

Politechnika Wrocławska, Katedra Inżynierii Biomedycznej

Systemy Pomiarowo-Diagnostyczne, laboratorium

Zajęcia wprowadzające

1. Cel ćwiczenia

Przyswojenie podstawowych informacji dotyczących zasad tworzenia wirtualnych urządzeń z wykorzystaniem języków graficznych na przykładzie NI LabView. Zapoznanie się z najczęściej wykorzystywanymi elementami, metodami i strukturami wykorzystywanego na zajęciach środowiska programistycznego.

2. Zakres wprowadzenia

Na zajęciach zostaną omówione przez prowadzącego i praktycznie zaprezentowane następujące elementy oraz cechy środowiska programistycznego LabView:

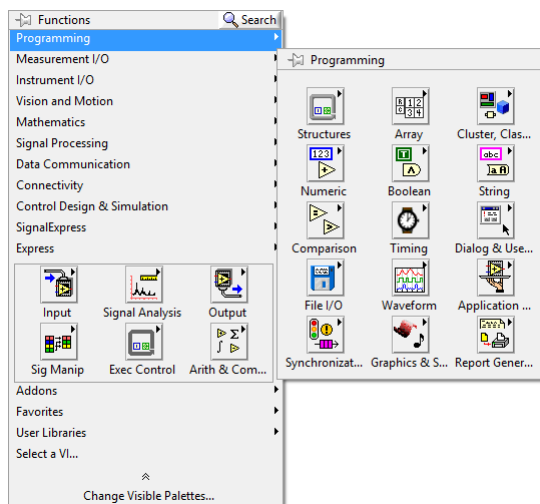
- podstawowe właściwości języka graficznego G oraz zasady przepływu danych (data flow) w programie
- zasady tworzenia urządzeń wirtualnych (vi), praca z panelem i diagramem
- zapisywanie programów, przenoszenie między różnymi wersjami środowiska
- metody analizy programu i poszukiwania błędów
 - analiza przepływu danych w diagramie (data flow)
 - wykorzystanie sondy danych (probe)
 - praca krokowa



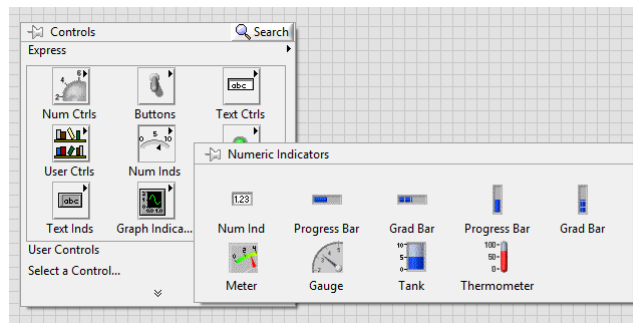
- palety dostępne w środowisku
 - paleta narzędzi (tryby pracy z diagramem i panelem)



- paleta diagramu (funkcje)

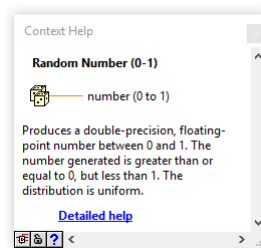


c. paleta panelu (kontrolki i wskaźniki)

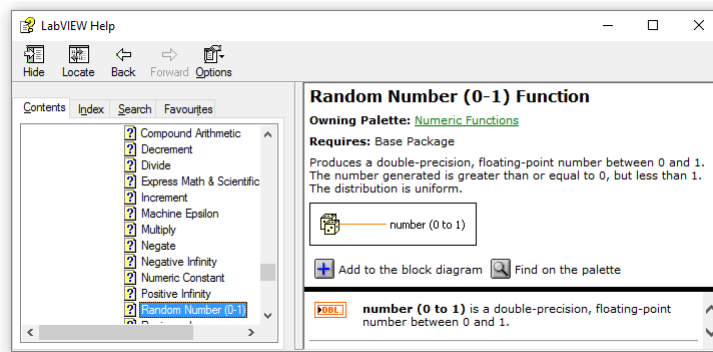


f) dostępne rodzaje pomocy w środowisku programistycznym w tym:

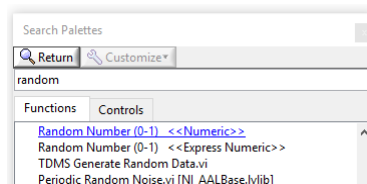
a. pomoc kontekstowa



b. pomoc główna



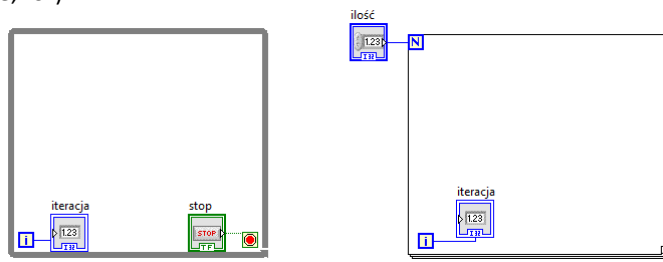
c. wyszukiwanie funkcji



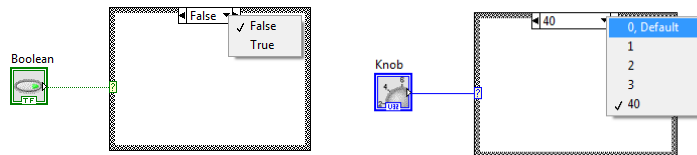
g) typy oraz podtypy danych (integer, double, string, boolean)

Wire Type	Scalar	1D Array	2D Array	Color
Numeric				Orange (floating-point), Blue (integer)
Boolean				Green
String				Pink

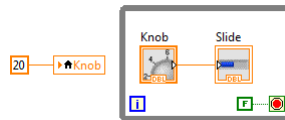
- h) podstawowe struktury
 a. pętle (while, for)



- b. warunki/rozgałęzienia (case/switch)



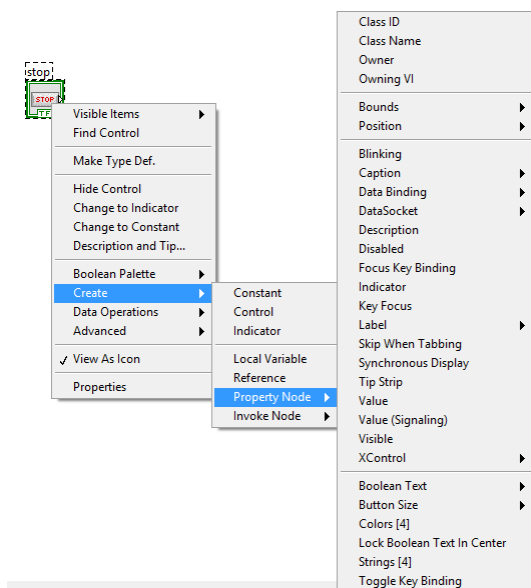
- i) zmienne lokalne



- j) grupowanie danych jednego oraz różnych typów (array, cluster)



- k) zmiana właściwości i parametrów kontrolki i wskaźników (property node)



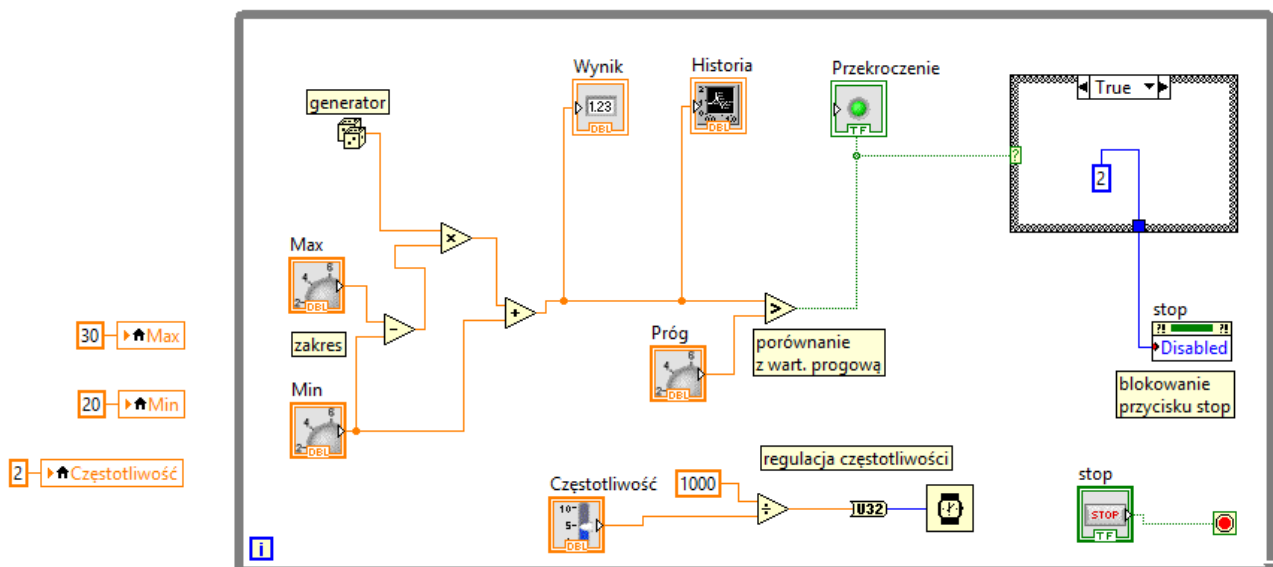
3. Zadania do wykonania

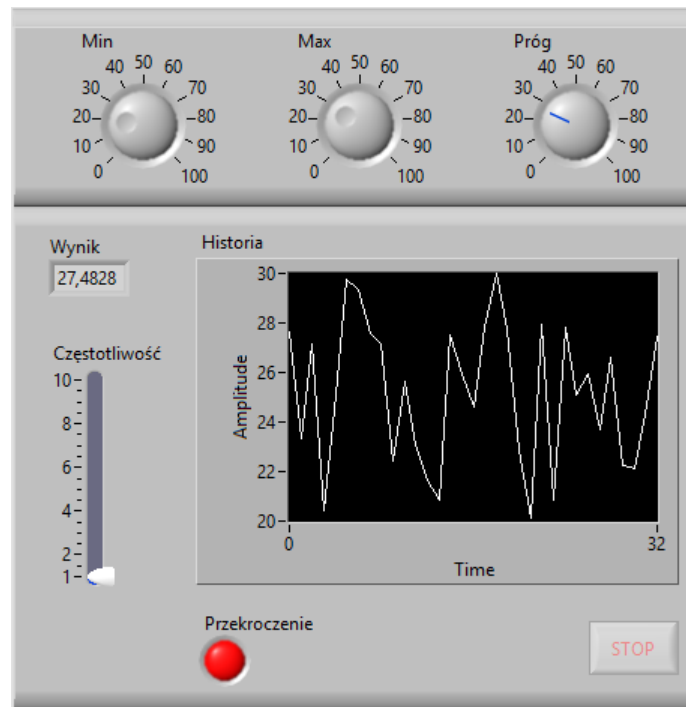
Podążając za wskazówkami prowadzącego należy zrealizować pierwszy program, w którym zostaną wykorzystane opisane powyżej funkcje języka graficznego i środowiska LabView.

Przykładowe założenia i funkcje do zrealizowania (należy realizować etapami i sprawdzać poprawność działania po wykonaniu każdego z podpunktów):

- a) Program symuluje pozyskiwanie serii wyników pomiarowych poprzez wykorzystanie generatora liczb losowych (*random numer*). Uzyskany wynik jest prezentowany na panelu w postaci liczby.
- b) Program jest realizowany w pętli zatrzymywanej przyciskiem STOP.
- c) Częstotliwość generowania wyników jest regulowana na panelu operatora w zakresie 1-10Hz
- d) Historia wartości wyników jest prezentowana na wykresie.
- e) Operator ma udostępnione dwa regulatory: wartości minimalnej i maksymalnej. Wyniki z generatora przeliczane są w taki sposób by wartość końcowa mieściła się w zakresie tych dwóch wartości.
- f) Regulatory wartości minimalnej i maksymalnej mają wymuszony zakres pracy od 0 do 100 z rastrem 1.
- g) Korzystając z funkcji porównania (mniejsze lub większe) program sygnalizuje przekroczenie zadanej przez operatora wartości progowej wyniku. Sygnalizacja odbywa się poprzez zapalenie czerwonej kontrolki (np. typu LED).
- h) W przypadku przekroczenia zadanej wartości (kontrolka jest zapalona) nie jest możliwe zatrzymanie działania głównej pętli programu poprzez naciśnięcie przycisku STOP na panelu (wykorzystać *property node – disabled*)
- i) Podczas uruchomienia programu nastawione są początkowe wartości regulatorów (np. częstotliwość 2Hz, odpowiedź: można wykorzystać zmienne lokalne)
- j) *Zadanie dodatkowe: Jak zapewnić by nastawa wartości minimalnej nie mogła być ustawiona powyżej wartości maksymalnej w trakcie ciągłego działania programu (podpowiedź: można wykorzystać właściwości kontrolki *property node* lub zmienne lokalne)

Przykładowa realizacja zadania wygląda następująco:





4. Zadania dodatkowe

Propozycja zadań uzupełniających w ramach pozostałego czasu na zajęciach lub do pracy indywidualnej poza zajęciami:

- a) Zrealizować program wyświetlający historię dwóch różnych nastaw lub wyników na jednym wykresie, tak aby było możliwe ich łatwe porównanie. Na przykład można jednocześnie wyświetlać wartość wygenerowaną w zadaniu z punktu 3 z wartością nastawioną na regulatorze progu porównania. Przy takiej prezentacji wyników łatwo jest określić kiedy uzyskane wyniki przekroczyły zadany próg. Do realizacji zadania wykorzystać element *bundle* umożliwiający stworzenie pary wartości w postaci klastra.
- b) Zrealizować program symulujący wykorzystanie czujników dwustanowych (do tego celu można wykorzystać elementy kontrolne w postaci przycisków). Na bazie stanów przycisków podejmować w programie działania z wykorzystaniem funkcji logicznych (*AND*, *OR*, *NOT*) wg. scenariusza zaproponowanego przez prowadzącego.