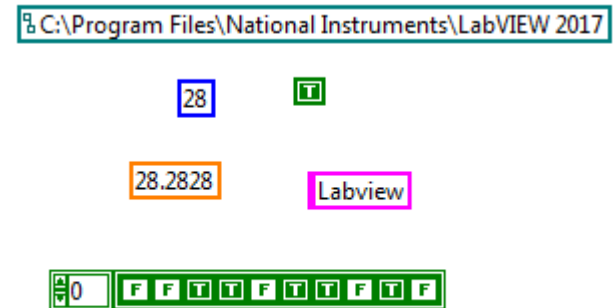
- paski progresu,
- diody,
- wykresy,
- tekst...

String



Stałe

- Dostarczają danych wejściowych, tak jak kontrolki.
- Są niezmiennie podczas działania programu.
- Użytkownik nie może ich zmieniać z poziomu Front panelu.



Wires - przewody

	Floating-point	Integer	String	Boolean
Scalar				
1-D Array				
2-D Array				

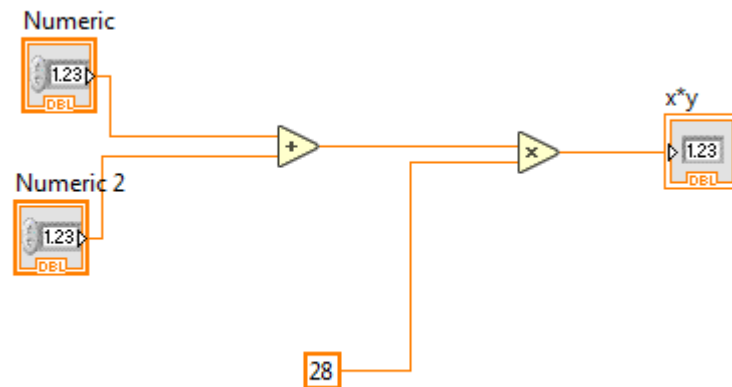
Przewód błędnie połączony
oznaczony jest czerwonym X



Ctrl+B



Przepływ danych



Funkcja wykonuje się dopiero, gdy dostępne są wszystkie dane wejściowe.

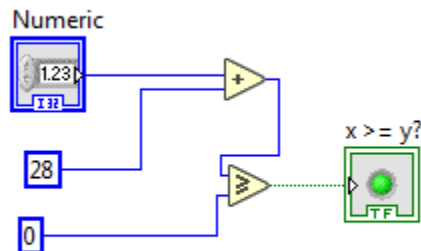
Funkcja przekazuje dane dalej, dopiero po wykonaniu funkcji.



Pytanie 1

Wskaż kontrolki, indykatory, stałe i funkcje.

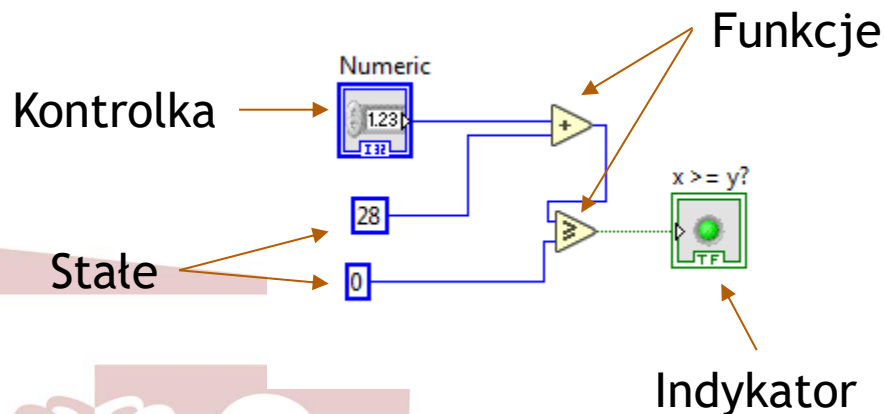
Która funkcja wykona się jako pierwsza?



Odpowiedź

Wskaż kontrolki, indykatory, stałe i funkcje.

Która funkcja wykona się jako pierwsza?

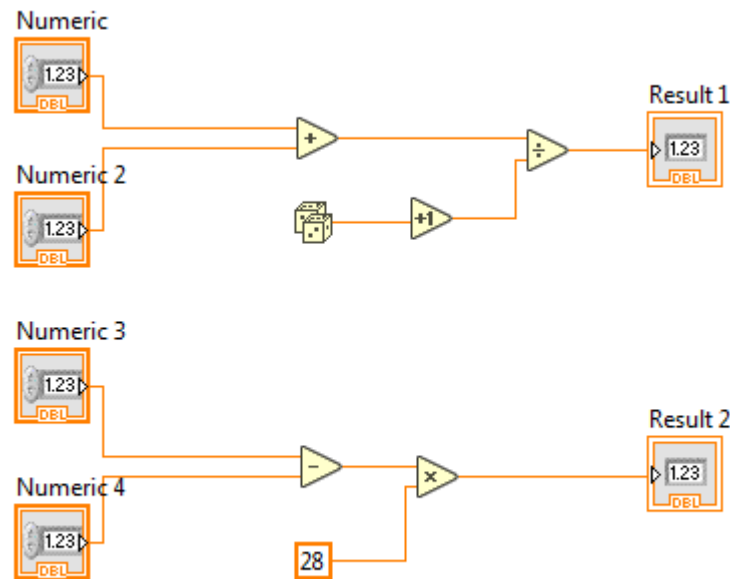


Dodawanie wykona się jako pierwsze.



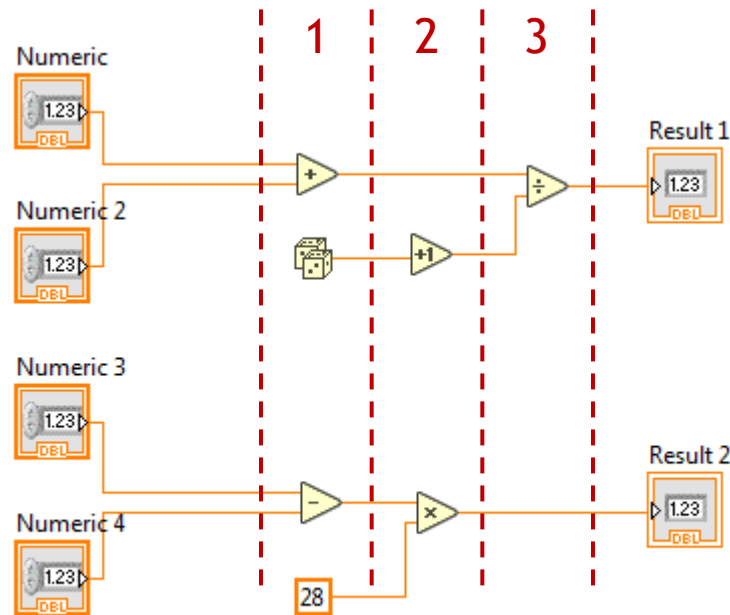
Pytanie 2

- Która funkcja wykona się jako pierwsza?



Odpowiedź

- Która funkcja wykona się jako pierwsza?



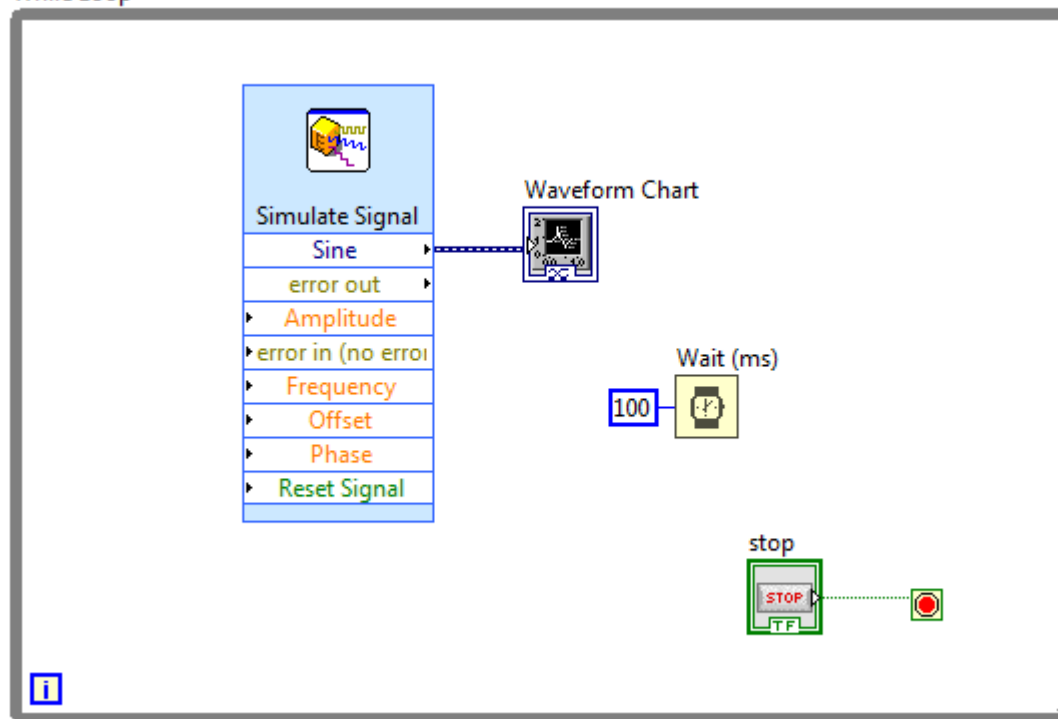
Zadanie 1: Mój pierwszy program

Wykorzystując funkcję *Simulate signal* zasymuluj sygnał sinusoidalny o częstotliwości 0.8 Hz i amplitudzie 28. Wykorzystaj *Waveform chart* do wizualizacji danych. Całość umieść w pętli *While* podłączając przycisk *Stop* jako warunek kończący pętlę. Wykorzystaj funkcję *Wait* (100 ms) do spowolnienia programu.



Rozwiązanie

While Loop



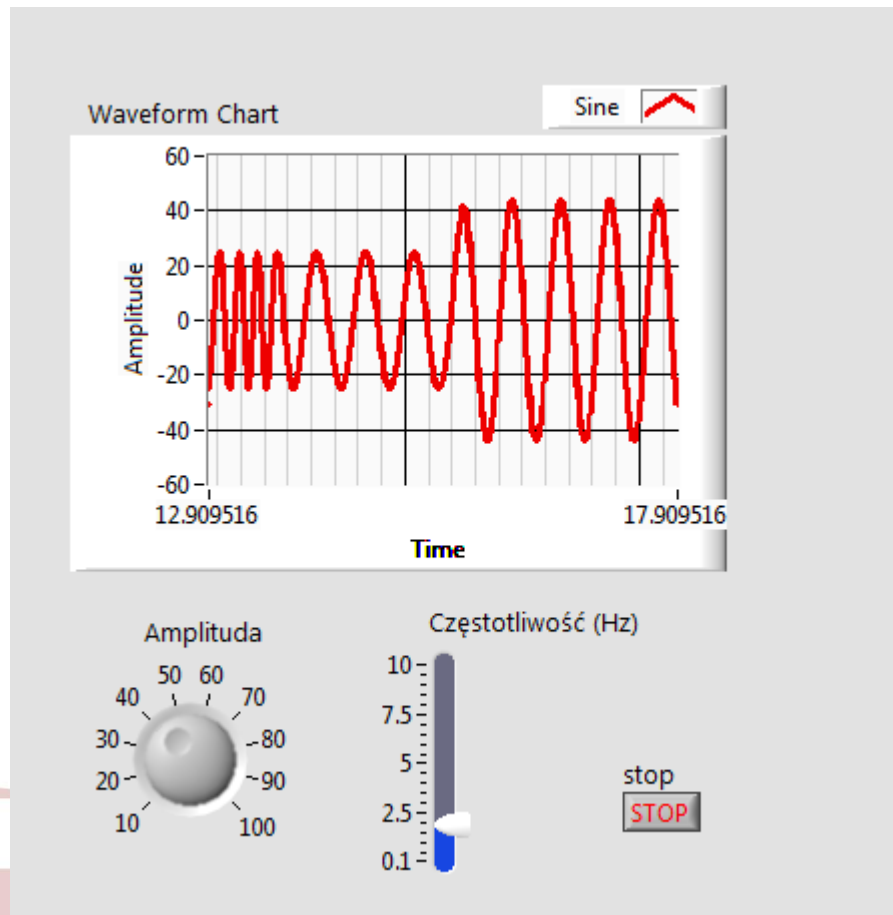
Zadanie 2

Zmodyfikuj swój program, w taki sposób aby użytkownik mógł zmieniać amplitudę sinusoidy (w granicach 10-100) oraz częstotliwość (w granicach 0.1-10 Hz).

Podpowiedź: wykorzystaj odpowiednie kontrolki i ich opcje.



Rozwiązanie 2



Zadania

3. Używając *Tool Palette* zmodyfikuj wygląd *Waveform chart*.
4. Stwórz drugi generator fali sinusoidalnej. Wykorzystaj funkcje *Merge signals* oraz *Add*. Jaka jest różnica?
5. Wykorzystując odpowiedni *Express Vi* oblicz pochodną po czasie sygnału sinusoidalnego i przedstaw wyniki na jednym wykresie.

