

Wykład 1
01.03.2024 r.
Wprowadzenie, wstęp
teoretyczny dotyczący obsługi
LabVIEW
Adrian Kaim



HR EXCELLENCE IN RESEARCH



Politechnika Wroclawska

O prowadzącym

- mgr inż. Adrian Kaim
- Katedra Fizyki Doświadczanej
Wydział Podstawowych Problemów Techniki
- p. 434 bud. A1
- Konsultacje: Piątki 13¹⁵-14¹⁵
- Email: adrian.kaim@pwr.edu.pl
- Strona: kaim.wppt.pwr.edu.pl



Adrian Kaim – doktorant Szkoły Doktorskiej Politechniki Wrocławskiej i członek Laboratorium Badania Półprzewodników Metodami Elektrycznymi w Katedrze Fizyki Doświadczalnej na Wydziale Podstawowych Problemów Techniki. W ramach doktoratu prowadzi badania związane z fotodetektorami opartymi m. in. na efekcie pirofototronicznym. Jest posiadaczem certyfikatu CLAD

Plan zajęć

- | | |
|--|--|
| 1. Wprowadzenie (2h) | 8. Obsługa joysticka (2h) |
| 2 i 3. Funkcje i struktury (4h) | 9. Komunikacja z komputerem (2h) |
| 4. Dokumentacja i wizualizacja danych (2h) | 10. Wyświetlacz 7-segmentowy (2h) |
| 5. Macierze, tablice, klastry (2h) | 11. Potencjometr, przełącznik DIP (2h) |
| 6. Funkcje graficzne (2h) | 12. Silniczek servo (2h) |
| 7. Struktura event (2h) | 13. Wstęp do projektu (2h) |
| | 14–15. Projekt, zaliczenie (4h) |

Zaliczenie

Na podstawie:

- **Projektu** – ostatnie trzy zajęcia + praca poza zajęciami
- Aktywności
- Pracy na zajęciach
- Obecności

4



Politechnika Wrocławska

Kurs podlega zaliczeniu w trakcie semestru. Główną składową oceny końcowej jest ocena z projektu zaliczeniowego przygotowywanego na trzech ostatnich zajęciach (oraz poza zajęciami). Dodatkowo na ocenę końcową składają się:

- aktywność w trakcie zajęć,
- praca na zajęciach (wykonywanie zadań),
- frekwencja.

Warunki konieczne do uzyskania pozytywnej oceny końcowej:

- obecność na przynajmniej 13 z 15 zajęć (nie dotyczy sytuacji nadzwyczajnych, w których nieobecności zostaną usprawiedliwione),
- złożenie projektu zaliczeniowego, ocenionego pozytywnie przez prowadzącego.

Warunki konieczne do uzyskania celującej oceny końcowej:

- obecność na przynajmniej 13 z 15 zajęć (nie dotyczy sytuacji nadzwyczajnych, w których nieobecności zostaną usprawiedliwione),
- złożenie projektu zaliczeniowego, ocenionego *bardzo dobrze* lub *celująco* przez prowadzącego,
- wykazywanie aktywności na zajęciach,

- wykonanie ponad połowy przewidzianych zadań.

Projektem zaliczeniowym jest program przygotowany w języku graficznym G (LabVIEW). Program oceniany jest pod kątem samodzielności, czytelności kodu oraz interfejsu, udokumentowania kodu, efektywności działania kodu, zaawansowania struktur wykorzystanych w programie oraz obsługi błędów (programistycznych oraz użytkownika).

Materiały

- Dokumentacja LabVIEW → www.ni.com/getting-started/LabVIEW-basics/online-help
- Wykłady dostępne na stronie prowadzącego
- „LabVIEW w praktyce” Marcin Chruściel, Wydawnictwo BTC, 2008.
- „Środowisko LabVIEW™ w eksperymencie wspomaganym komputerowo” Wiesław Tłaczała, Wydawnictwo WNT, 2014.

LabVIEW dla studentów

Opis instalacji oraz klucz licencyjny dostępne na stronie:

di.pwr.edu.pl/oprogramowanie/LabVIEW

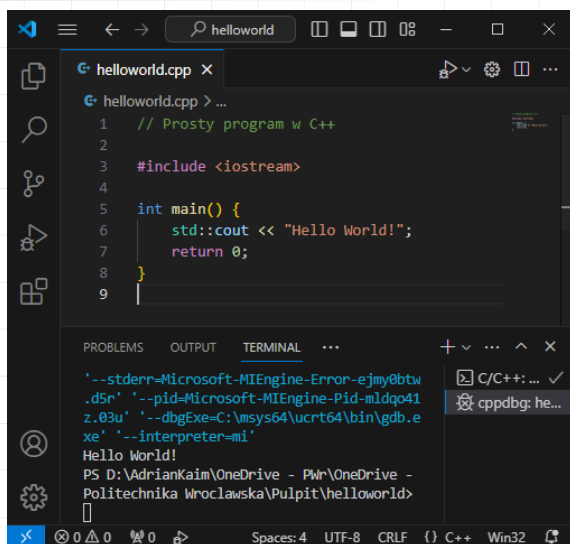
Studenci Politechniki Wrocławskiej mają dostęp do licencjonowanego pakietu LabVIEW. Wszystkie informacje, pliki instalacyjne oraz klucze licencyjne dostępne są na stronie Działu Informatyzacji.

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.”

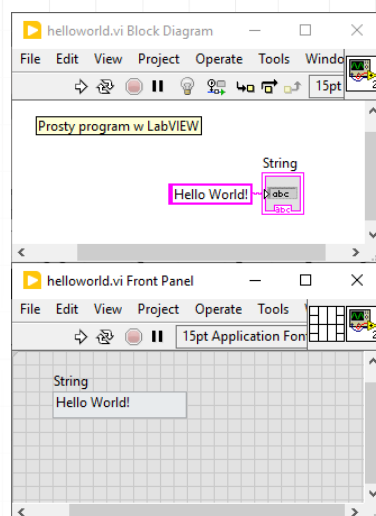
Wikipedia



```
helloworld.cpp X
helloworld.cpp > ...
1 // Prosty program w C++
2
3 #include <iostream>
4
5 int main() {
6     std::cout << "Hello World!";
7     return 0;
8 }
9

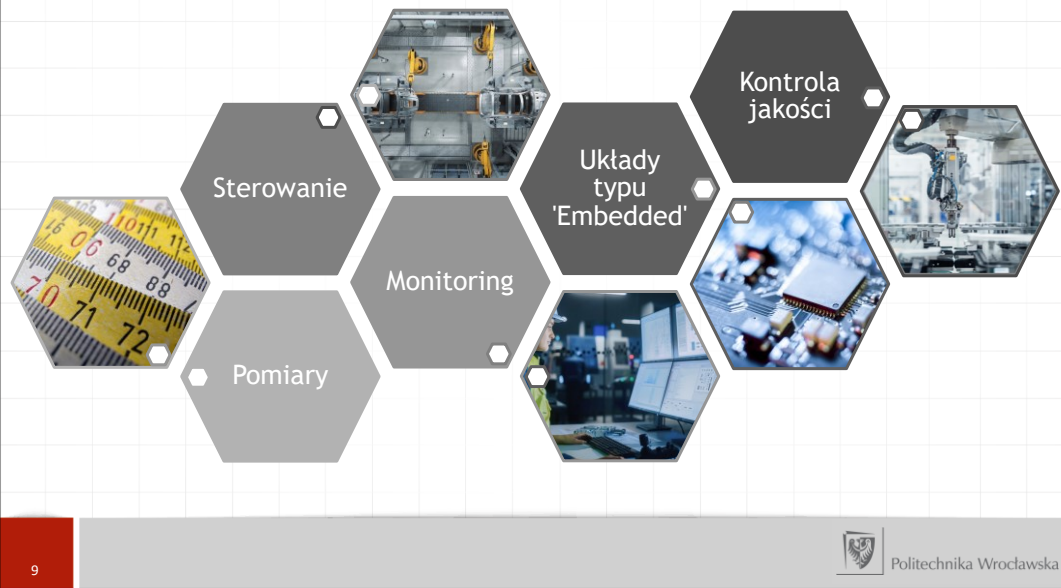
PROBLEMS OUTPUT TERMINAL ...
'--stderr=Microsoft-MIEngine-Error-ejmy0btw
.d5r' '--pid=Microsoft-MIEngine-Pid-mldqo41
z.03u' '--dbgExe=C:\msys64\ucrt64\bin\gdb.e
xe' '--interpreter=mi'
Hello World!
PS D:\AdrianKaim\OneDrive - PWr\OneDrive -
Politechnika Wroclawska\Pulpit\helloworld>
```

Prosty program w języku tekstowym C++



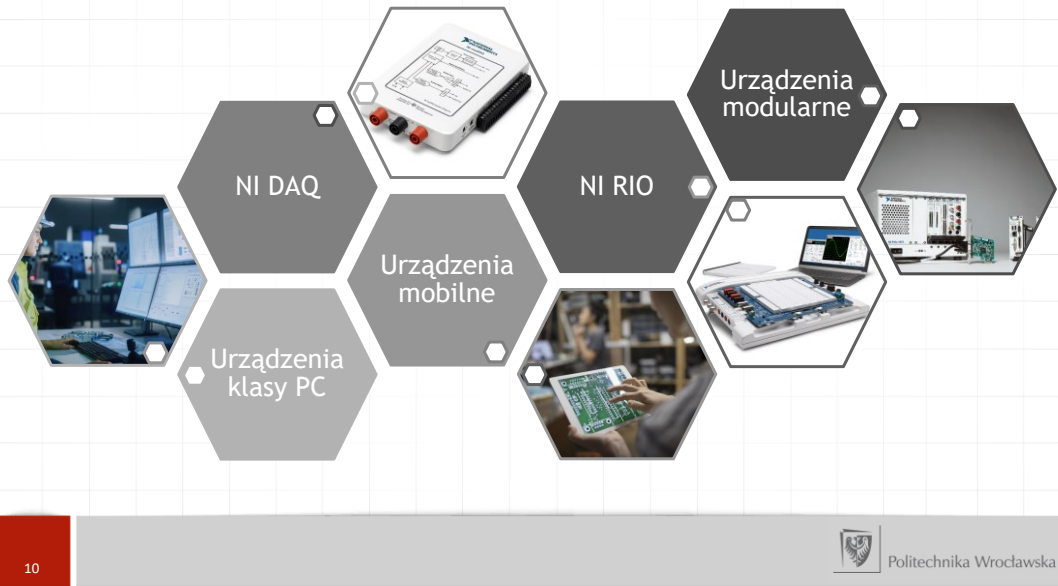
Prosty program w języku graficznym LabVIEW

Zastosowania LabVIEW



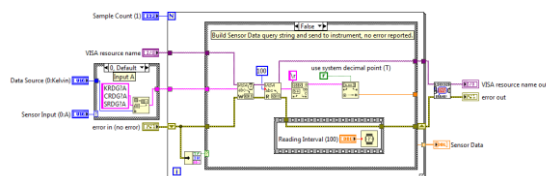
Korzystając z LabVIEW można tworzyć programy, których zadaniami mogą być: zbieranie danych pomiarowych, sterowanie procesami, sterowanie układami typu „Embedded”, monitorowanie procesów, czy kontrola jakości. Głównym celem tworzenia oprogramowania do tych zastosowań, jest częściowe lub całkowite zautomatyzowanie tych czynności.

Platformy docelowe LabVIEW



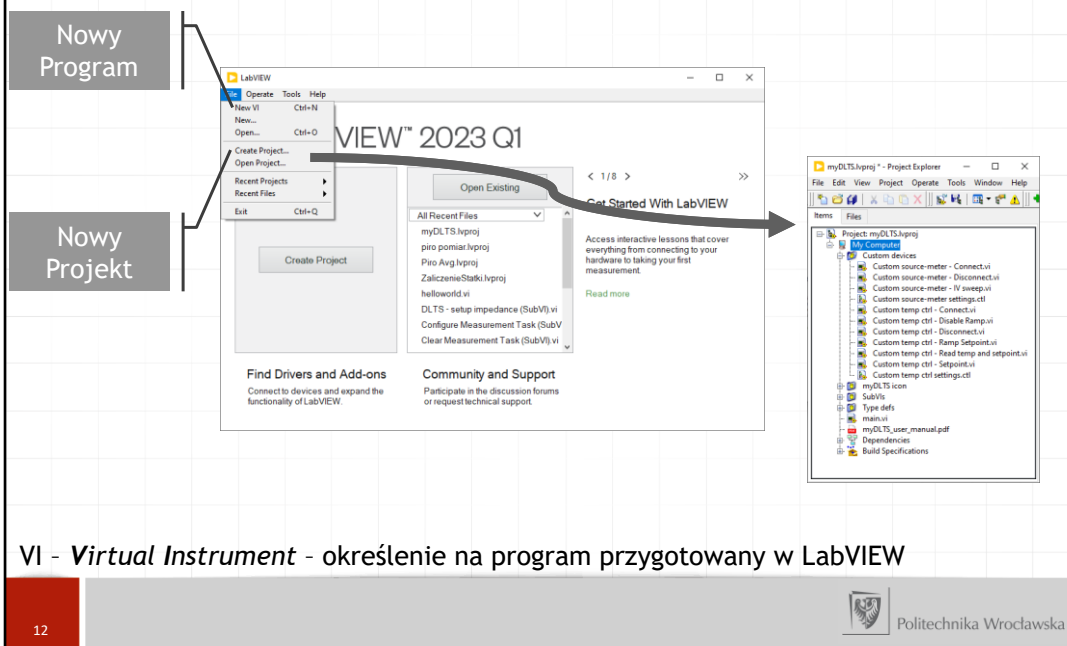
Programy tworzone w LabVIEW mogą być przeznaczone na urządzenia klasy PC (oparte o system MS Windows, Apple OSX oraz Linux), urządzenia mobilne oraz urządzenia typu „Embedded” produkowane przez National Instruments. Program może wykorzystywać wiele z tych urządzeń, komunikując się przy pomocy różnych interfejsów (np. USB, Ethernet, RS-232, GPIB itp.)

Zastosowania LabVIEW w LEPS (K63)



LabVIEW znalazło wiele zastosowań w Laboratorium Charakteryzacji Półprzewodników Metodami Elektrycznymi (Katedra Fizyki Doświadczalnej). Wykorzystywane jest do sterowania urządzeniami takimi jak np.: kontroler temperatury Lake Shore 325, analizator impedancji Zurich Instruments MFIA, czy kamera CCD Symphony II. Urządzenia te, odpowiednio oprogramowane, dają dostęp do technik takich jak niestacjonarna spektroskopia pojemnościowa, czy pomiary widm fotoluminescencji.

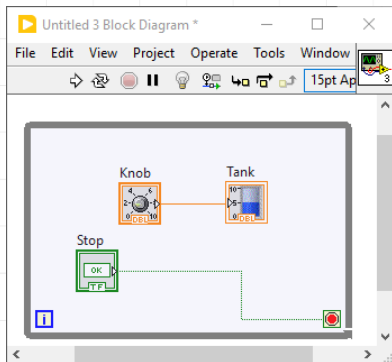
LabVIEW – okno startowe



VI - *Virtual Instrument* - określenie na program przygotowany w LabVIEW

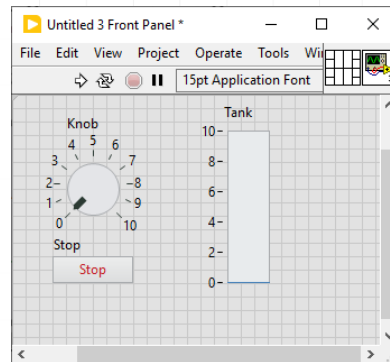
Programy w LabVIEW noszą miano VI (ang. *Virtual Instrument*), czyli wirtualnych przyrządów (pomiarowych, sterujących itp.). Nowy program można stworzyć w oknie startowym klikając File>New VI (lub CTRL+N). Programy tworzone w LabVIEW zwykle organizowane są w postaci projektów. Nowy projekt można stworzyć w oknie startowym klikając File>Create Project.

Block diagram



Funkcje, przewody, subvi, struktury...

Front panel

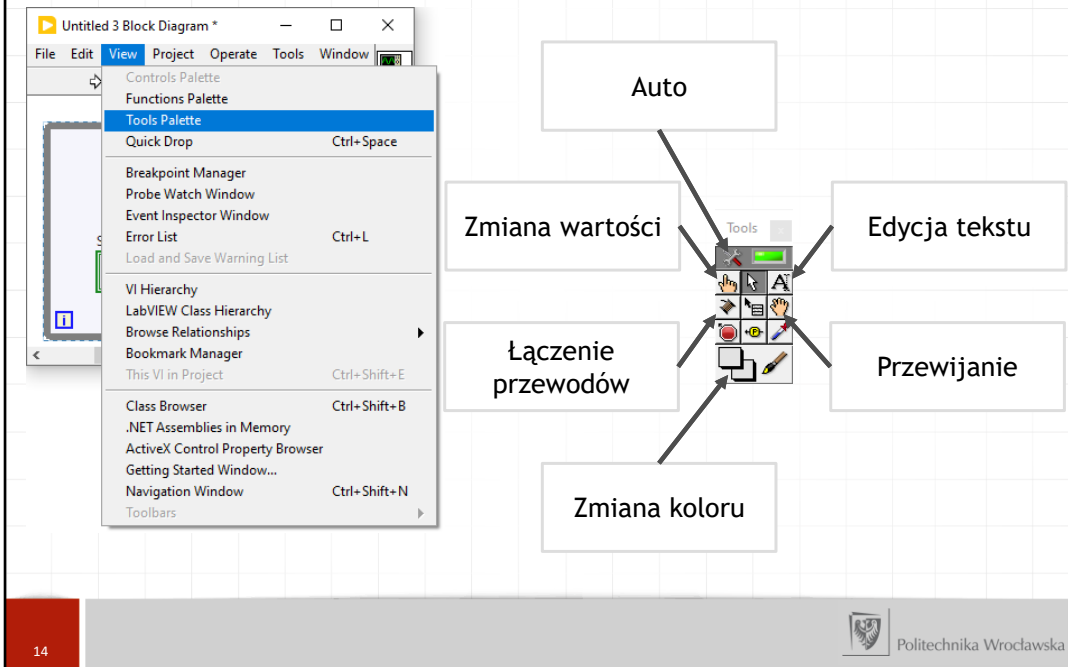


Interfejs użytkownika: kontrolki, indykatory, opisy ...

CTRL+E

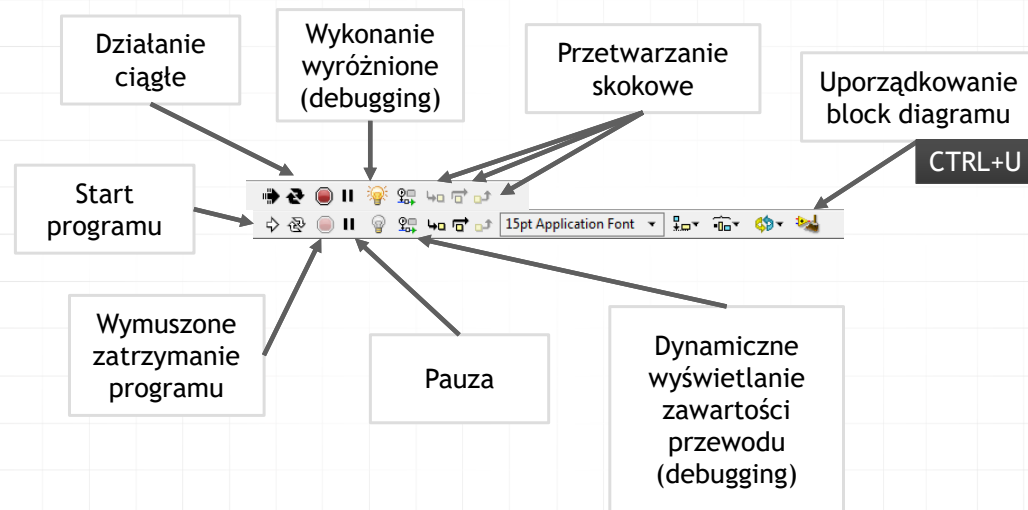
Każdy program w LabVIEW posiada dwie części: Block Diagram oraz Front Panel. Block Diagram jest miejscem w którym definiowane są wszystkie czynności które zostaną wykonane: funkcje, SubVI, struktury, łączące je przewody itp.. Front panel stanowi swoisty interfejs użytkownika i służy do komunikacji z użytkownikiem (oraz innymi programami). Front panel składa się z kontroltek, indykatorów, opisów, elementów dekoracyjnych itp.. Skrót CTRL+E umożliwia szybkie przełączanie między tymi oknami.

Tools Palette – paleta funkcji

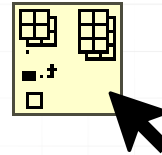
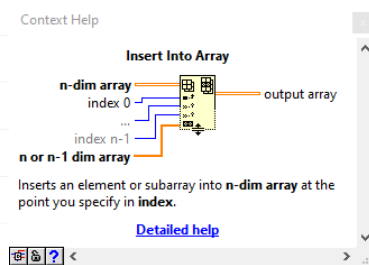


W trakcie tworzenia programu pomocna może być tzw. paleta funkcji. Można ją wyświetlić, klikając View>Tools Palette. Paleta ta zawiera zestaw przycisków przełączających między funkcjami kursora myszy.

Pasek statusu



Context Help – pomoc kontekstowa

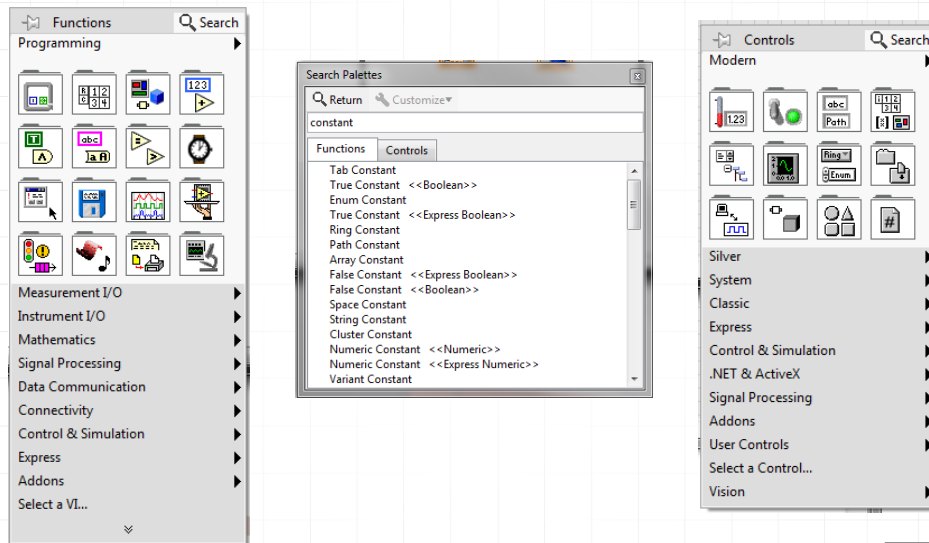


CTRL+H



Okno pomocy kontekstowej można otworzyć klikając Help>Show Context Help, Klikając okno pytajnika, lub CTRL+H. Po najechaniu kursorem na ikonę funkcji, kontrolki, indykatora itp., wyświetla podstawowe informacje o tych elementach.

Paleta funkcji i kontrolek

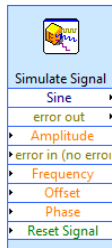


PPM

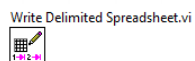


Klikając prawym przyciskiem myszy na block diagram (lub front panel), uzyskuje się dostęp do palety funkcji (lub kontrolek). Palety te zawierają wszystkie wbudowane w środowisko elementy pozwalające budować programy. Możliwe jest tekstowe wyszukiwanie elementów.

Nodes – węzły



Express Vi
konfigurowalne w osobnym okienku

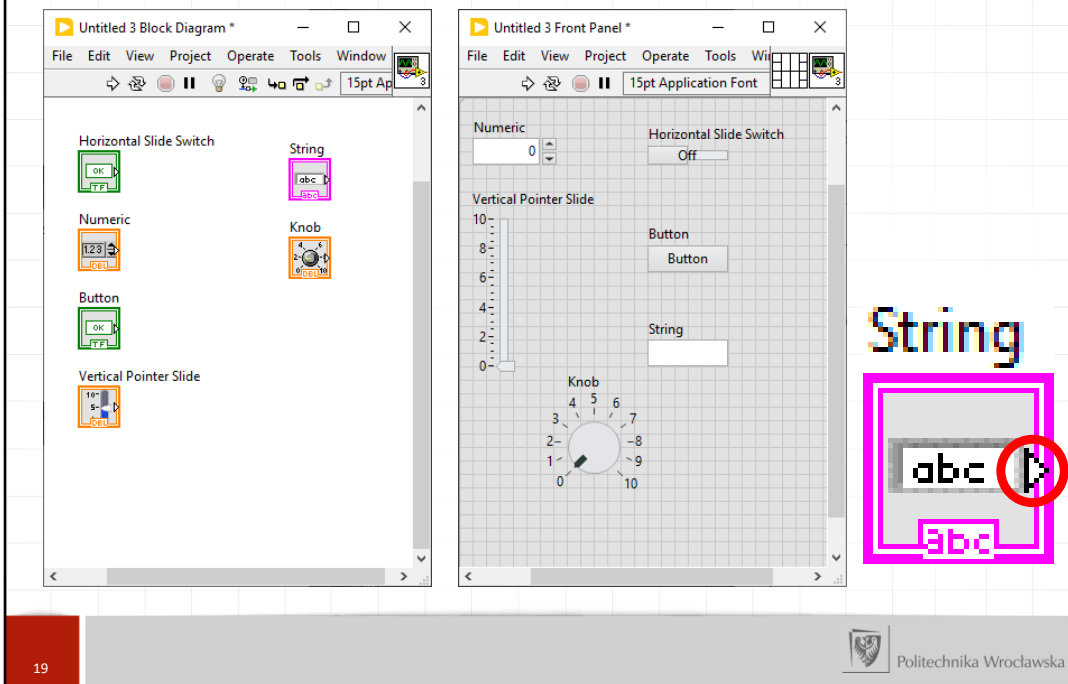


Standard Vi
ograniczona konfiguracja,
zależna od danych wejściowych



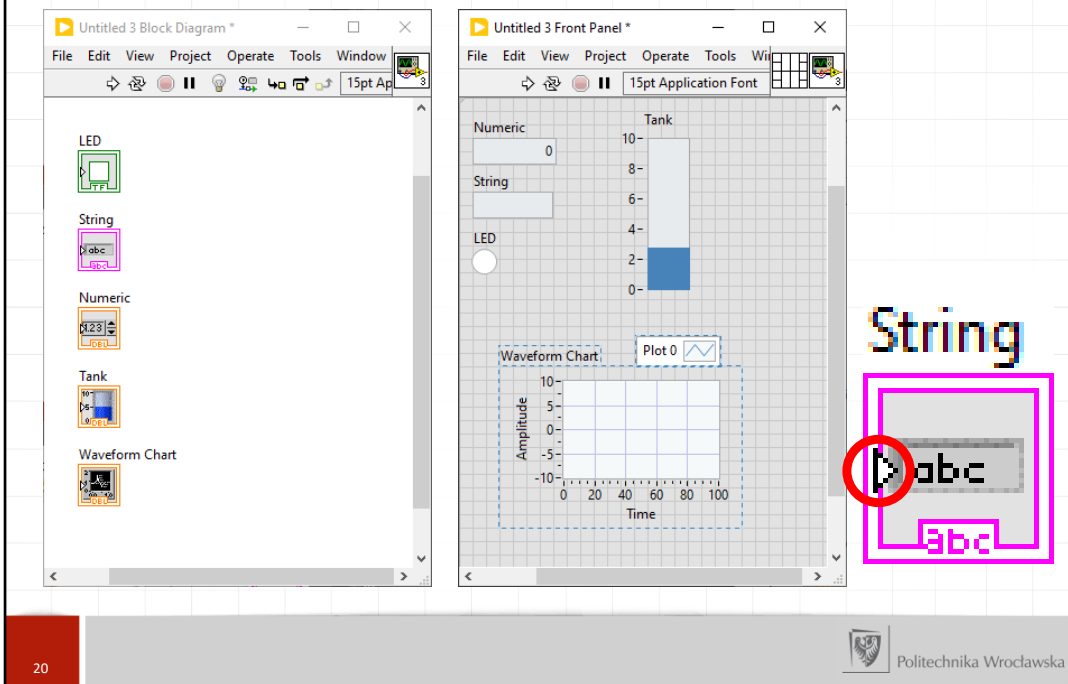
Proste funkcje
niekonfigurowalne

Controls – elementy sterujące



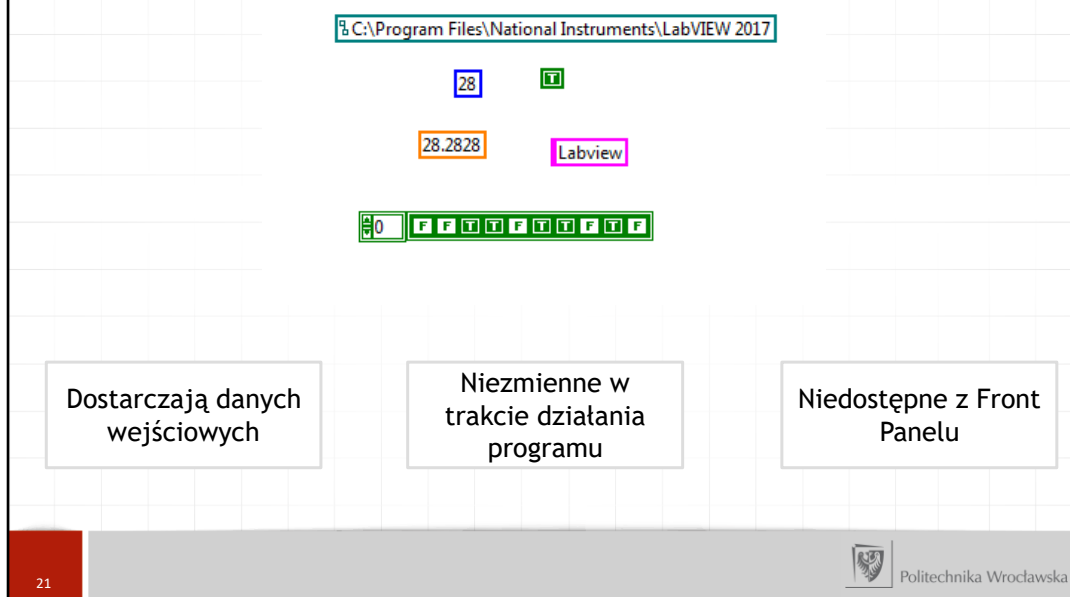
„Kontrolki” to elementy Front Panelu odpowiedzialne za wprowadzanie danych do programu. Przykłady kontrolki to m.in.: pokrętki, suwaki, przyciski, przełączniki. Na Block Diagramie ikona kontrolki oznaczona jest charakterystycznym trójkątem, sygnalizującym kierunek danych wychodzących z elementu.

Indicators – wskaźniki



„Indykatory” to elementy Front Panelu odpowiedzialne za wyświetlanie danych wyjściowych programu. Przykłady indykatorów to m.in.: paski stanu, diody, wykresy, tekst. Na Block Diagramie ikona indykatora oznaczona jest charakterystycznym trójkątem, sygnalizującym kierunek danych wchodzących do elementu.

Stałe



Jako źródło danych wejściowych w programie wykorzystane mogą zostać także stałe. Wartość stałych pozostaje niezmienna w trakcie działania programu, a użytkownik nie ma do nich dostępu z poziomu Front Panelu.

Wires – przewody

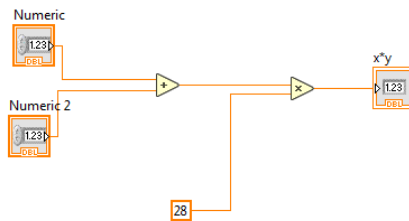
	Floating-point	Integer	String	Boolean
Scalar				
1-D Array				
2-D Array				

Przewód błędnie połączony
oznaczony jest czerwonym X



Do przesyłania danych pomiędzy różnymi elementami programu wykorzystuje się przewody łączące ich terminale. Przewody przyjmują różne kolory i grubości w zależności od typu danych jaki przenoszą. W przypadku błędnego połączenia przewodu (np. połączenie terminali różnych typów), przewód oznaczany jest czerwonym X. Takie błędnie połączone przewody można szybko usunąć skrótem CTRL+H.

Przepływ danych



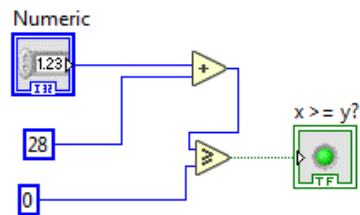
Funkcja wykonuje się dopiero, gdy dostępne są wszystkie dane wejściowe.

Funkcja przekazuje dane dalej, dopiero po wykonaniu funkcji.

Pytanie 1

Wskaż kontrolki, indykatory, stałe i funkcje.

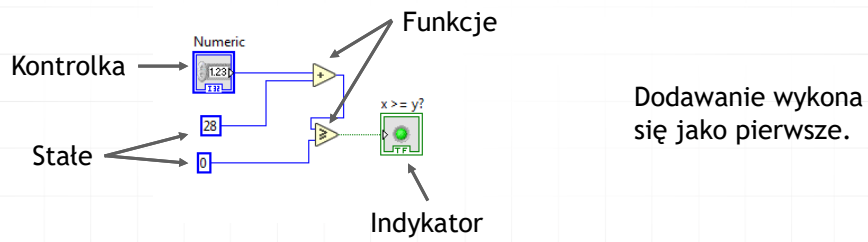
Która funkcja wykona się jako pierwsza?



Pytanie 1

Wskaż kontrolki, indykatory, stałe i funkcje.

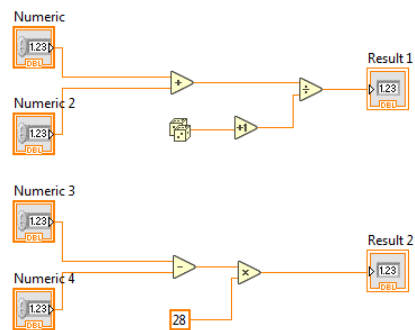
Która funkcja wykona się jako pierwsza?



Dodawanie wykona się jako pierwsze.

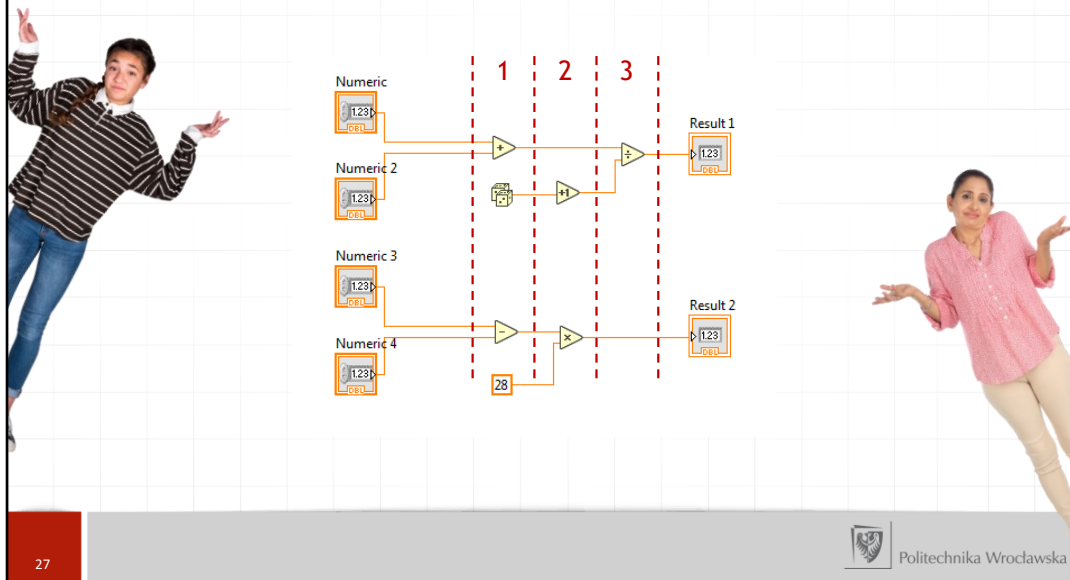
Pytanie 2

Która funkcja wykona się jako pierwsza?



Pytanie 2

Która funkcja wykona się jako pierwsza?

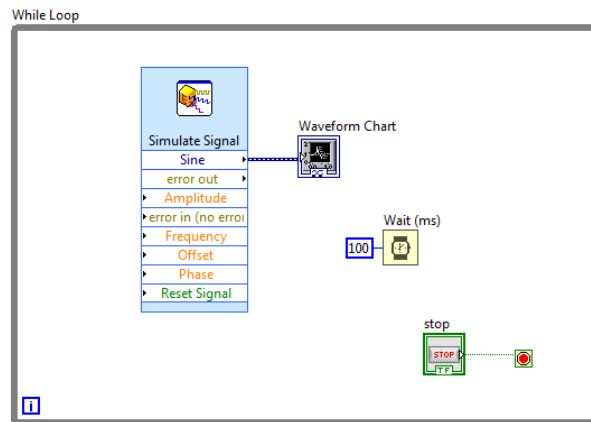


Zadanie 1: Mój pierwszy program

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.

Zadanie 1: Mój pierwszy program

Przykładowe rozwiązanie:



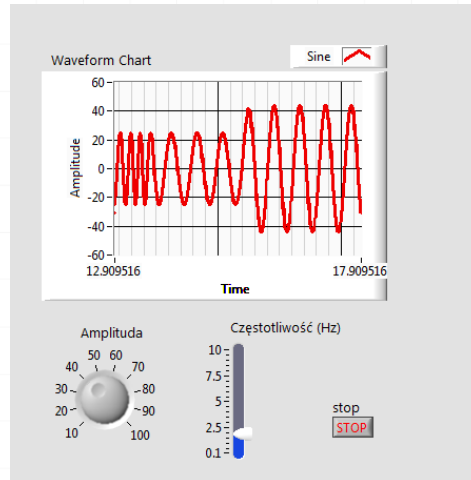
Zadanie 2

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.

Zadanie 2

Przykładowe rozwiązanie:



Zadania

3. Używając *Tools 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.

Zadanie domowe

Zapoznaj się z funkcjami, które dostępne są w palecie funkcji w kategorii Programming (Structures, Array, Numeric, Boolean, String, Comparison).

Zwróć uwagę na ogólne przeznaczenie tych funkcji oraz na to w jaki sposób podzielono je na podkategorie. Skorzystaj z pomocy kontekstowej dostępnej w środowisku LabVIEW lub z dokumentacji [On-Line](#).