

INSTYTUT TELEKOMUNIKACJI MULTIMEDIALNEJ

LABORATORIUM KOMPUTEROWYCH SYSTEMÓW POMIAROWYCH

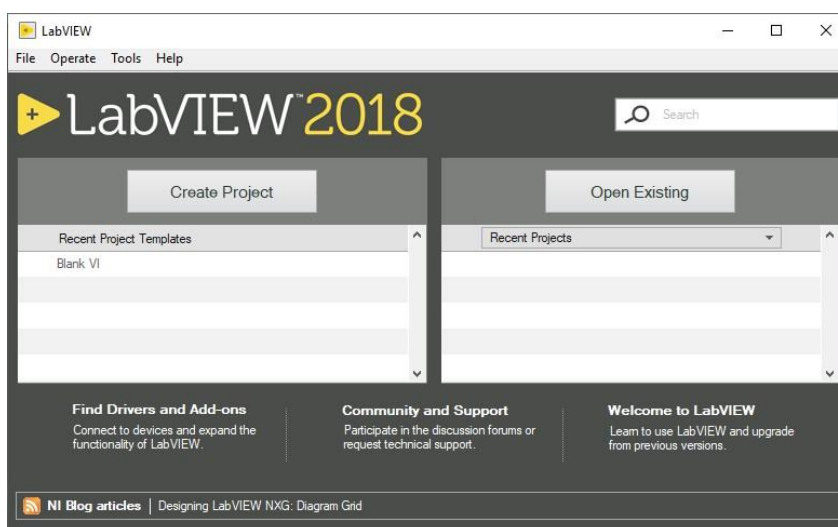
Instrukcja do ćwiczenia:

Wprowadzenie do programowania w LabVIEW

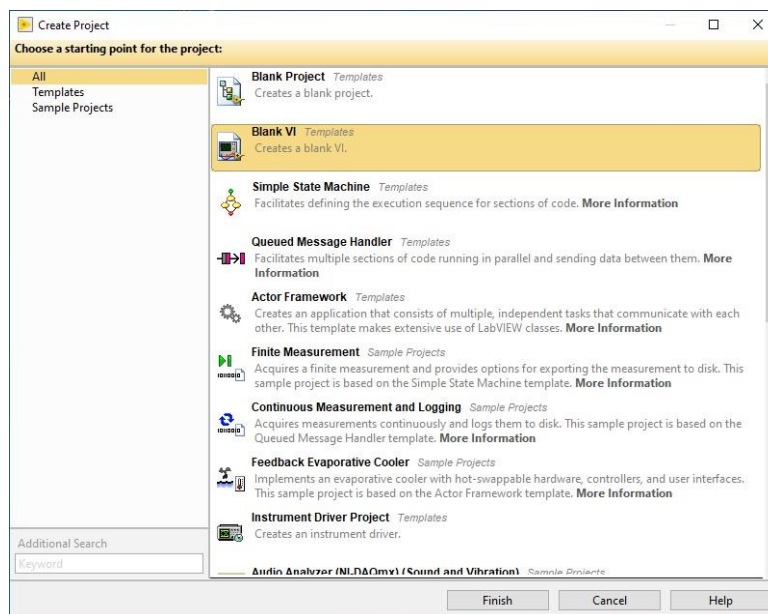
1. Wprowadzenie

LabVIEW jest środowiskiem programowania systemów pomiarowych-sterujących opracowanym przez firmę National Instruments. Tradycyjny język tekstowy, w którym polecenia umieszczane są w kolejnych wierszach programu, został zastąpiony przez zintegrowane środowisko graficzne. Program tworzony jest w postaci schematu (diagramu), na którym poszczególne instrukcje przedstawiane są jako symbole (obiekty) graficzne. Połączenia pomiędzy obiektami obrazują przepływ sygnałów (danych w programie).

Po uruchomieniu programu LabVIEW 2018 pojawia się okno przedstawione na rysunku 1, w którym możemy utworzyć nowy projekt (*Create Project*) lub otworzyć wcześniej utworzony (*Open Existing*). Po naciśnięciu przycisku *Create Project* ukazuje się kolejne okno przedstawione na rysunku 2, w którym w celu utworzenia pliku VI wybieramy *Blank VI* lub w celu utworzenia zaawansowanego projektu LabVIEW wybieramy *Blank Project*. W trakcie niniejszego ćwiczenia laboratoryjnego będzie wykorzystywana pierwsza metoda tworzenia nowych plików.

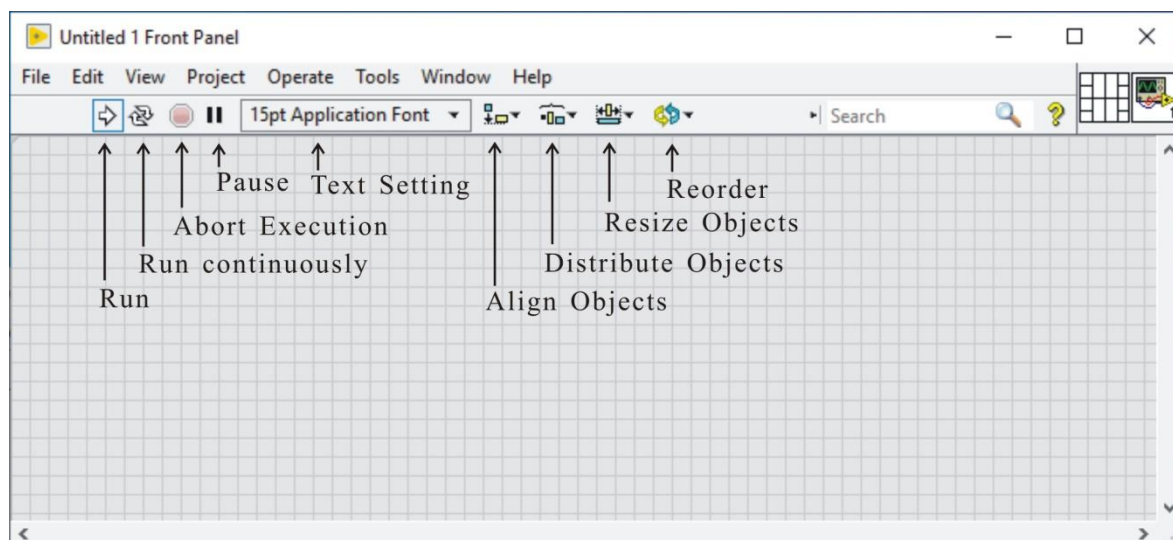


Rys. 1. Okno powitalne LabView.



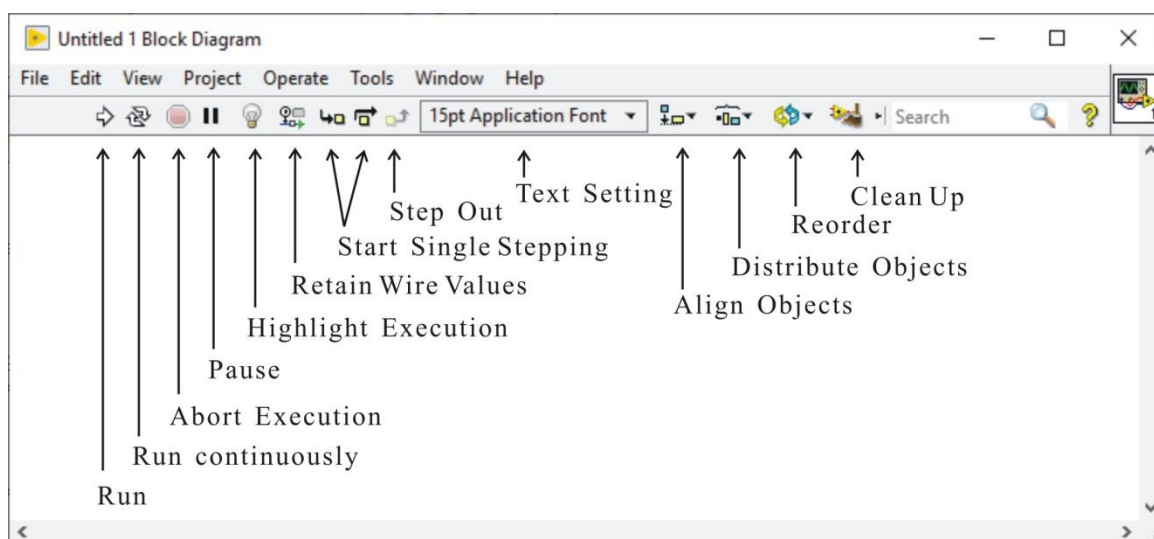
Rys. 2. Okno wyboru szablonów LabVIEW.

Po wybraniu *Blank VI* i naciśnięciu przycisku *Finish* na ekranie pojawiają się dwa okna podstawowe: *Front Panel* (okno z szarym tłem – rys. 3) oraz *Block Diagram* (okno z białym tłem – rys. 4).



Rys. 3. Okno *Front Panel*.

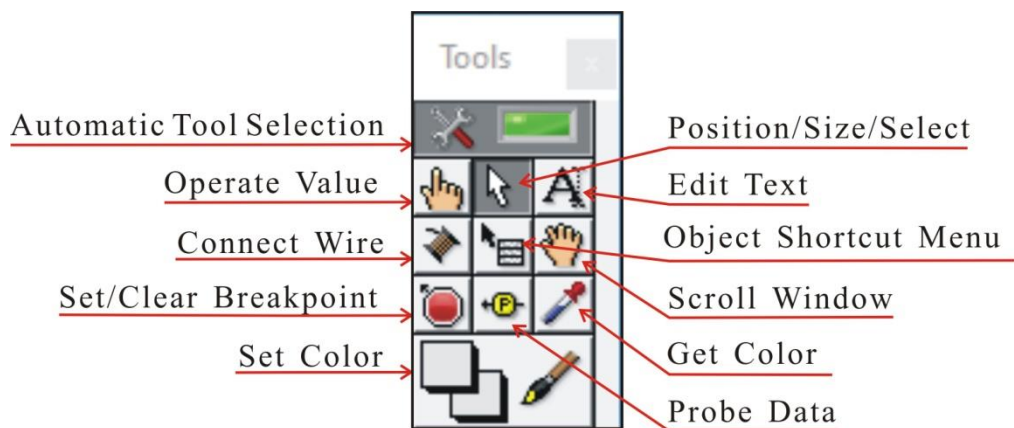
Okno *Front Panel* stanowi interfejs użytkownika korzystającego z programu napisanego w LabVIEW. Za jego pomocą tworzy się panel, na którym można umieścić przyciski, przełączniki, wskaźniki, okna dialogowe, okna z wynikami lub wykresy. Za pośrednictwem tego okna użytkownik steruje wykonaniem programu i odczytuje wyniki jego wykonania.



Rys. 4. Okno *Block Diagram*.

Okno *Block Diagram* zawiera kod programu napisany w języku graficznym. Wszystkie instrukcje języka graficznego LabVIEW mają reprezentacje graficzne (obiekty graficzne). Programowanie polega na wyborze odpowiednich instrukcji i ich łączeniu zgodnie z kolejnością przepływu danych lub wykonywania obliczeń.

Na rysunku 5 przedstawiono zestaw narzędzi *Tools*. Zestaw ten dostępny jest dla obu okien podstawowych (*Front Panel* i *Block Diagram*). Jeżeli paleta *Tools* nie jest widoczna na ekranie, należy je uaktywnić, wybierając z paska menu opcję *View»Tools Palette*.

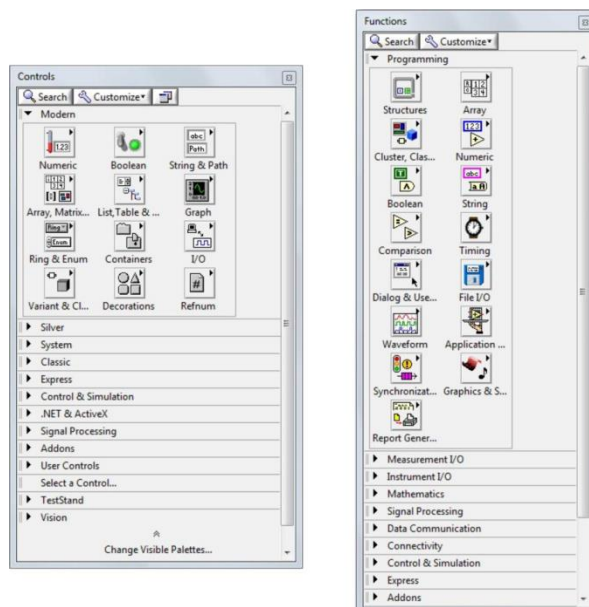


Rys. 5. Paleta Tools

Zawartość palety Tools:

- *Operate Value* – służy do zmiany nastaw obiektów na panelu,
- *Position/Size/Select* – służy do zmiany położenia i wymiarów oraz do wyboru obiektów,
- *Edit Text* – umożliwia edycję tekstów i tworzenie napisów,
- *Connect Wire* – służy do łączenia obiektów,
- *Object Shortcut Menu* – uaktywnia menu kontekstowe wybranego obiektu,
- *Scroll Window* – służy do przewijania zawartości okna panelu,
- *Set/Clear Breakpoint* – umożliwia ustawienie/usunięcie pułapki programowej,
- *Probe Data* – służy do ustawienia punktów obserwacji wartości zmiennych w programie,
- *Get Color* – służy do pobrania (kopiowania) koloru z obiektu,
- *Set Color* – umożliwia zmianę koloru obiektu.

Podczas tworzenia aplikacji do okna *Front Panel* wstawia się obiekty, które są dostępne w palecie *Controls* (rys. 6). Jeżeli paleta ta nie jest widoczna na ekranie, należy je uaktywnić, wybierając z paska menu opcję *View* i polecenie *Controls Palette*. Wygląd palety *Controls* można dostosować do wymagań użytkownika wybierając *Change Visible Palletes*.



Rys. 6. Palety Controls i Functions.

Podstawowe dostępne obiekty są podzielone na następujące grupy:

- *Numeric* – obiekty służące do wprowadzania lub wyświetlania danych liczbowych (pokręta, suwaki, wskaźniki cyfrowe),

- *Boolean* – obiekty do wprowadzania i odczytywania danych typu logicznego (przełączniki, kontrolki LED),
- *String & Path* – obiekty do wprowadzania danych w postaci łańcucha znakowego lub tabeli,
- *Array, Matrix & Cluster* – obiekty umożliwiające wprowadzenie i odczytanie danych w postaci macierzy lub rekordów,
- *List, Table & Tree* – obiekty do wprowadzania informacji wybranej z listy, tabeli lub drzewa,
- *Graph* – obiekty do prezentacji danych w postaci wykresów,
- *Ring & Enum* – obiekty wyboru opcji,
- *Containers* – obiekty grupujące inne obiekty,
- *User Controls* – obiekty utworzone przez użytkownika,
- *I/O* – obiekty obsługujące operacje wejścia/wyjścia.

Jeżeli aktywne jest okno podstawowe – *Block Diagram* – dostępne jest paleta *Functions* (rys. 6). Jeżeli paleta ta nie jest widoczna na ekranie, można ją uaktywnić, wybierając z paska menu opcję *View* i polecenie *Functions Palette*.

Paleta *Functions* zawiera podstawowe funkcje (obiekty programowania graficznego), które mogą być wstawiane do okna *Block Diagram*.

Ze znaczeniem i sposobem wykorzystania poszczególnych palet studenci zapoznają się w trakcie ćwiczenia laboratoryjnego.

2.2. Cel ćwiczenia

Celem ćwiczenia jest zapoznanie studentów z podstawami tworzenia programu w języku graficznym LabVIEW, służącym do programowania systemów pomiarowych.

- W przykładach pokazano sposób tworzenia interfejsu użytkownika (edycja zawartości okna *Front Panel*) oraz struktury programu (edycja zawartości okna *Block Diagram*).
- W tworzonych programach wykorzystano podstawowe obiekty (przyciski, wskaźniki, okna wykresów) oraz struktury sterujące: *While Loop* oraz *For Loop*.
- Przedstawiono sposób wprowadzania danych, wykonywania prostych operacji arytmetycznych, prezentacji wyników oraz zapisu do pliku.
- Pokazano sposób debugowania programu, uruchamiania programu w trybie pracy krokowej oraz tworzenie pętlań programowych.
- Zaprezentowano również sposób usuwania niezgodności typów danych przekazywanych pomiędzy połączonymi obiektami oraz sposób lokalizacji błędów programowych (analiza zawartości okna *Error List*).

3. Przebieg ćwiczenia

3.1. Utworzenie interfejsu użytkownika

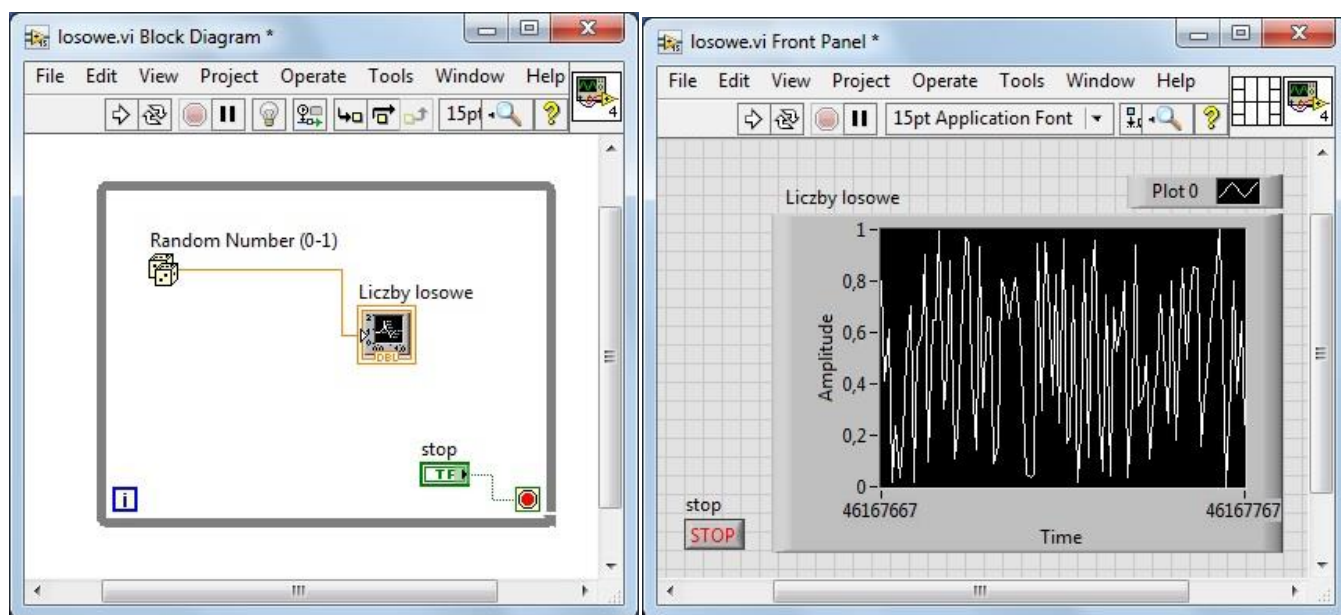
Tworzenie aplikacji VI rozpoczynamy od utworzenia panelu użytkownika.

- Utworzyć nowy plik poleceniem *Create Project » Blank VI*.
- Z palety *Tools* wybrać przycisk *Position/Size/Select* (jeżeli paleta *Tools* jest niewidoczna, uaktywnić ją poleceniem *View » Tools Palette*). Uwaga! W palecie *Tools* programu LabView możliwy jest wybór opcji *Automatic Tool Selection*. Opcja ta powoduje automatyczny wybór narzędzi podczas tworzenia programu. Na etapie nauki programowania, w celu zrozumienia efektów działania poszczególnych narzędzi z palety *Tools* opcja *Automatic Tool Selection* powinna być wyłączona. Poniższy program ćwiczenia został napisany dla przypadku, kiedy opcja ta została wyłączona.
- Z palety *Controls » Modern* wybrać obiekt *Boolean » Stop Button* i umieścić w oknie *Front Panel*. Wskazać kursorem wstawiony obiekt, nacisnąć prawy przycisk myszy i z menu obiektu wybrać *Visible Items*; sprawdzić działanie poleceń *Label*, *Caption*, *Boolean Text*.
- Z palety *Controls » Modern* wybrać obiekt *Graph » Waveform Chart* i wstawić do okna *Front Panel*. Z palety *Tools* wybrać przycisk *Edit Text* (jeżeli paleta *Tools* jest niewidoczna, uaktywnić ją, korzystając z polecenia *View » Tools Palette*), wskazać kursorem na napis *Waveform Chart*, nacisnąć lewy przycisk myszy i zmienić napis na *Liczby losowe*. Z palety *Tools* wybrać przycisk *Operate Value*; wskazać kursorem liczbę –10 na skali osi y, nacisnąć lewy przycisk myszy i zmienić początek skali osi y na 0; w podobny sposób zmienić wartość końca skali osi y z 10 na 1.

3.2. Utworzenie i uruchomienie programu w języku graficznym

Po rozmieszczeniu obiektów panelu użytkownika przechodzimy do tworzenia programu.

- Z menu wybrać opcję *Window » Show Diagram*; spowoduje to przejście do okna *Block Diagram* (do przełączania między oknami *Front Panel* i *Blok Diagram* można użyć skrótu <Ctrl-E>).
- Obiekty *Stop* i *Liczby losowe* są odpowiednikami obiektów wstawionych do okna *Front Panel*.
- Jeżeli na ekranie nie jest wyświetlana paleta *Functions*, wybrać z menu polecenie *View » Functions Palette*.
- Z palety *Functions » Programming* wybrać obiekt *Numeric » Random Number (0-1)* i przenieść do okna *Block Diagram*.
- Z palety *Tools* wybrać narzędzie *Connect Wire*; wskazać kursorem wyjście obiektu *Random Number (0-1)*, nacisnąć i zwolnić lewy przycisk myszy, następnie wskazać wejście obiektu *Liczby losowe* oraz ponownie nacisnąć i zwolnić lewy przycisk myszy; obiekty *Random Number (0-1)* i *Liczby losowe* zostaną połączone.



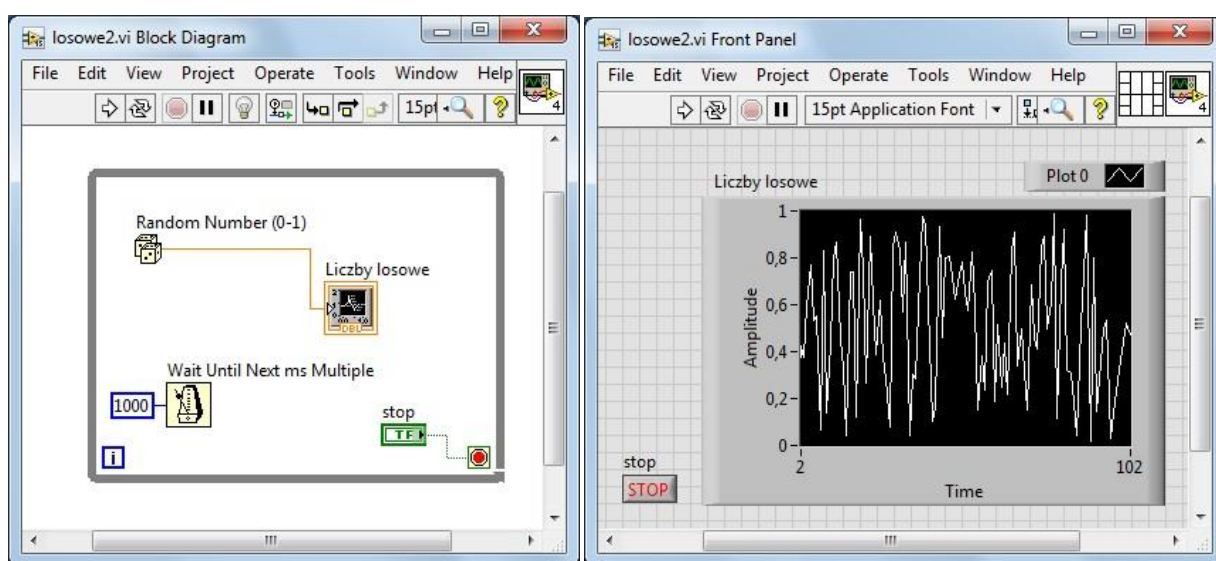
Rys. 7. Okna *Block Diagram* i *Front Panel* programu *losowe.vi*.

- W celu utworzenia pętli *While Loop* z palety *Functions » Programming* wybrać obiekt *Structures » While Loop*. Wybrany obiekt przenieść do okna *Block Diagram*. Nacisnąć i zwolnić lewy przycisk myszy. Przesuwając kursor, objąć powstałym prostokątem obiekty, które będą wykonywane w pętli (*Random Number (0-1)*, *Liczby losowe*, *Stop*).
- W prawym dolnym rogu powstałego prostokąta znajduje się wejście powodujące przerwanie wykonywania pętli. Z palety *Tools* wybrać przycisk *Connect Wire* i wejście powodujące przerwanie wykonywania pętli połączyć z obiektem *Stop*.
- Pętla *While Loop* jest wykonywana (ustawienie domyślne), gdy na wejście przerywające wykonywanie pętli jest podana wartość logiczna *FALSE*. Obiekt *Stop* generuje wartość *TRUE* po jego uaktywnieniu, dlatego po naciśnięciu przycisku *Stop* wykonywanie pętli zostanie przerwane. W celu zmiany ustawienia domyślnego warunku wykonania pętli należy wybrać z palety *Tools* narzędzie *Position/Size/Select*, wskazać kursorem ramkę pętli *While Loop*, nacisnąć prawy przycisk myszy i wybrać z menu kontekstowego opcję *Continue if True*. **Jak LabVIEW sygnalizuje zmianę opcji *Stop if True* na *Continue if True*?** Powrócić do ustawienia domyślnego *Stop if True*.
- Zawartość okna *Block Diagram* porównać z oknem przedstawionym na rys. 7.
- W oknie *Front Panel* wybrać opcję *Edit » Make Current Values Default* a następnie *File » Save* i zapisać program, nadając mu nazwę „*losowe_xx.vi*” (gdzie *xx* inicjały twórcy programu).
- Uruchomić program naciskając przycisk *Run* (okno *Front Panel* lub *Block Diagram*).
- Nacisnąć przycisk *STOP* i porównać zawartość okna *Front Panel* z oknem przedstawionym na rys. 7.

3.3. Dodanie taktowania do programu „losowe.vi”

Uzupełniamy tworzoną aplikację o obiekt odmierzający zadaną wartość czasu, tak aby uzyskać taktowanie działania pętli.

- Uaktywnić okno *Block Diagram*, a następnie z palety *Functions* » *Programming* wybrać przycisk *Timing* » *Wait Until Next ms Multiple* i przyporządkowany temu przyciskowi obiekt graficzny umieścić wewnątrz obszaru pętli *While Loop*.
- Z palety *Tools* wybrać narzędzie *Connect Wire*, wskazać kursorem lewą stronę obiektu *Wait Until Next ms Multiple* i prawym przyciskiem myszy wybrać z menu obiektu opcję *Create* » *Constant*; wpisać z klawiatury liczbę 1000 (czas w ms).
- W oknie *Front Panel* wybrać opcję *Edit* » *Make Current Values Default* a następnie *File* » *Save* i zapisać zmiany wprowadzone do programu w pliku „losowe2_xx.vi” (gdzie xx inicjały twórcy programu).
- Zawartość okna *Blok Diagram* porównać z rys. 8.
- Uruchomić program i sprawdzić jego działanie. **Jaki jest efekt wprowadzenia do programu obiektu *Wait Until Next ms Multiple*?**

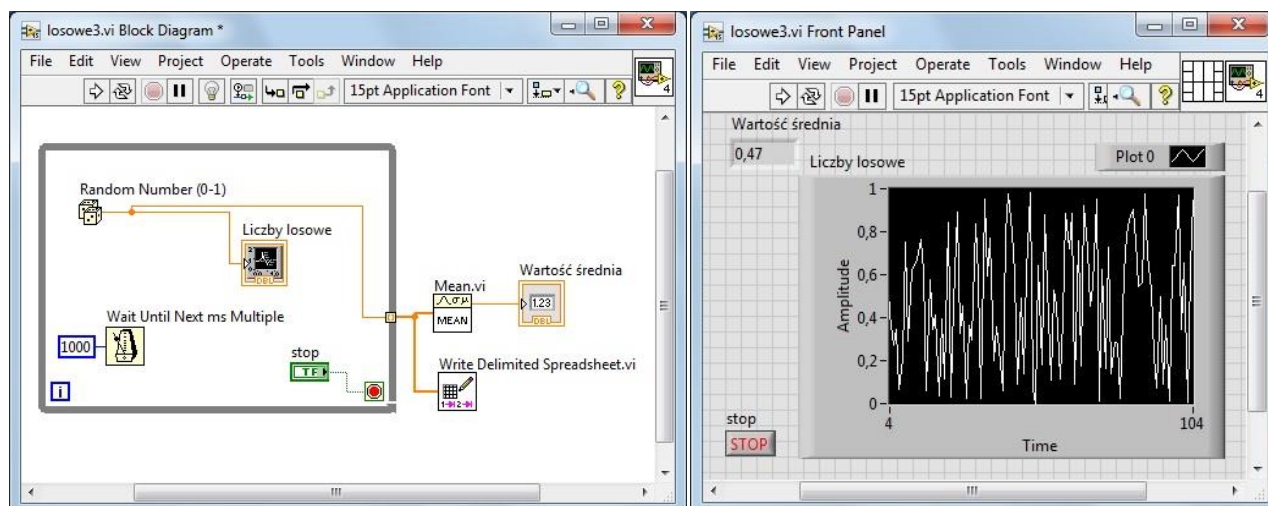


Rys. 8. Okna *Block Diagram* i *Front Panel* programu *losowe2.vi*.

3.4. Analiza wyników i zapis w pliku

- Przejść do okna *Block Diagram*.
- Z palety *Functions* » *Mathematics* wybrać obiekt *Probability&Statistics* » *Mean.vi* i umieścić go na zewnątrz pętli *While* » *Loop*. **Uwaga:** obiekt *Mean.vi* został umieszczony na zewnątrz pętli w celu obliczenia wartości średniej po przerwaniu pętli.
- Z palety *Functions* » *Programming* wybrać obiekt *File I/O* » *Write Delimited Spreadsheet.vi* i umieścić go również na zewnątrz pętli *While Loop*.
- Z palety *Tools* wybrać przycisk *Connect Wire*, wskazać kursorem wyjście *mean* obiektu *Mean.vi* i po kliknięciu prawym przyciskiem myszy wybrać z menu obiektu *Create* » *Indicator*. Zarówno w oknie *Block Diagram*, jak i w oknie *Front Panel* pojawi się obiekt *mean*, służący do wyświetlania wyników obliczeń, który automatycznie zostanie połączony z obiektem *mean.vi*. Zmienić nazwę obiektu *mean* na *Wartość średnia*.
- Połączyć wyjście obiektu *Random Number (0-1)* z wejściem *X* obiektu *Mean.vi*. Połączenie za pętlą zostanie narysowane przerywaną linią, co sygnalizuje, że zostały połączone wyprowadzenia reprezentujące różne typy danych.
- Pomarańczowy kwadrat w prostokącie symbolizującym pętlę *While Loop* sygnalizuje rodzaj danych generowane przez pętlę, które są przekazywane do dalszej części programu po przerwaniu pętli. Wskazać kursorem pomarańczowy kwadrat i nacisnąć prawy przycisk myszy.
- Wybrać z menu opcję *Tunnel Mode* » *Indexing*. Zmiana tej opcji spowoduje, że pętla będzie gromadziła dane w postaci tablicy jednowymiarowej. **Jak program sygnalizuje zmianę tej opcji?**

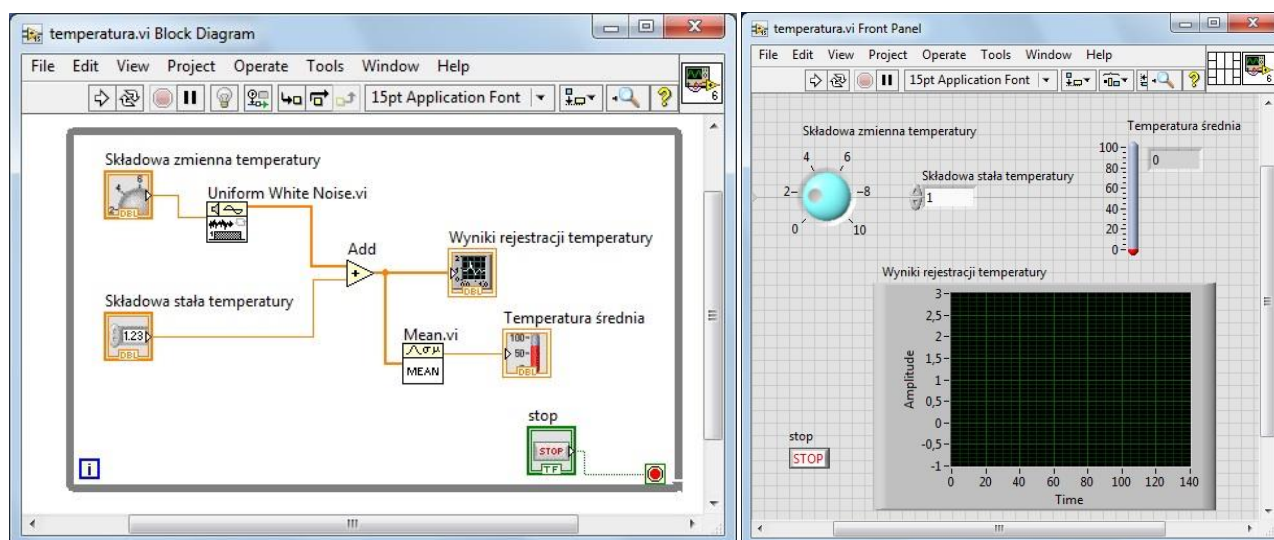
- Dane z tablicy jednowymiarowej mogą być bezpośrednio wprowadzone na wejście obiektu *Mean.vi*. Po usunięciu niezgodności typów danych linia przerywana łącząca obiekty zmieni się w linię ciągłą.
- Połączyć wyjście z pętli z wejściem *1D data* obiektu *Write Delimited Spreadsheet.vi*. Należy wybrać wejście *1D data*, ponieważ pętla *While Loop* tworzy jednowymiarową tablicę danych z liczb losowych generowanych przez obiekt *Random Number (0–1)*.
- W oknie *Front Panel* wybrać opcję *Edit » Make Current Values Default* a następnie *File » Save* i zapisać zmiany wprowadzone do programu w pliku „*losowe3_xx.vi*” (gdzie xx inicjały twórcy programu).
- Zawartość okna *Block Diagram* porównać z rys. 9.
- Uruchomić program i sprawdzić jego działanie. Po naciśnięciu przycisku *stop* pojawi się okno dialogowe, do którego należy wpisać nazwę pliku, w którym zostaną zapisane wyniki. Za pomocą dowolnego edytora tekstu odczytać utworzony plik.



Rys. 9. Okna *Block Diagram* i *Front Panel* programu *losowe3.vi*.

3.5. Edycja i debugowanie programu

W kolejnym przykładzie pokazano sposób debugowania programu i korekty błędów programowych oraz wprowadzania zmian wyglądu obiektów:



Rys. 10. Okna *Block Diagram* i *Front Panel* programu *temperatura.vi*.

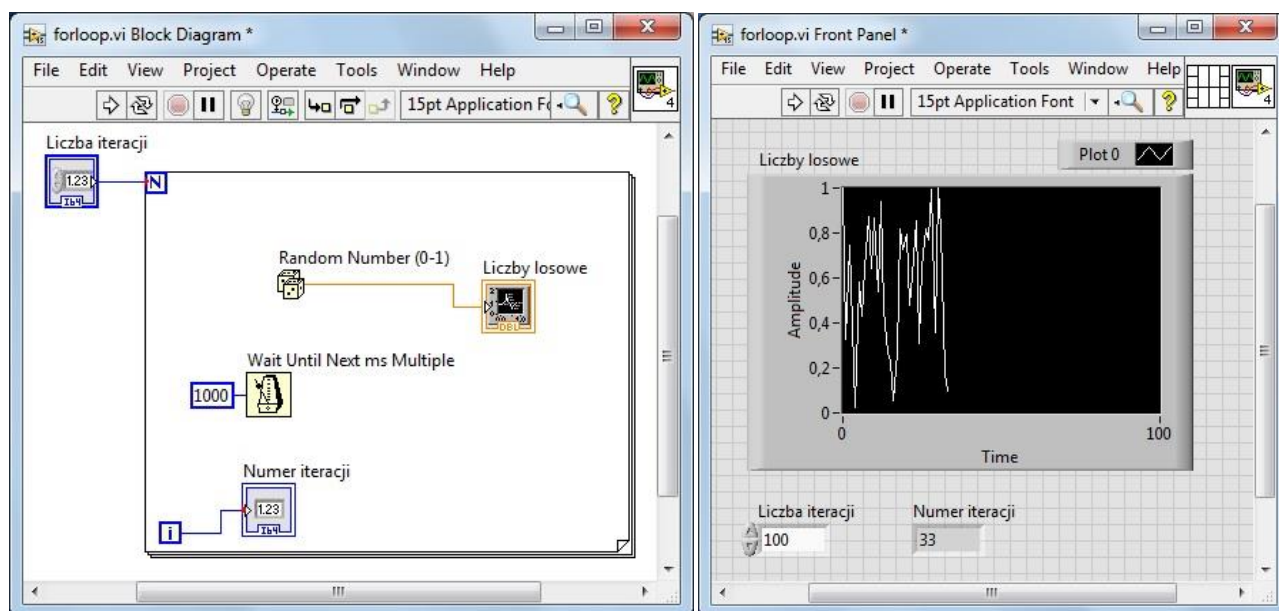
- Utworzyć nowy plik *Blank VI*, z palety *Controls » Modern* wybrać obiekty *Graph » Waveform Graph*, *Numeric » Numeric Control*, *Numeric » Knob*, *Numeric » Thermometer*, *Boolean » Stop Button*.

- Z palety *Tools* wybrać przycisk *Object Shortcut Menu*, wskazać kursorem dowolny obiekt i nacisnąć lewy przycisk myszy. Spowoduje to uaktywnienie menu obiektu, w którym należy sprawdzić ustawienie opcji *Visible Items* » *Label* (jeżeli nie jest ustawiona, to należy ją ustawić).
- Czynność tę powtórzyć dla wszystkich obiektów znajdujących się w oknach *Front Panel* i *Block Diagram*.
- Z palety *Tools* wybrać narzędzie *Edit text* i zmienić etykiety poszczególnych obiektów: z *Numeric* na *Składowa stała temperatury*, z *Thermometer* na *Temperatura średnia*, z *Knob* na *Składowa zmienna temperatury*, z *Waveform Graph* na *Wyniki rejestracji temperatury*.
- Z palety *Tools* wybrać przycisk *Object Shortcut Menu*, otworzyć menu obiektu *Wyniki rejestracji temperatury* i opcje *Visible Items* » *Plot Legend*, *Visible Items* » *Scale Legend*, *Visible Items* » *Graph Palette* ustawić jako nieaktywne.
- Z palety *Tools* wybrać narzędzie *Set Color*, wskazać kursorem obiekt *STOP* i kliknąć prawym przyciskiem myszy, aby zmienić jego kolor; w podobny sposób zmienić kolor obiektu *Składowa zmienna temperatury*. Sprawdzić także działanie przycisku *Get Color* z palety *Tools*.
- Otworzyć menu obiektu *Temperatura średnia* i wybrać opcję *Visible Items* » *Digital Display*.
- Zawartość okna *Front Panel* porównać z oknem przedstawionym na rys. 10.
- Przejść do okna *Block Diagram*, z palety *Functions* » *Signal Processing* wybrać obiekt *Signal Generation* » *Uniform White Noise.vi* i umieścić w oknie.
- Z palety *Functions* » *Mathematics* wybrać obiekt *Numeric* » *Add* i umieścić w oknie.
- Z palety *Functions* » *Mathematics* wybrać obiekt *Probability & Statistics* » *Mean.vee* i umieścić w oknie.
- Z palety *Functions* » *Programming* wybrać obiekt *Structures* » *While Loop* i prostokątem pętli otoczyć wszystkie obiekty znajdujące się w oknie *Block Diagram*; ustawić opcję *Stop if True* pętli *While Loop*.
- Z menu *Tools* wybrać przycisk *Connect Wire*.
- Wyjście obiektu *Składowa zmienna temperatury* połączyć z wejściem *amplitude* obiektu *Uniform White Noise.vi*.
- Wyjście *Uniform White Noise* obiektu *Uniform White Noise.vi* połączyć z wejściem *x* obiektu *Add*.
- Wyjście obiektu *Składowa stała temperatury* połączyć z wejściem *y* obiektu *Add*.
- Wyjście *x+y* obiektu *Add* połączyć z wejściem obiektu *Wyniki rejestracji temperatury* oraz wejściem *x* obiektu *Mean.vi*.
- Wyjście *mean* obiektu *Mean.vi* połączyć z wejściem obiektu *Temperatura średnia*.
- Wyjście obiektu *Stop* połączyć z wejściem *Stop if True* obiektu *While Loop*.
- W oknie *Front Panel* wybrać opcję *Edit* » *Make Current Values Default* a następnie *File* » *Save* i zapisać program w pliku „temperatura_xx.vi” (gdzie xx inicjały twórcy programu).
- Zawartość okna *Block Diagram* porównać z rys. 10, uruchomić program i sprawdzić jego działanie.
- Z palety *Tools* wybrać przycisk *Position/Size/Select*, zaznaczyć linię łączącą obiekty *Składowa stała temperatury* i *Add*, a następnie nacisnąć przycisk *Delete* na klawiaturze. Spowoduje to usunięcie połączenia.
- Ponownie spróbować uruchomić program. **Jak LabVIEW sygnalizuje błąd w programie? Co zawiera okno *Error List*?**
- W oknie *Error List* wybrać kursorem napis *Add:contains unwired or bad terminal*. Spowoduje to lokalizację błędu.
- Poprawić błąd i uruchomić program.
- Przejść do okna *Block Diagram*, nacisnąć przycisk *Highlight Execution* i uruchomić program. **Jaki jest efekt wyboru opcji *Highlight Execution*?** Sprawdzić także działanie przycisku *Do Not Retain Wires Values*.
- Z palety *Tools* wybrać przycisk *Set/Clear Breakpoint*, wskazać kursorem obiekt *Add* i kliknąć lewym przyciskiem myszy. Spowoduje to ustawienie pułapki programowej. Uruchomić program i sprawdzić działanie pułapki.
- Z palety *Tools* wybrać przycisk *Probe Data*, wskazać kolejno linie łączące obiekty i nacisnąć lewy przycisk myszy. Spowoduje to pojawienie się okna *Probe Watch Window* z nazwami i wartościami danych przesyłanych poszczególnymi liniami.
- Usunąć pułapkę programową.
- Sprawdzić krokowe uruchamianie programu poprzez naciskanie przycisku *Start Single Stepping*. W oknie *Block Diagram* są dwa przyciski *Start Single Stepping*. **Jaka jest różnica w ich działaniu?**

3.6. Użycie pętli For Loop

Zmodyfikować program losowe2.vi w sposób przedstawiony na rysunku 11.

- Wprowadzić zamiast pętli *While Loop* pętlę *For Loop* (*Functions » Programming » Structures » For Loop*) oraz obiekty *Liczba iteracji* (*Controls » Modern » Numeric » Numeric Control*) i *Numer iteracji* (*Controls » Modern » Numeric » Numeric Indicator*).
- W oknie *Front Panel* wybrać opcję *Edit » Make Current Values Default* a następnie *File » Save* i zapisać program, nadając mu nazwę „forloop_xx.vi” (gdzie xx inicjały twórcy programu).
- Z użyciem przycisku *Start Single Stepping* uruchamiać program krokowo; obserwować okna *Front Panel* i *Block Diagram*.
- Użyć narzędzia *Line Probe* z palety *Tools* do obserwacji danych przesyłanych poszczególnymi liniami.

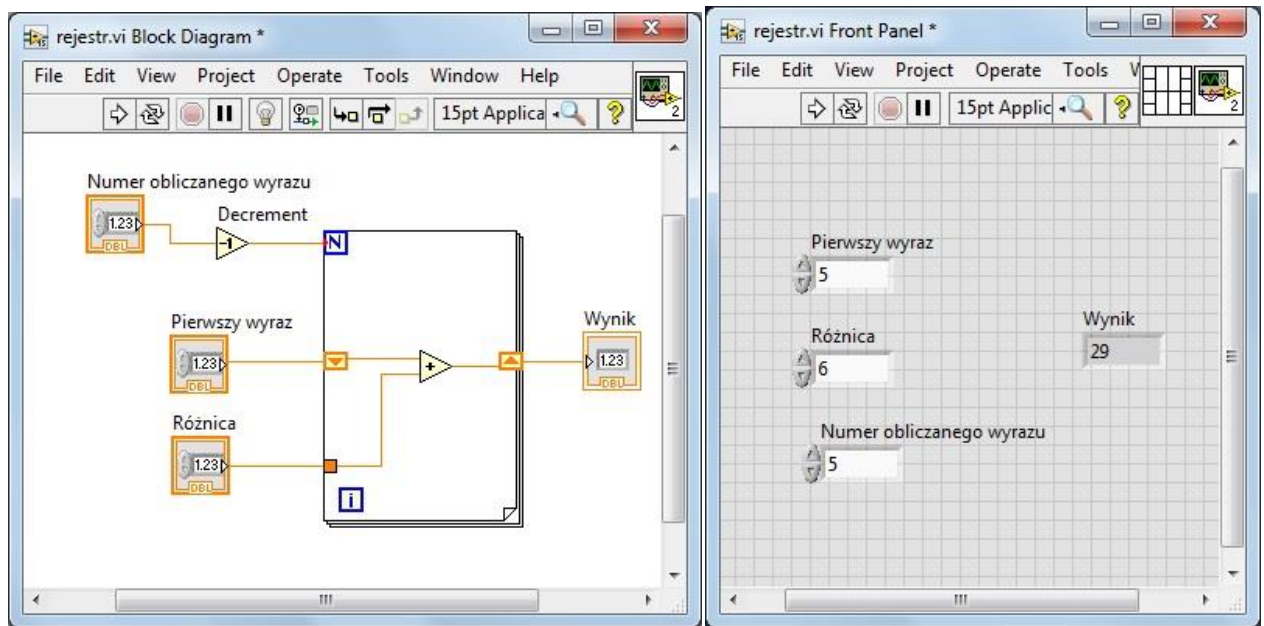


Rys. 11. Okna *Block Diagram* i *Front Panel* programu *forloop.vi*.

3.7. Użycie rejestru przesuwającego

W kolejnym przykładzie pokazano zastosowanie rejestru przesuwającego w pętli *For Loop*, do obliczenia n-tego wyrazu ciągu arytmetycznego.

- Utworzyć nowy program VI.
- W oknie *Front Panel* wstawić obiekty: *Pierwszy wyraz* (*Controls » Modern » Numeric » Numeric Control*), *Różnica* (*Controls » Modern » Numeric » Numeric Control*), *Numer obliczanego wyrazu* (*Controls » Modern » Numeric » Numeric Control*), *Wynik* (*Controls » Modern » Numeric » Numeric Indicator*).
- Przejść do okna *Block Diagram* i wstawić pętlę *For Loop* (*Functions » Programming » Structures » For Loop*), z palety *Tools* wybrać przycisk *Object Shortcut Menu*, wskazać kursorem ramkę pętli i kliknąć lewym przyciskiem myszy; z menu pętli wybrać opcję *Add Shift Register*.
- Z palety *Functions » Mathematics* wybrać obiekty *Numeric » Add*, *Numeric » Decrement* i wstawić je do programu.
- Przejść do okna *Front Panel* i wpisać zadane wartości (*Pierwszy wyraz* = 5, *Różnica* = 6, *Numer obliczanego wyrazu* = 5).
- Obiekty w oknie *Block Diagram* połączyć w sposób przedstawiony na rysunku 12.
- Z użyciem przycisku *Start Single Stepping* uruchamiać program krokowo; obserwować okna *Front Panel* i *Block Diagram*. Użyć narzędzia *Line Probe* z palety *Tools* do obserwacji danych przesyłanych poszczególnymi liniami. Sprawdzić jaka jest różnica w działaniu dwóch przycisków *Start Single Stepping*.
- W oknie *Front Panel* wybrać opcję *Edit » Make Current Values Default* a następnie *File » Save* i zapisać program, nadając mu nazwę „rejestr_xx.vi” (gdzie xx inicjały twórcy programu). **Czy do obliczeń iteracyjnych można zastosować pętlę *While Loop*?**



Rys. 12. Okna *Block Diagram* i *Front Panel* programu *rejestr.vi*.

4. Zadania sprawdzające do samodzielnego wykonania

- Uruchomić jeden z wcześniej zapisanych programów korzystając ze wskazówek prowadzącego sprawdzić co to jest i jak działa pomoc kontekstowa w LabVIEW.
- Napisać program do wyznaczania wartości maksymalnej, minimalnej i średniej z zadanych liczby wartości losowych.
- Napisać program do obliczania n -tego wyrażenia ciągu geometrycznego oraz sumy n wyrazów tego ciągu.
- Zaproponować i napisać własny program.

5. Pytania sprawdzające

- 5.1. Do czego służy okno *Font Panel* i jakie palety są związane z tym oknem?
- 5.2. Do czego służy okno *Block Diagram* i jakie palety są związane z tym oknem?
- 5.3. Jak działa pomoc kontekstowa w LabVIEW?
- 5.4. Dlaczego przed zapisem programu z menu LabVIEW należy wybrać polecenie *Edit » Make Current Values Default*?
- 5.5. Jak program sygnalizuje zmianę opcji *Tunnel Mode » Last Value* na *Tunnel Mode » Indexing* w pętli *While Loop*?
- 5.6. Jaka jest różnica w działaniu pętli *While Loop* oraz *For Loop* przy zmianie wartości parametru *Stop if True* na *Continue if True*?
- 5.7. Jak LabVIEW sygnalizuje zmianę warunku zakończenia działania pętli?
- 5.8. Jaki jest efekt wprowadzenia obiektu *Wait Until Next ms Multiple* do pętli programowej?
- 5.9. Jak program sygnalizuje niezgodność typów danych połączonych obiektów?
- 5.10. Jak LabVIEW sygnalizuje błąd w programie?
- 5.11. Jakie informacje zawiera okno *Error List*?
- 5.12. Jak zlokalizować miejsce błędu w programie?
- 5.13. Jaki jest efekt wyboru opcji *Highlight Execution*?
- 5.14. W oknie *Block Diagram* są dwa przyciski *Start Single Stepping*. Jaka jest różnica w ich działaniu?
- 5.15. Jaki jest efekt wyboru opcji *Do Not Retain Wires Values*?
- 5.17. Jak działają polecenia *Get Color* i *Set Color* z palety *Tools*?
- 5.18. Jak działa *Shift Register* w pętli *For Loop*?
- 5.19. Czy do obliczeń iteracyjnych można zastosować pętlę *While Loop* (uzasadnij odpowiedź)?

6. Zawartość sprawozdania

Sprawozdanie z ćwiczenia powinno zawierać:

- zrzuty ekranów uruchomionych programów z odpowiednimi podpisami,
- programy w postaci plików,
- ponumerowane odpowiedzi na pytania zadane w instrukcji (Odpowiedzieć tylko na pytania z punktu 5. Odpowiedzi umieścić w sprawozdaniu przed wnioskami.),
- uwagi i wnioski.