

Ćwiczenie: "Środowisko programistyczne LabVIEW"

Opracowane w ramach projektu: "Wirtualne Laboratoria Fizyczne nowoczesną metodą nauczania" realizowanego przez Warszawską Wyższą Szkołę Informatyki.

Zakres ćwiczenia:

1. Zapoznanie ze środowiskiem LabVIEW i językiem G.
2. Podstawowe funkcje LabVIEW.
3. Wzorce projektowe w środowisku LabVIEW.
4. LabVIEW w przykładach.
 - Podstawowe operacje matematyczne.
 - Operacje na zbiorach.
 - Podstawowe operacje na plikach.
 - Instrukcje sterujące i pętle.
 - Ciągi tekstowe.
 - Pojęcie funkcji w LabVIEW.
 - Generacja przebiegów i ich wyświetlanie.
 - Komunikacja sieciowa.
 - LabVIEW w przykładach – przykładowe ćwiczenia.



Środowisko programistyczne LabVIEW

**WIRTUALNE
LABORATORIA
FIZYCZNE
NOWOCZESNĄ
METODĄ
NAUCZANIA**

Innowacyjny program
nauczania fizyki w szkołach
ponadgimnazjalnych

Projekt jest realizowany przez
Warszawską Wyższą Szkołę Informatyki

Człowiek – najlepsza inwestycja

 KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

 WARSZAWSKA
WYŻSZA SZKOŁA
INFORMATYKI

EUROPEJSKI
FUNDUSZ SPOŁECZNY 

Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Autor: Marcin Godziemba-Maliszewski

Radom 2013

Scenariusz prowadzenia ćwiczenia

Celem prezentowanego kursu jest zapoznanie się z podstawami graficznego programowania w środowisku LabVIEW firmy National Instrument. Kurs pozwala na zapoznanie się ze środowiskiem i jego możliwościami jak i ze sposobem programowania. Kurs i jego prezentacja zostały podzielone na cztery funkcjonalne bloki:

1. Zapoznanie ze środowiskiem LabVIEW i językiem G – opisuje możliwości i podstawowe funkcje środowiska.
2. Podstawowe funkcje LabVIEW – omawia strukturę aplikacji tworzonych z wykorzystaniem środowiska LabVIEW i języka G oraz opisuje podstawowe funkcje używane przy programowaniu i dostęp do nich.
3. Wzorce projektowe w środowisku LabVIEW – omawia struktury budowy programów w środowisku.
4. LabVIEW w przykładach – prezentuje podstawowe techniki programowania aplikacji i kodowania funkcjonalności z wykorzystaniem języka G.






1. Zapoznanie ze środowiskiem LabVIEW i językiem G.

Celem ćwiczenia jest zapoznanie się ze środowiskiem programistycznym LabVIEW oraz poznanie jego podstawowej funkcjonalności i możliwości. Obejmuje slajdy od 4 do 54.

Gdzie można uzyskać więcej informacji i pomoc – slajd 5.

Przedstawienie słuchaczowi informacji związanych z możliwością uzyskania pomocy przy pracy ze środowiskiem w tym pozycji drukowanych i przykładów.

- Programy przykładowe
Help » Find Examples ...
- Zasoby sieci Web (ni.com)
LabVIEW (ni.com/labview oraz ni.com/labviewse)
Application Notes
- Pozycje książkowe:

	Wiesław Tłaczała: „Środowisko LabVIEW w eksperymencie wspomaganym komputerowo”
	Dariusz Świsulski: „Oprogramowanie wirtualnych przyrządów pomiarowych w LabVIEW”
	Wiesław Winięcki: „Graficzne zintegrowane środowiska programowe do programowania komputerowych systemów pomiarowo-kontrolnych”
	Piotr Lesiak, Dariusz Świsulski: „Komputerowa technika pomiarowa w przykładach”
	NI LabVIEW Student Edition Software Suite – książka wraz z oprogramowaniem nieograniczonym czasowo w wersji studenckiej

Opis środowiska i jego możliwości – slajdy: 5 do 22.

Przedstawienie genezy środowiska oraz jego zastosowań i możliwości.

LabVIEW Project – slajdy: 23 do 54.

Prezentacja LabVIEW Project jako sposobu zarządzania projektem i utrzymania porządku w plikach projektu. Zaprezentowanie możliwości automatyzacji procesów wytwarzania aplikacji w tym sposobów tworzenia pakietów dystrybucyjnych aplikacji.

2. Podstawowe funkcje LabVIEW i języka G.

Celem modułu jest zaprezentowanie środowiska programistycznego LabVIEW, językiem programowania graficznego (język G) oraz poznanie jego podstawowych funkcji programistycznych. Obejmuje slajdy od 54 do 80. W ramach modułu przedstawione zostaną następujące zagadnienia:

- Interfejs graficzny środowiska
- Panel czołowy – interfejs graficzny aplikacji i funkcji
- Diagram programu
- Graficzna reprezentacja i budowa funkcji używanych w LabVIEW
- Palety kontrolki i funkcji: organizacja i podział
- Paleta narzędzi
- Paski narzędzi
- Typy danych i ich reprezentacja graficzna na diagramie
- Konwersja typów i jej zasady
- Przydatne skróty klawiaturowe w trakcie pracy ze środowiskiem

3. Wzorce projektowe i dobre praktyki w LabVIEW.

Celem modułu jest zaprezentowanie podstawowych modeli budowy oprogramowania oraz dobrych praktyk w trakcie tworzenia oprogramowania w środowisku LabVIEW. Obejmuje slajdy od 81 do 109. W ramach modułu przedstawione zostaną następujące zagadnienia:

- Jak pisać dobre programy w graficznym środowisku programowania
- Omówienie podstawowych architektur oprogramowania budowanego w ramach LabVIEW
- Co to jest wzorzec programowania i jak on przekłada się na działanie i funkcjonalność tworzonego oprogramowania
- Wzorce projektowe - Single Loop Architecture
- Wzorce projektowe - Parallel Loop Architecture
- Wzorce projektowe – Event Structure
- Wzorce projektowe – State Machine
- Wzorce projektowe – producer/consumer
- Wzorce projektowe – „oo – object orientet” programming
- Programy działające w oparciu o pętle Master / Slave
- Programy działające w oparciu o kolejki Queued Message Handler
- „Daemony” – usługi działające w tle
- Splash Screen (popularny Luncher)

4. LabVIEW w przykładach.

Celem modułu jest zaprezentowanie podstaw programowania graficznego w środowisku LabVIEW, w tym tworzenie prostych aplikacji (VI), podstawowe struktury programistyczne, obróbka danych i zapis danych oraz komunikacja sieciowa. Obejmuje slajdy od 110 do 315. W ramach modułu przedstawione zostaną następujące zagadnienia:

- Podstawowe operacje matematyczne.
- Operacje na zbiorach.
- Podstawowe operacje na plikach.
- Instrukcje sterujące i pętle.
- Ciągi tekstowe.
- Pojęcie funkcji w LabVIEW.
- Generacja przebiegów i ich wyświetlanie.
- Komunikacja sieciowa.
- LabVIEW krok po kroku – przykłady budowy aplikacji w tym budowa prostej aplikacji typu klient-serwer.

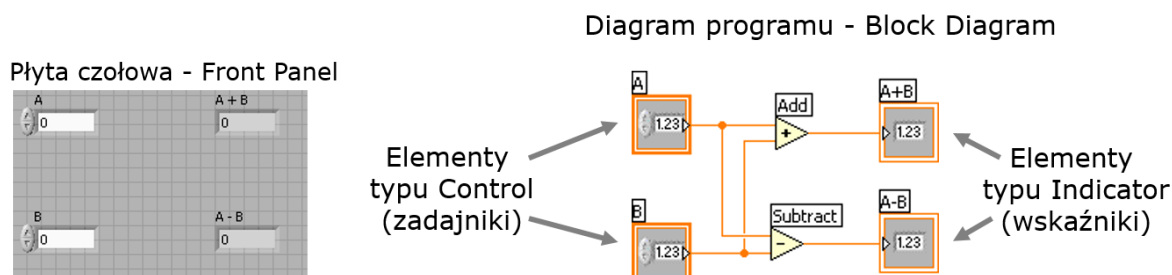
Wstęp do programowania w środowisku LabVIEW – slajdy: 111 do 132.

Celem modułu jest zaprezentowanie sposobu budowania aplikacji w środowisku LabVIEW na podstawie realizacji prostych przykładów obrazujących poruszane zagadnienie.

Ćwiczenie 1 – slajdy 112 do 114:

Napisać program realizujący dodawanie i odejmowanie dwóch liczb A i B. W trakcie ćwiczenia prezentowane są podstawowe funkcje palety narzędzi i obsługi diagramu połączeń. Ćwiczenie składa się z następujących etapów:

1. Umieszczenie na panelu czołowym zadajników i wskaźników
2. Wstawienie odpowiednich funkcji do diagramu programu
3. Połączenie elementów diagramu
4. Przetestowanie działania programu



Omówienie posługiwania się paletą narzędzi i systemu pomocy środowiska – slajd 115.

Omówienie funkcji śledzący i sposobu prezentacji błędów w środowisku – slajd 116.

Omówienie przepływu danych w programie – slajd 117.

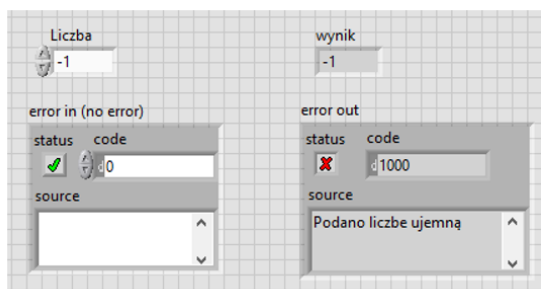
Funkcje matematyczne w LabVIEW – slajdy 118 do 124

Grupowanie danych – struktury (clusters) – slajdy 125 do 126

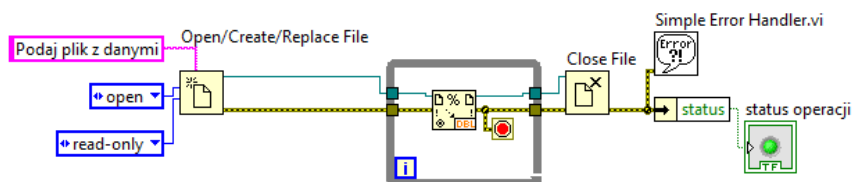
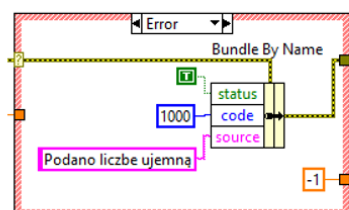
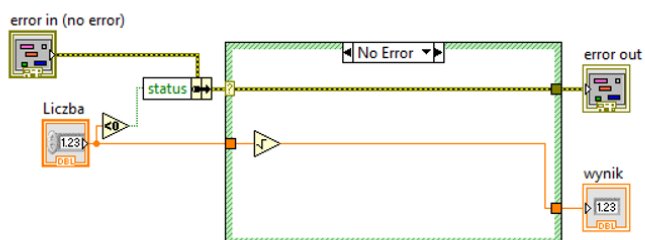
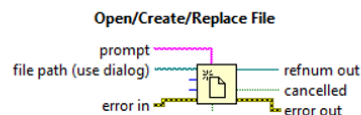
Struktura do wyświetlania błędów i obsługa błędów – slajdy 126 do 127

Ćwiczenie 2 – slajdy 129 do 131:

W ramach ćwiczenia zaprezentowana zostanie struktura do obsługi błędów w aplikacji.



Niektóre funkcje mają wyjście błędu i nie ma potrzeby sprawdzania poprawności danych wejściowych.

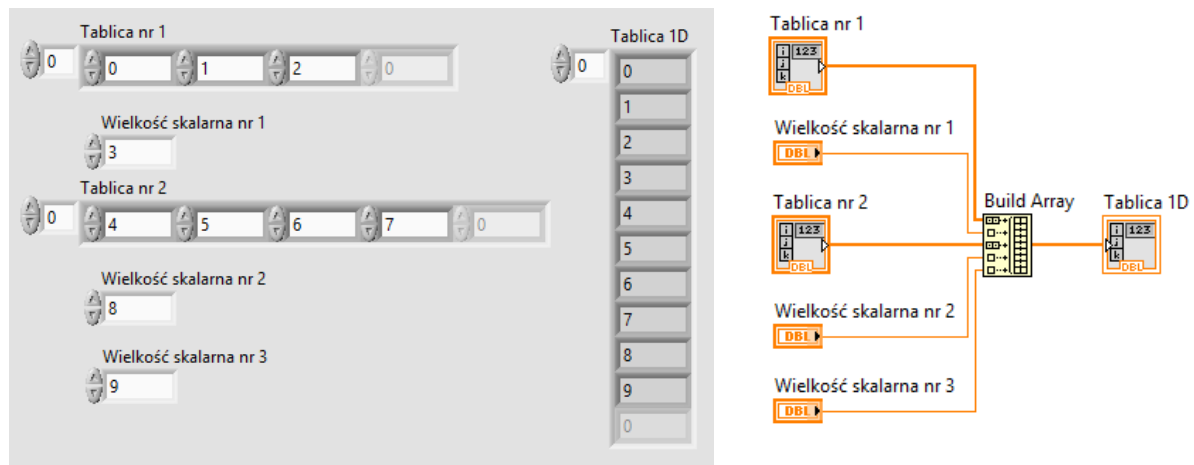


Operacje na zbiorach – slajdy: 132 do 139.

Celem modułu jest zaprezentowanie realizacji operacji na zbiorach danych oraz prezentacja polimorficznych funkcji automatycznie dostosowujących się do typów danych

Ćwiczenie 3 – slajdy 138 do 139:

W ramach przykładu zaprezentowana zostanie struktura do obsługi błędów w aplikacji.



Podstawowe operacje na plikach – slajdy: 140 do 145.

Celem modułu jest zaprezentowanie możliwości zapisu danych w plikach dyskowych

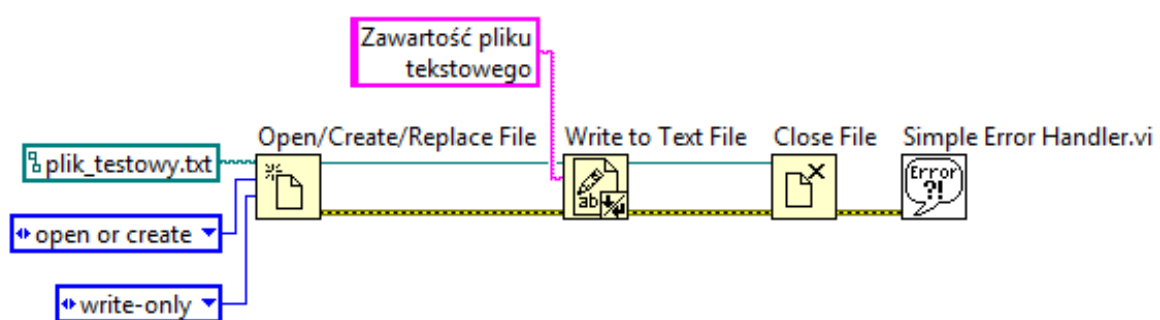
Omówienie sposobu przechowywania danych – slajd 141

Omówienie funkcji służących do odczytu i zapisu danych do pliku – slajdy: 142 do 143

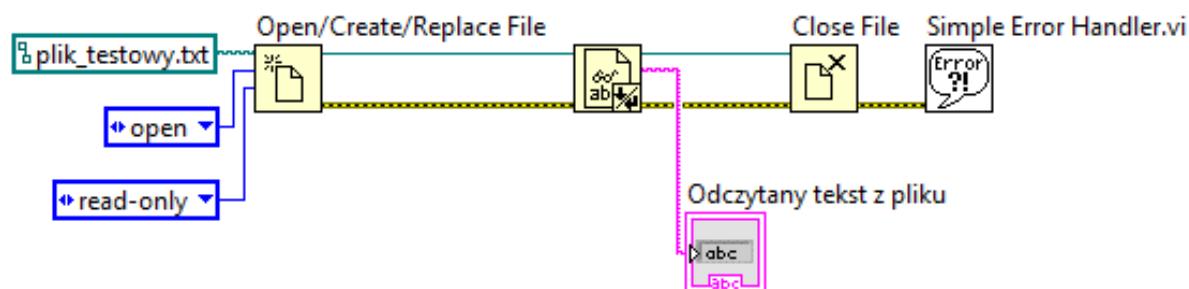
Ćwiczenie 4 – slajdy 144 do 145:

W ramach ćwiczenia zaprezentowana zostanie aplikacja pozwalająca na zapis i odczyt danych tekstowych do pliku.

Odczyt danych:



Zapis danych:



Instrukcje sterujące i pętle– slajdy: 146 do 154.

Celem modułu jest zaprezentowanie graficznej reprezentacji instrukcji sterujących i pętli w środowisku LabVIEW.

Omówienie pętli for loop – slajdy: 147 do 154

Ćwiczenie 5 – slajd 149 do 150:

W ramach ćwiczenia zaprezentowana zostanie pętla for loop budująca tablicę jednowymiarową. W ramach ćwiczenia zaprezentowane zostanie wyjście danych.

Interfejs użytkownika:

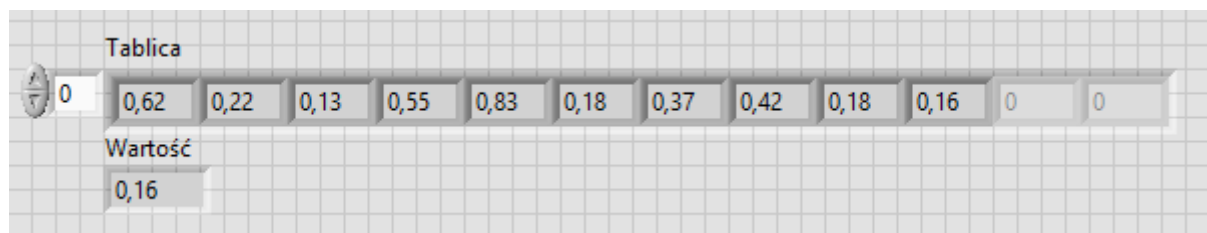
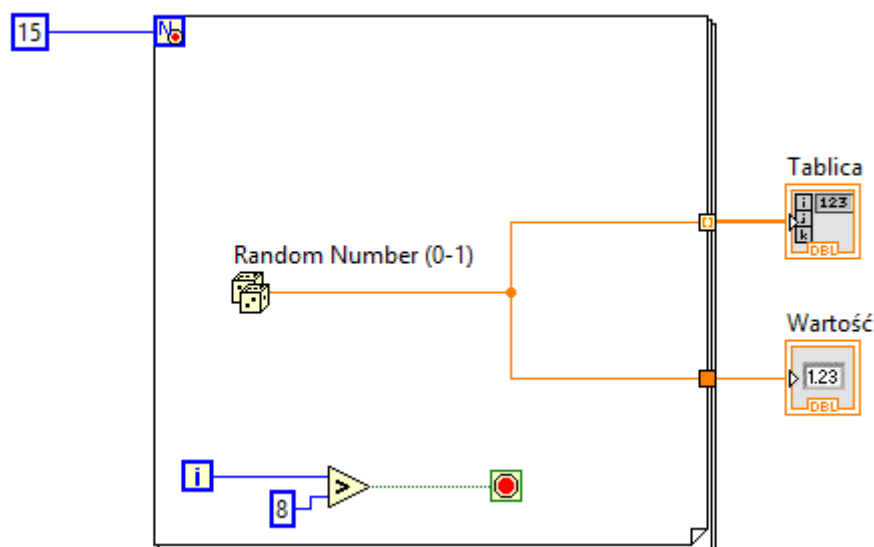


Diagram programu:



Ćwiczenie 7 – slajdy 152 do 1154:

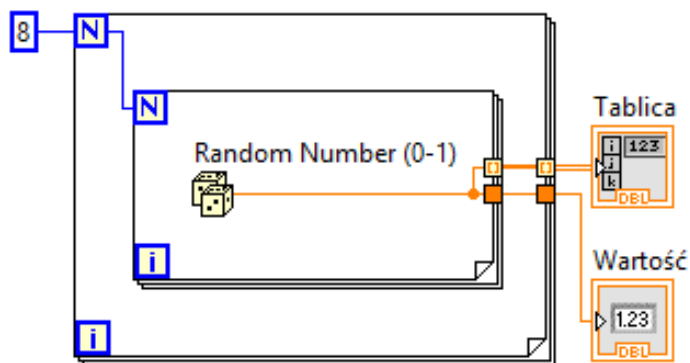
W ramach ćwiczenia zaprezentowana zostanie pętla for loop budująca tablicę dwuwymiarową. W ramach ćwiczenia zaprezentowane zostanie wyjście danych oraz sposoby auto indeksacji danych wyjściowych.

Interfejs użytkownika:

0	0,29	0,46	0,78	1	0,03	0,84	0,58	0,51	0
0	0,67	0,74	0,9	0,12	0,33	0,81	0,46	0,32	0
	0,98	0,69	0,45	0,45	0,71	0,93	0,16	0,97	0
	0,28	0,71	0,52	0,29	0,96	0,69	0,32	0,78	0
	0,06	0,25	0,21	0,44	0,71	0,5	0,96	0,67	0
	0,78	0,97	0,13	0,76	0,72	0,54	0,8	0,49	0
	0,99	0,58	0,09	0,73	0,05	0,95	0,22	0,12	0
	0,31	0,11	0,48	0,38	0,02	0,69	0,87	0,47	0
	0	0	0	0	0	0	0	0	0

Wartość
0,47

Diagram programu:



Ćwiczenie 8 – slajdy 144 do 145:

W ramach ćwiczenia zaprezentowana zostanie pętla for loop budująca tablice jednowymiarowe z danymi do wyświetlania na wykresie.

Interfejs użytkownika:

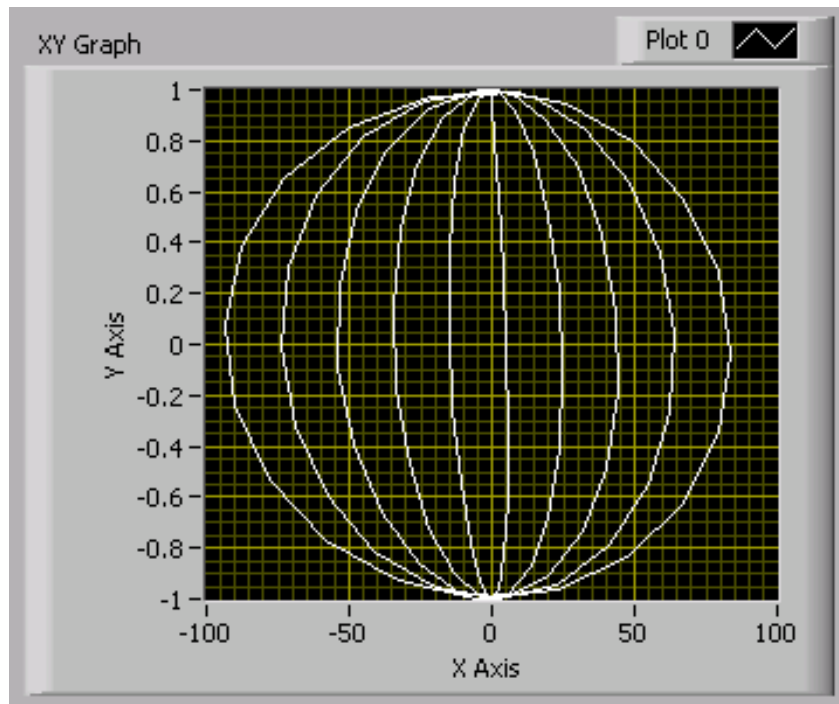
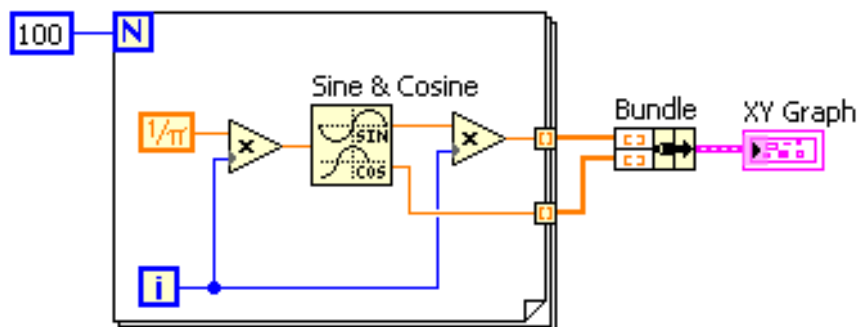


Diagram programu:



Omówienie pętli while – slajdy: 155 do 158

Ćwiczenie 9 – slajd 158:

W ramach ćwiczenia zaprezentowana zostanie pętla while. W ramach ćwiczenia zaprezentowane zostanie wyjście danych oraz sposoby zasady wejścia i wyjścia danych do i z pętli.

Interfejs użytkownika:

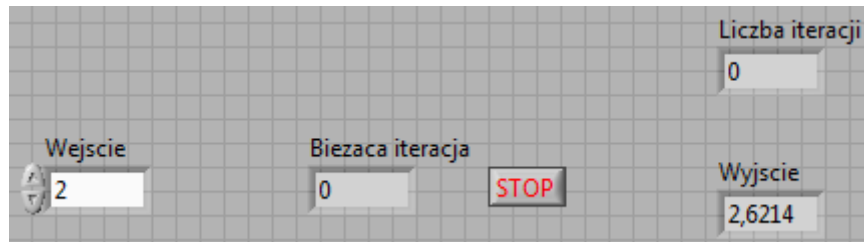
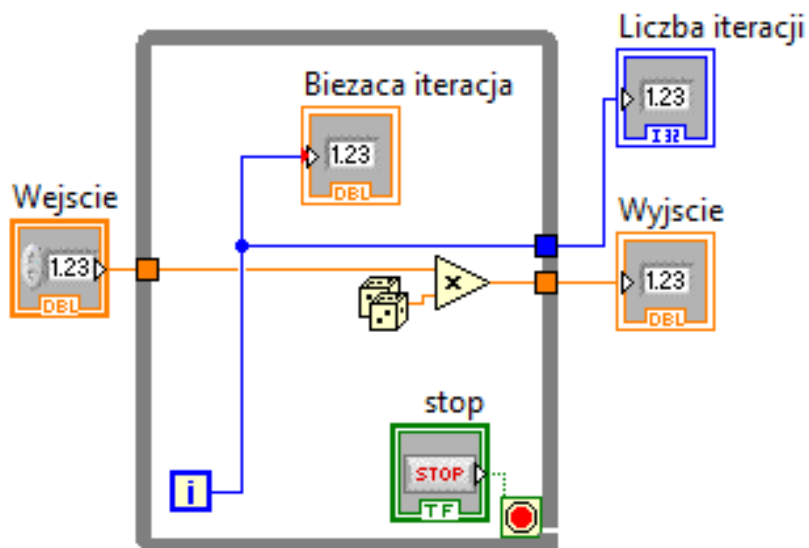


Diagram programu:



Instrukcje sterujące – case structure – slajdy: 159 do 164

Omówienie reprezentacji graficznej konstrukcji sterującej pozwalającej wykonać fragment kodu w zależności od wartości wyrażenia sterującego.

Ćwiczenie 10 – slajd 159:

W ramach ćwiczenia zaprezentowana zostanie konstrukcja Case structure pozwalająca wykonać fragment kodu w zależności od wartości wyrażenia sterującego – w tym przypadku wartości logicznej (true-false).

Interfejs użytkownika:

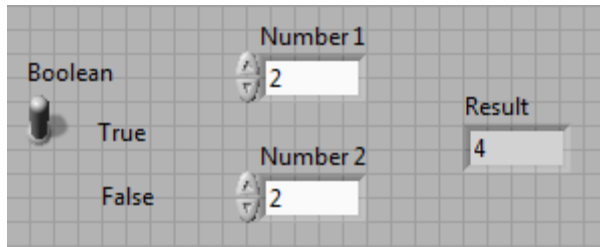
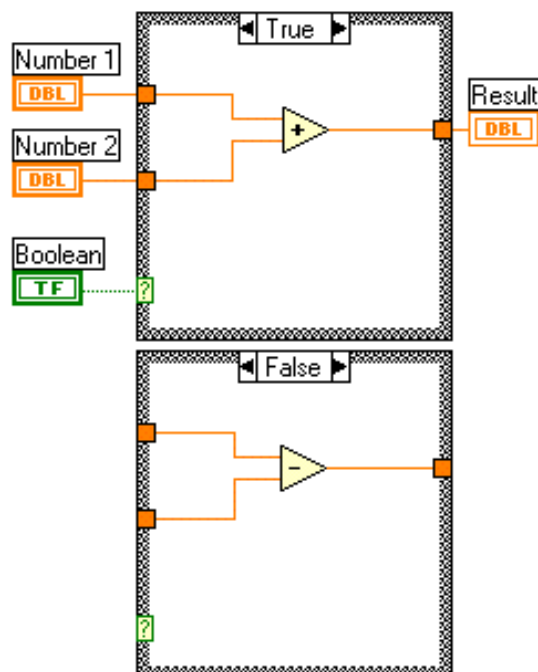


Diagram programu:



Ćwiczenie 11 – slajd 160:

W ramach ćwiczenia zaprezentowana zostanie konstrukcja case structure może być sterowana różnymi elementami – od logicznych przez numeryczne aż po tekstowe..

Interfejs użytkownika:

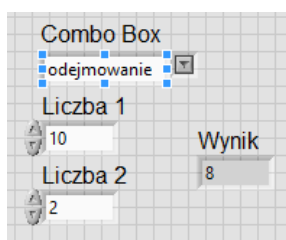
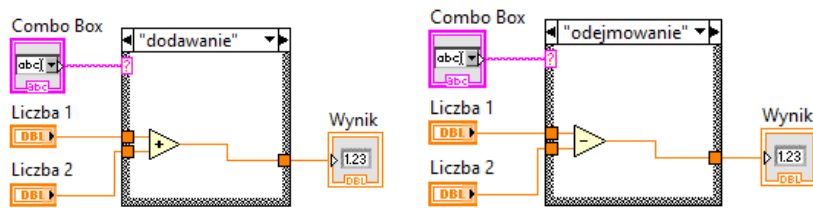


Diagram programu:



Instrukcje sterujące – select – slajd 161

Omówienie reprezentacji graficznej elementarnej instrukcji sterująca „select” umożliwiającą wybieranie ścieżki przyływu danych (dowolnych: numeryczne, tekstowe, struktury aż po dynamiczne strumienie) za pośrednictwem selektora binarnego (true, false)

Ćwiczenie 12 – slajd 161:

W ramach ćwiczenia zaprezentowana zostanie elementarna instrukcja sterująca „select” umożliwiającą wybieranie ścieżki przyływu danych.

Interfejs użytkownika:

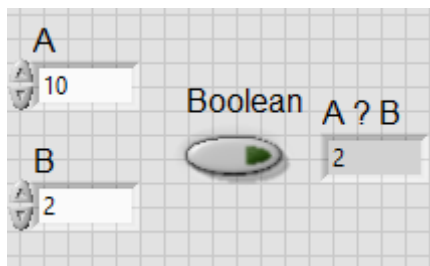
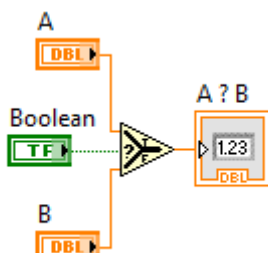
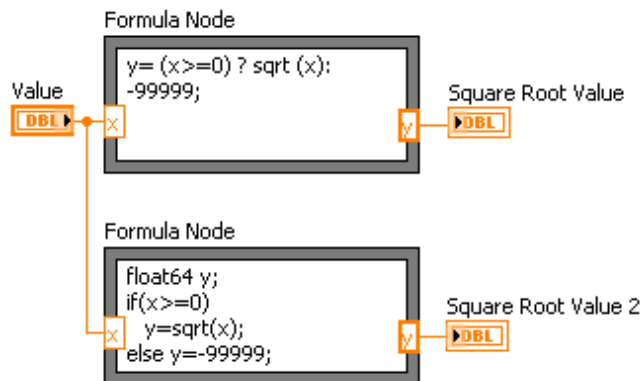


Diagram programu:



Instrukcje sterujące – formula node – slajd 162

Omówienie reprezentacji graficznej funkcji Formula node jako złożonej instrukcji warunkowej.



Funkcje sterujące opóźnieniem wykonania – slajd 163 i 164

Omówienie reprezentacji graficznej funkcji pozwalających na sterowanie opóźnieniem wykonania poszczególnych iteracji w pętlach.

Rejestry przesuwne i sprzężenia – slajd 165 i 170

Omówienie reprezentacji graficznej rejestru przesuwne i sprzężenia zwrotnego umożliwiające dostęp do danych z poprzedniej lub z poprzednich iteracji pętli..

Ćwiczenie 13 – slajdy 167 do 168:

W ramach ćwiczenia zaprezentowana zostanie generacja przebiegu losowego z uśrednianiem - demonstracja rejestrów przesuwnych.

Interfejs użytkownika:

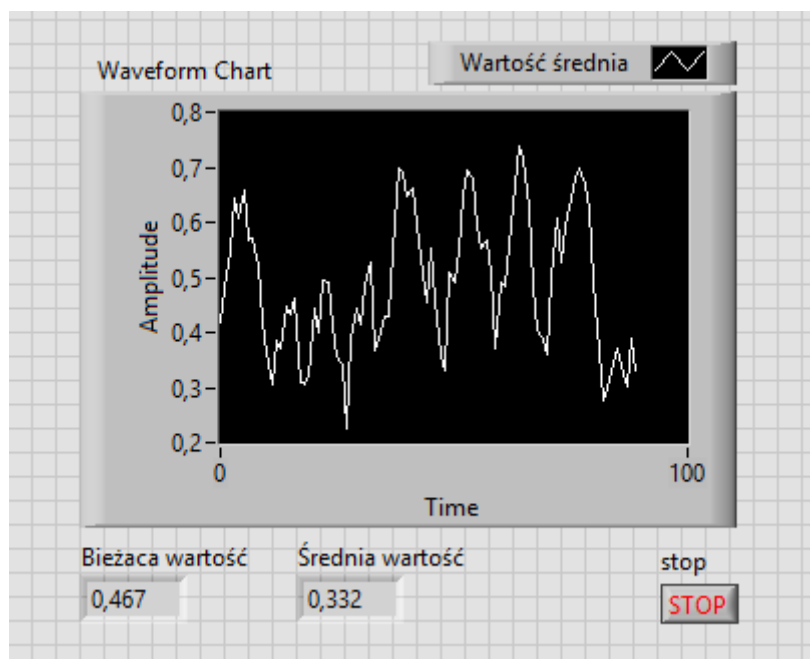
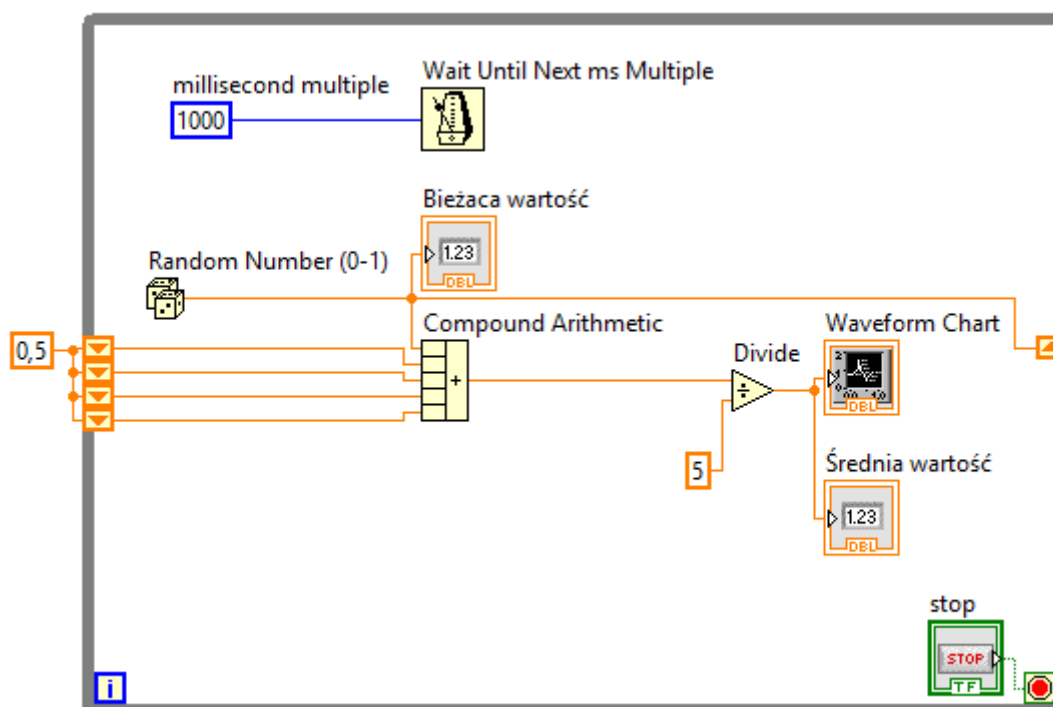


Diagram programu:



Ćwiczenie 14 – slajdy 169 do 170:

W ramach ćwiczenia zaprezentowana zostanie pętla for loop oraz instrukcje sterujące.

Interfejs użytkownika:

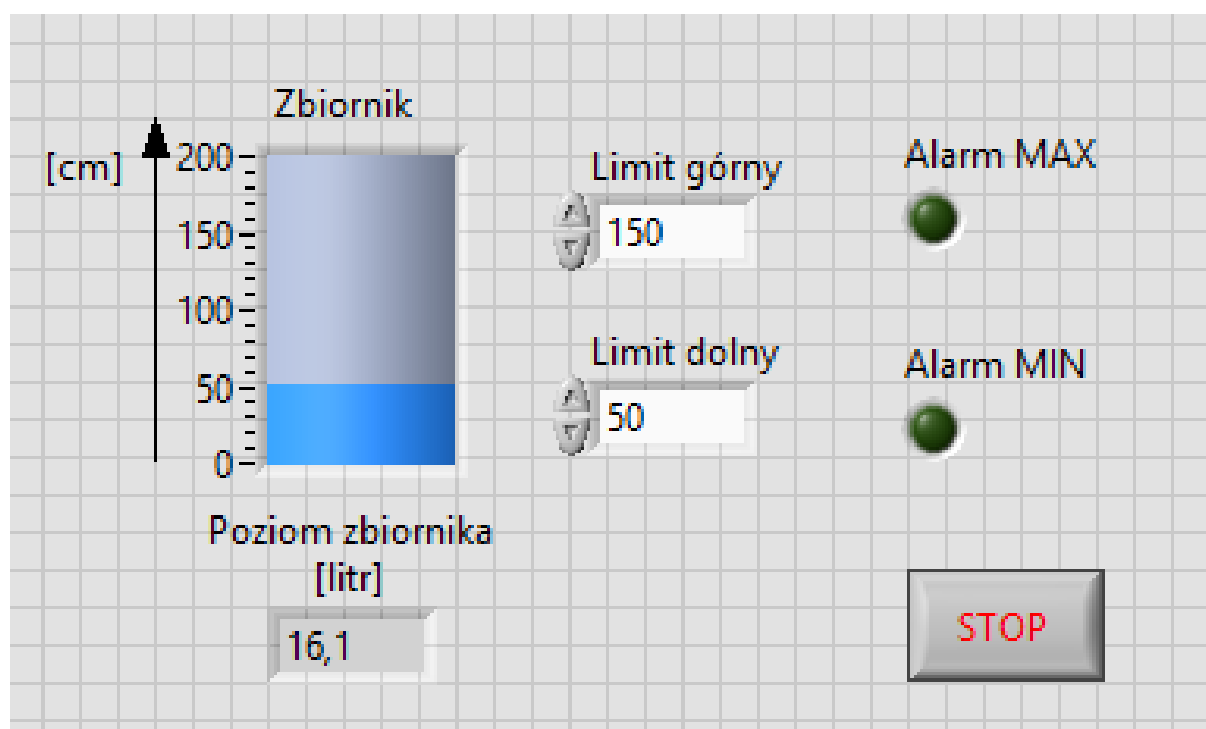
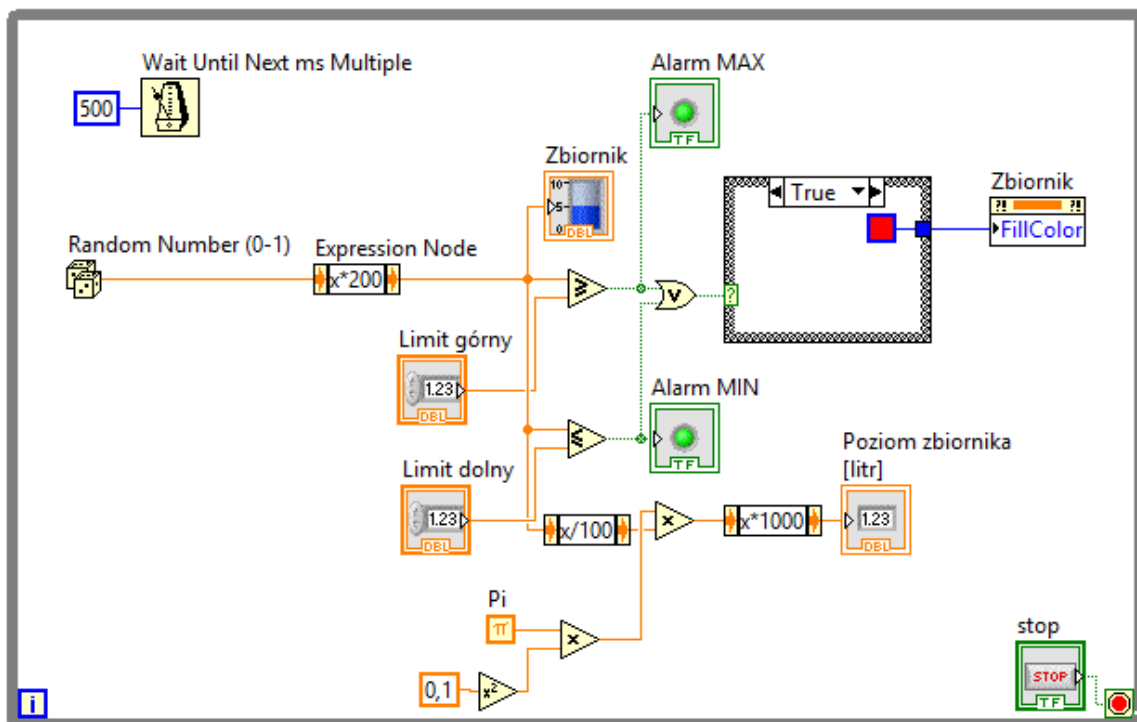


Diagram programu:



Ciągi tekstowe – slajdy: 171 do 179.

Celem modułu jest zaprezentowanie graficznej reprezentacji operacji związanych z przetwarzaniem danych tekstowych w środowisku LabVIEW.

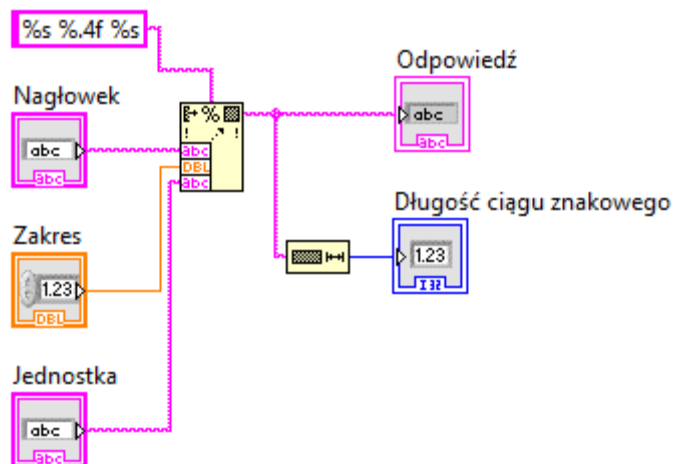
Ćwiczenie 15 – slajdy 178 do 179:

W ramach ćwiczenia zaprezentowana zostanie formatowanie danych wyjściowych pod postacią ciągu tekstowego.

Interfejs użytkownika:

Nagłówek	Odpowiedź
SOUR:VOLT:AMP	SOUR:VOLT:AMP 10,0000 [V]
Pole string control	Pole string indicator
Zakres	Długość ciągu znakowego
10	25
Pole numeric control	Pole numeric indicator
Jednostka	
[V]	
Pole string control	

Diagram programu:



Pojęcie funkcji w LabVIEW– slajdy: 180 do 189.

Celem modułu jest zaprezentowanie sposobu budowy funkcji oraz ich użycia w środowisku LabVIEW.

Ćwiczenie 16 – slajdy 183 do 145:

W ramach ćwiczenia zaprezentowana zostanie budowa programu przeliczającego stopnie Celsjusza na Fahrenheita. W ramach ćwiczenia zaprezentowany zostanie sposób budowy i wykorzystania funkcji:

Interfejs użytkownika funkcji:

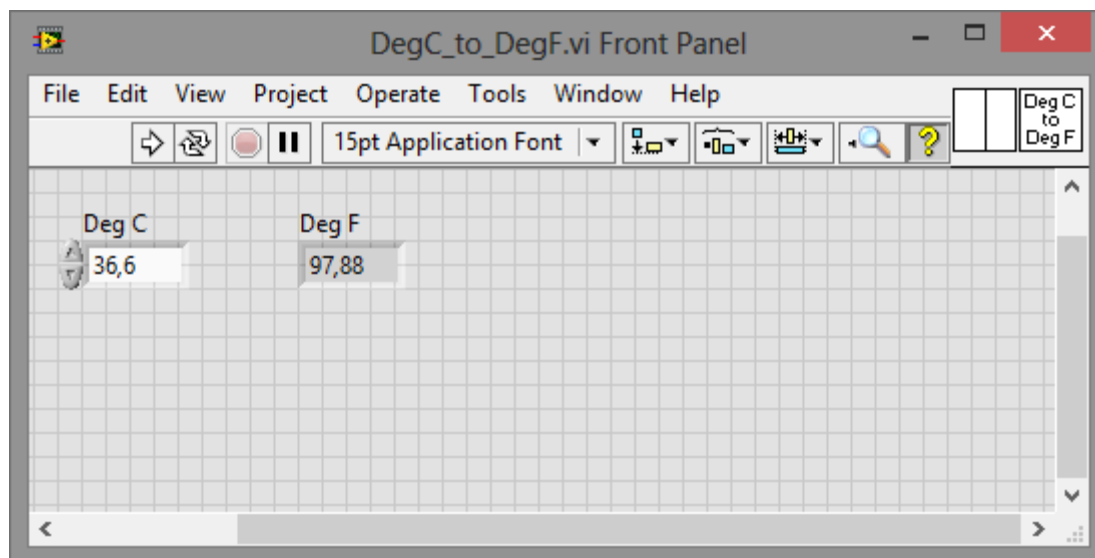
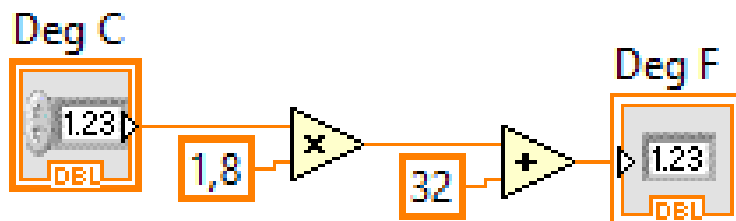


Diagram funkcji:



Interfejs użytkownika aplikacji korzystającej z funkcji:

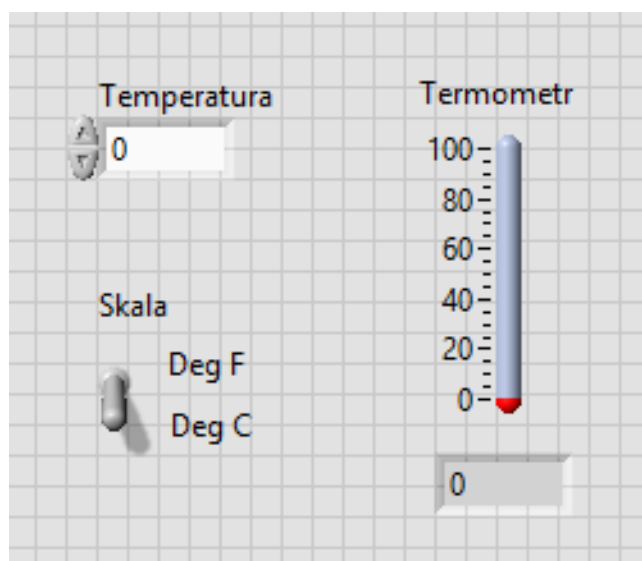
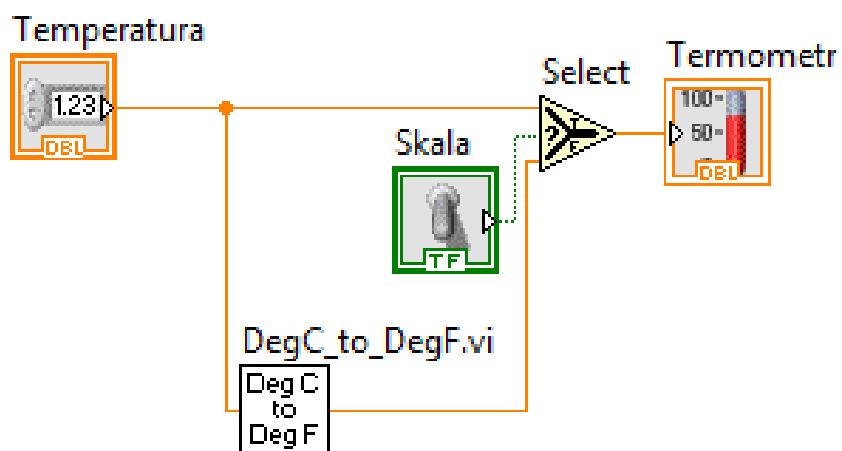


Diagram programu:



Ćwiczenie 17 – slajdy 190 do 193:

W ramach ćwiczenia jest zaprezentowanie sposobu budowy funkcji oraz jej wykorzystanie.

Interfejs użytkownika:

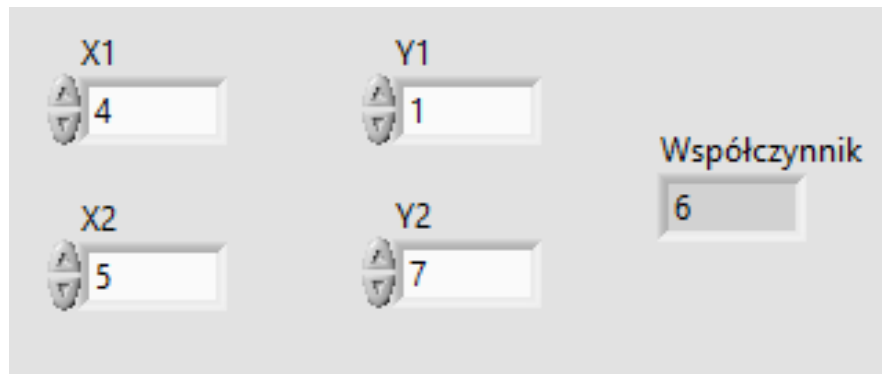
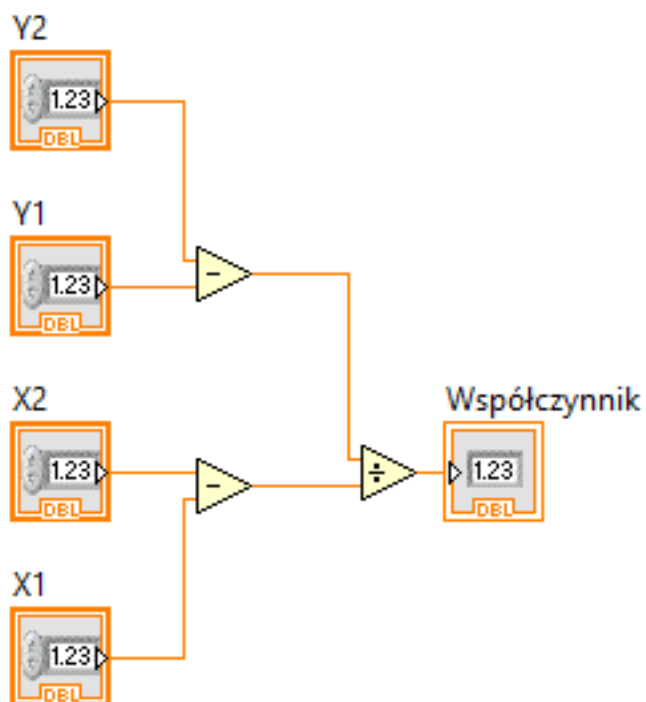


Diagram programu:



Generacja przebiegów i ich wyświetlanie – slajdy: 194 do 210.

Celem modułu jest omówienie i zaprezentowanie sposobu budowy aplikacji generującej przebiegi oraz sposobu ich wyświetlania w kontrolkach dostępnych w środowisku LabVIEW.

Ćwiczenie 18 – slajdy 197 do 198:

W ramach ćwiczenia zaprezentowany zostanie sposób generacji przebiegu losowego z funkcją uśredniania. Demonstracja wykresu Waveform Chart

Interfejs użytkownika:

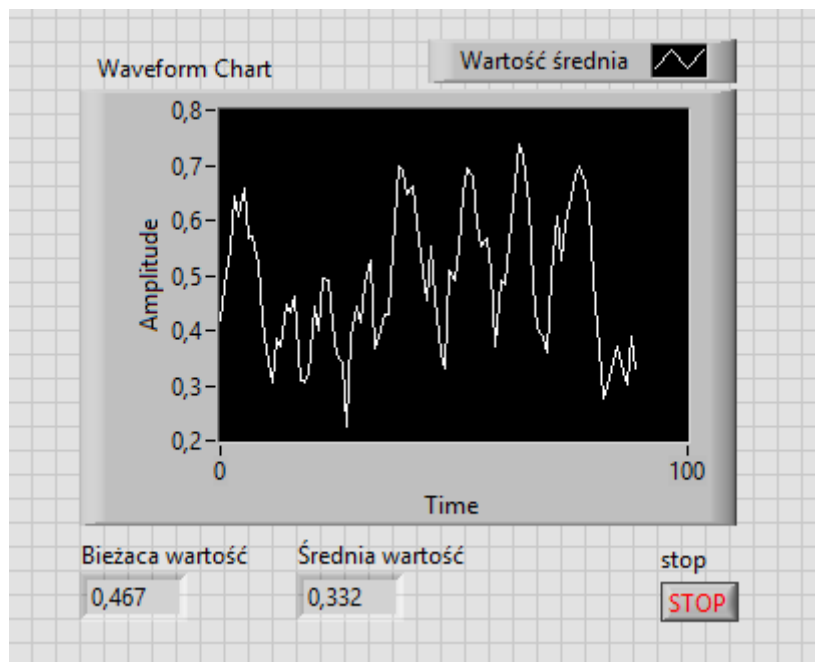
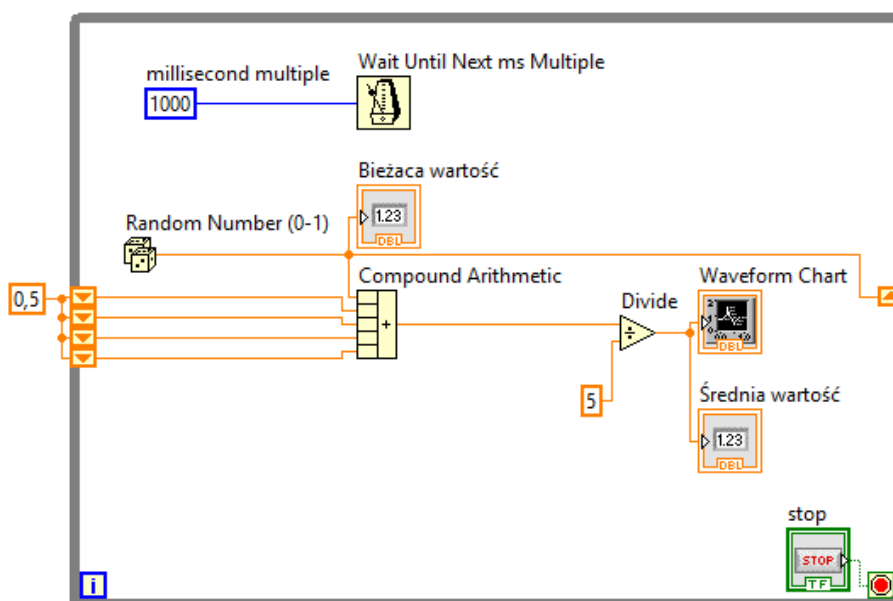


Diagram programu:



Ćwiczenie 19 – slajdy 199 do 200:

W ramach ćwiczenia zaprezentowany zostanie sposób generacji przebiegu okresowego opisanego funkcją matematyczną oraz wyświetlanie go na wykresie czasowym. Demonstracja wykresu Waveform Chart

Interfejs użytkownika:

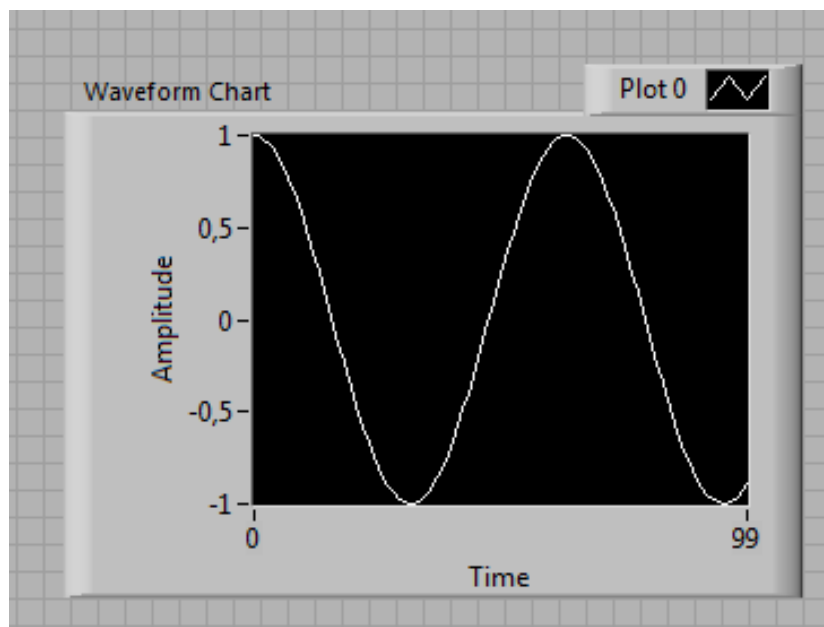
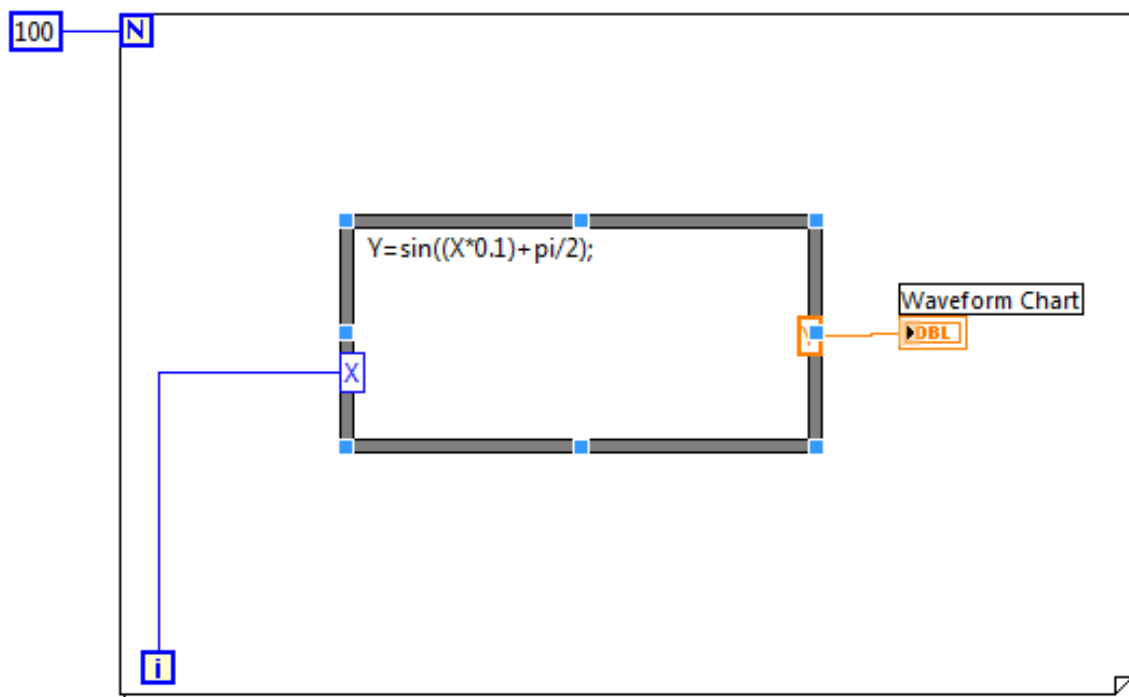


Diagram programu:



Ćwiczenie 20 – slajdy 201 do 202:

W ramach ćwiczenia zaprezentowany zostanie sposób generacji wielu przebiegów opisanych funkcjami matematycznymi oraz wyświetlanie ich na wykresie. Demonstracja wykresu Waveform Graph

Interfejs użytkownika:

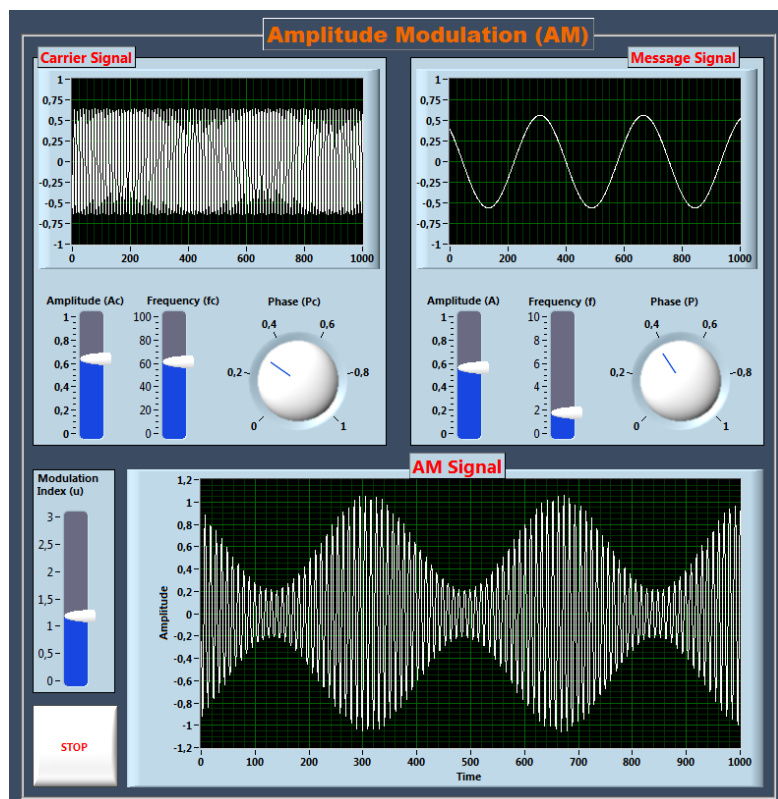
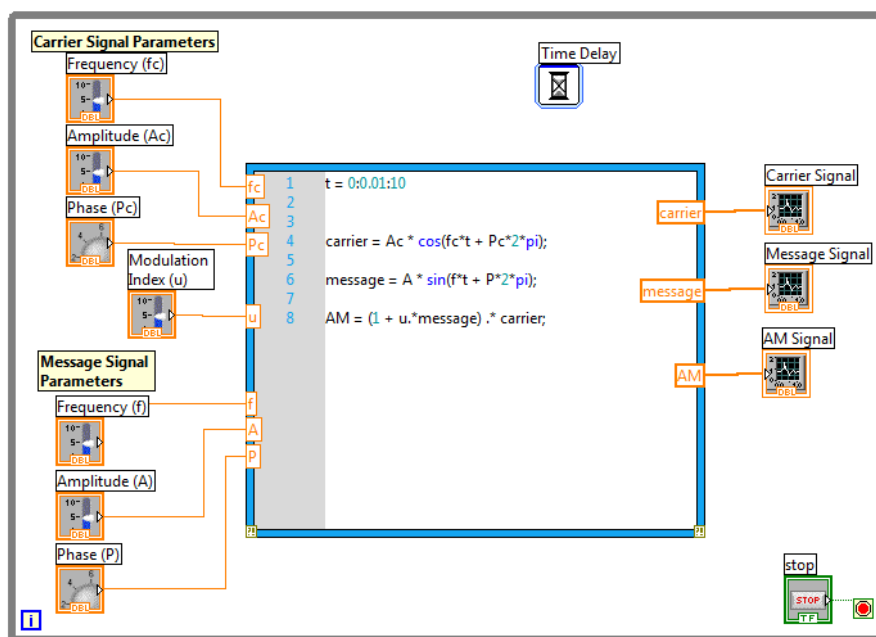


Diagram programu:



Ćwiczenie 21 – slajdy 201 do 202:

W ramach ćwiczenia zaprezentowany zostanie sposób generacji wielu przebiegów z wykorzystaniem gotowych bloków do generacji sygnałów okresowych oraz wyświetlanie ich na wykresie.

Interfejs użytkownika:

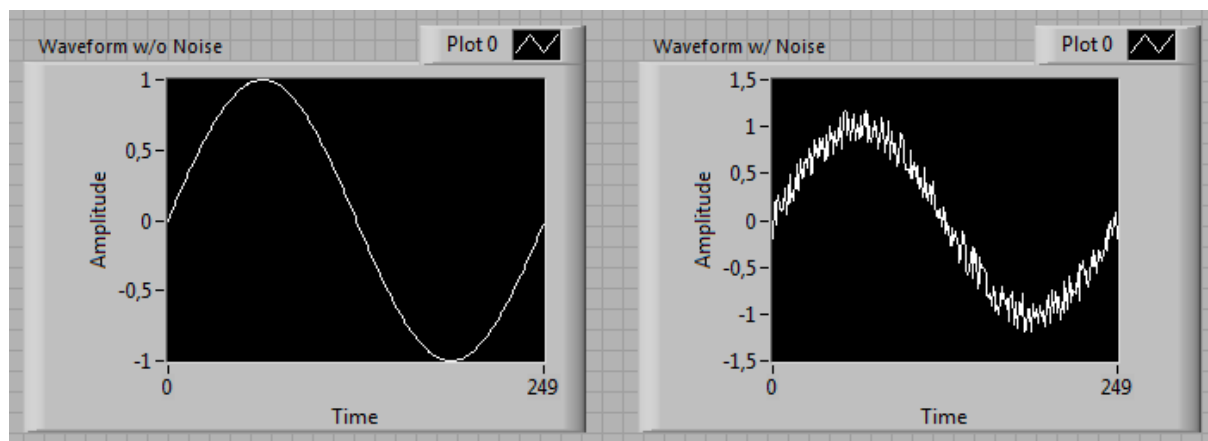
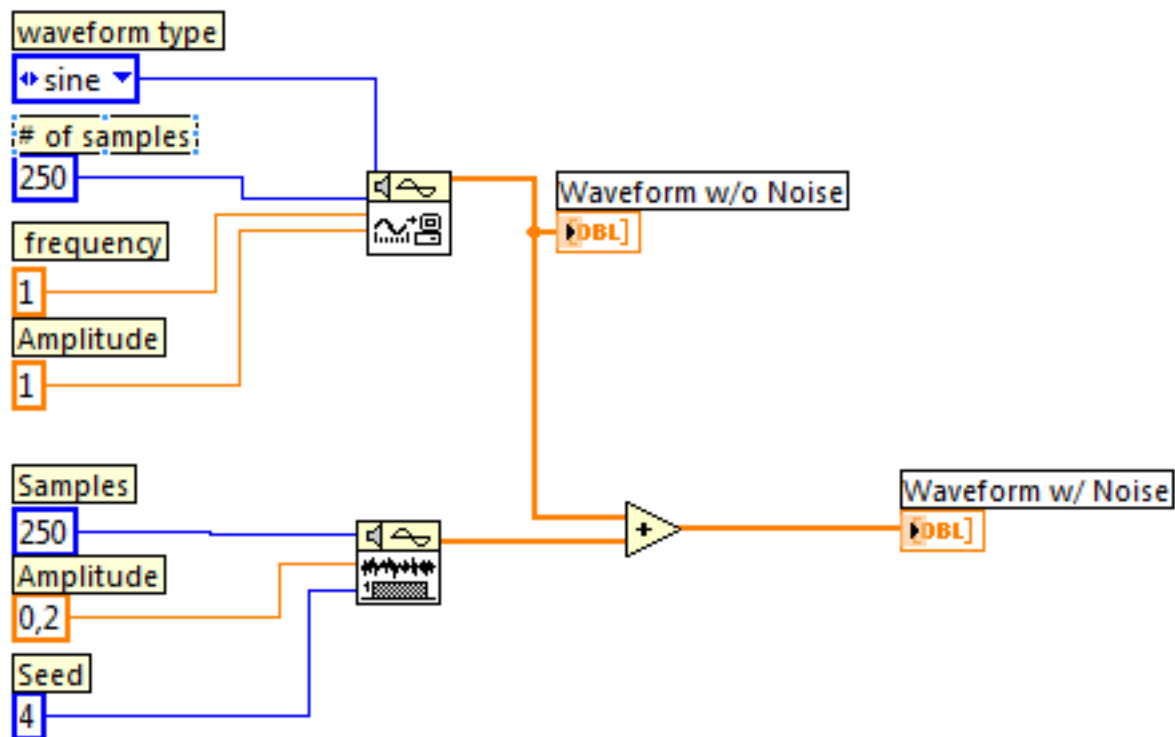


Diagram programu:



Ćwiczenie 22 – slajd 205:

W ramach ćwiczenia zaprezentowana zostanie generacja sygnału i wyświetlanie wielu przebiegów z wykorzystaniem kontrolki XY Graph.

Interfejs użytkownika:

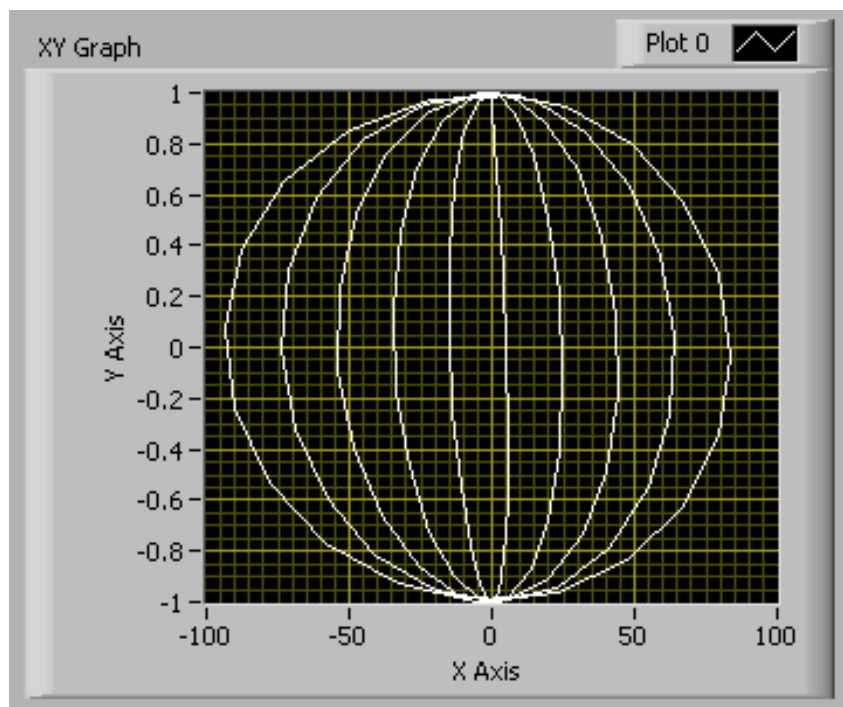
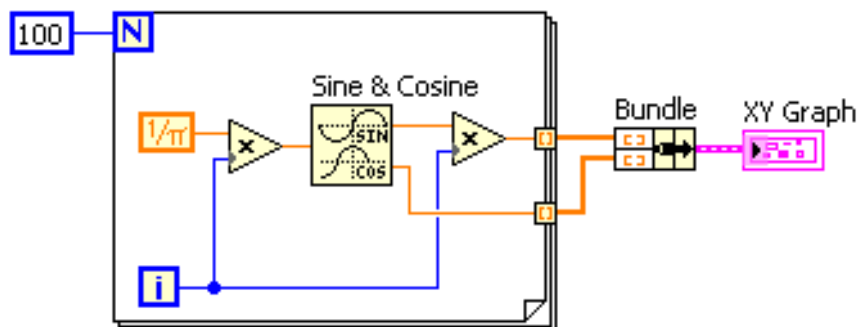


Diagram programu:



Ćwiczenie 23 – slajdy 206 do 207:

W ramach ćwiczenia zaprezentowana zostanie generacja sygnału i wyświetlanie wielu przebiegów z wykorzystaniem kontrolki XY Graph oraz sposób skalowania przebiegu – wyskalowanie osi X.

Interfejs użytkownika:

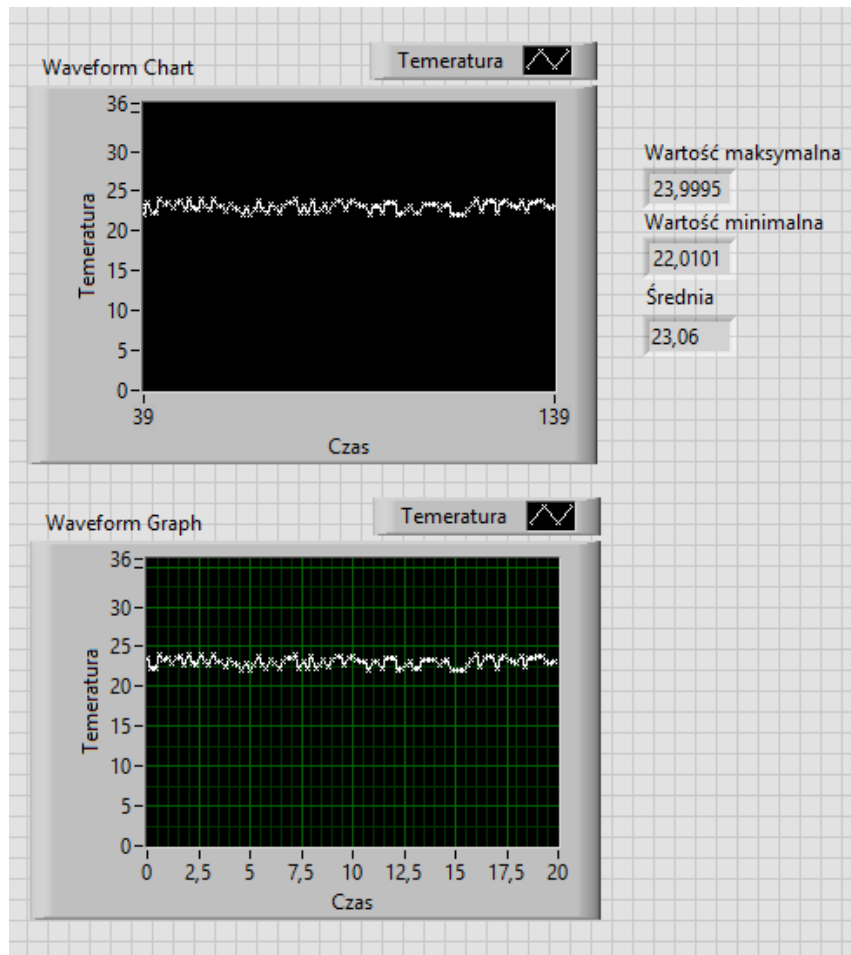
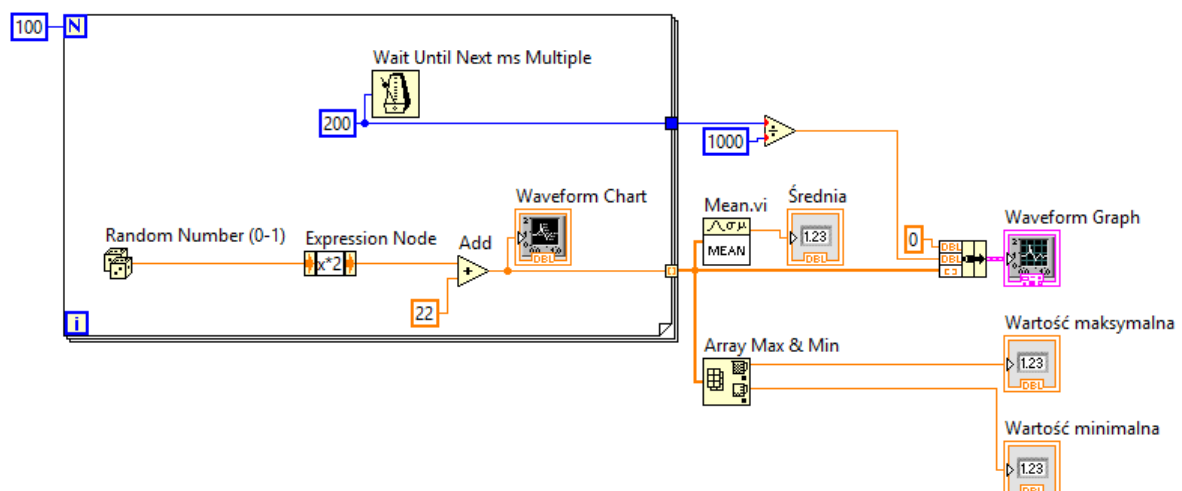


Diagram programu:



Ćwiczenie 24 – posumowanie modułu – slajdy 144 do 145:

W ramach ćwiczenia zaprezentowana zostanie generacja sygnału i wyświetlanie wielu przebiegów z wykorzystaniem kontrolki XY Graph wraz ze skalowaniem przebiegu.

Interfejs użytkownika:

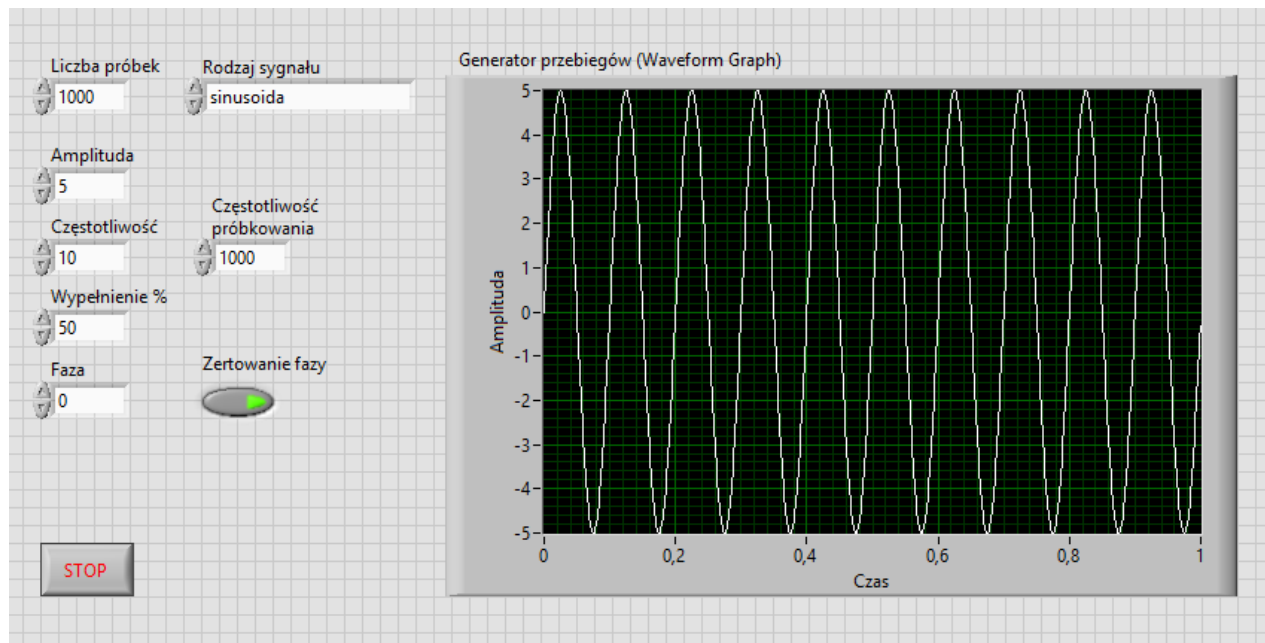
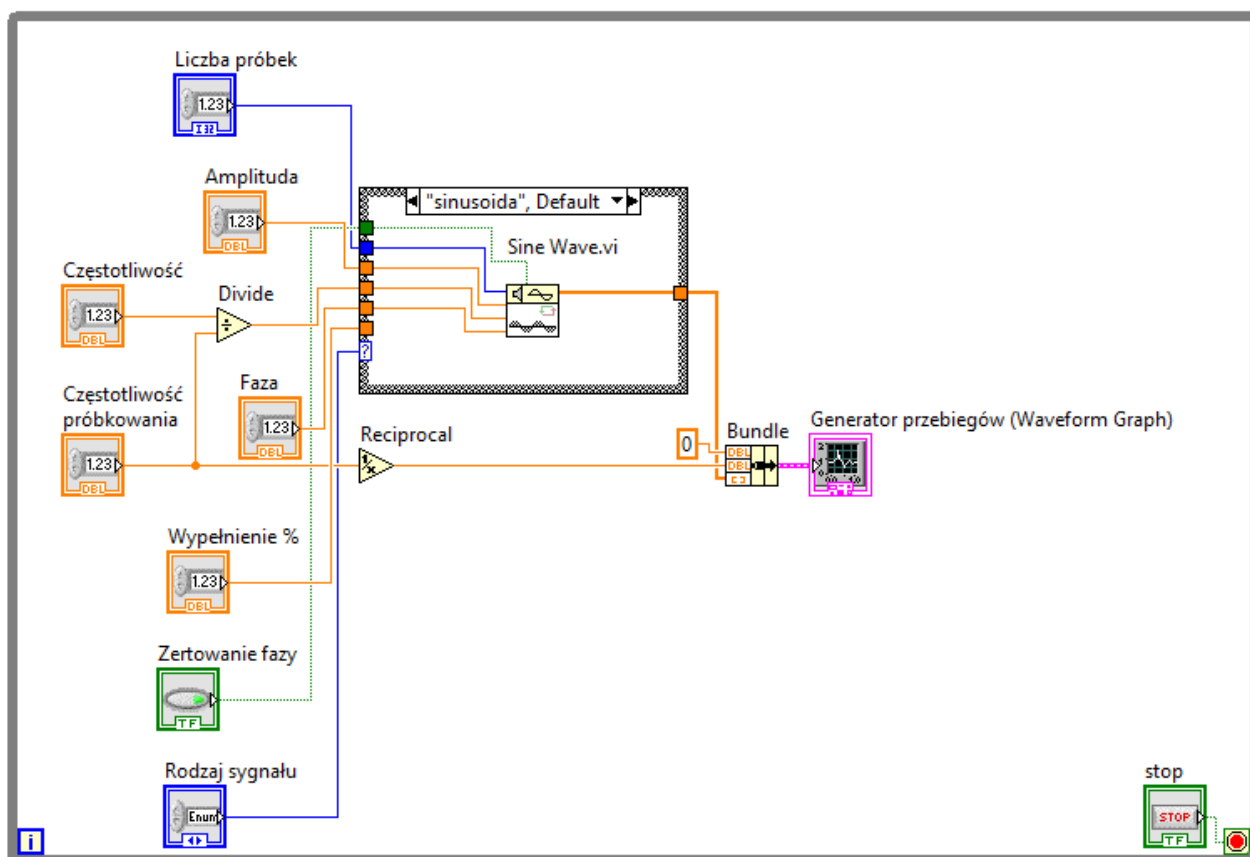


Diagram programu:



Komunikacja sieciowa w LabVIEW – slajdy: 211 do 262.

Celem modułu jest omówienie i zaprezentowanie sposobu budowy aplikacji wykorzystujących komunikację sieciową. W ramach części teoretycznej zostaną przedstawione następujące zagadnienia wraz z przykładami użycia:

- Komunikacja na „niskim poziomie”
- Komunikacja wysokiego poziomu
- Protokoły niskiego poziomu:
 - Protokół UDP
 - Protokół TCP
 - TCP – Simple TCP/IP Messaging Protocol
 - Bluetooth
- Protokoły wysokiego poziomu
 - Network-Published Shared Variable
 - Datasocket server
 - VI Server
 - WebServices

LabVIEW krok po kroku– slajdy: 263 do 315.

Celem modułu jest zapoznanie się z sposobem budowania aplikacji z wykorzystaniem graficznego języka programowania.

Budowanie DLL w LabVIEW – slajdy: 264 do 315.

Celem modułu jest zapoznanie się z sposobem budowania aplikacji z wykorzystaniem graficznego języka programowania do szybkiego zakodowania obliczeń czy skomplikowanych algorytmów i użycie we własnych aplikacjach napisanych w innych językach programowania.

Ćwiczenie 25 – slajdy 265 do 272:

W ramach ćwiczenia zaprezentowany sposób budowania biblioteki DLL na przykładzie funkcji do konwersji stopni F na C.

Interfejs użytkownika:

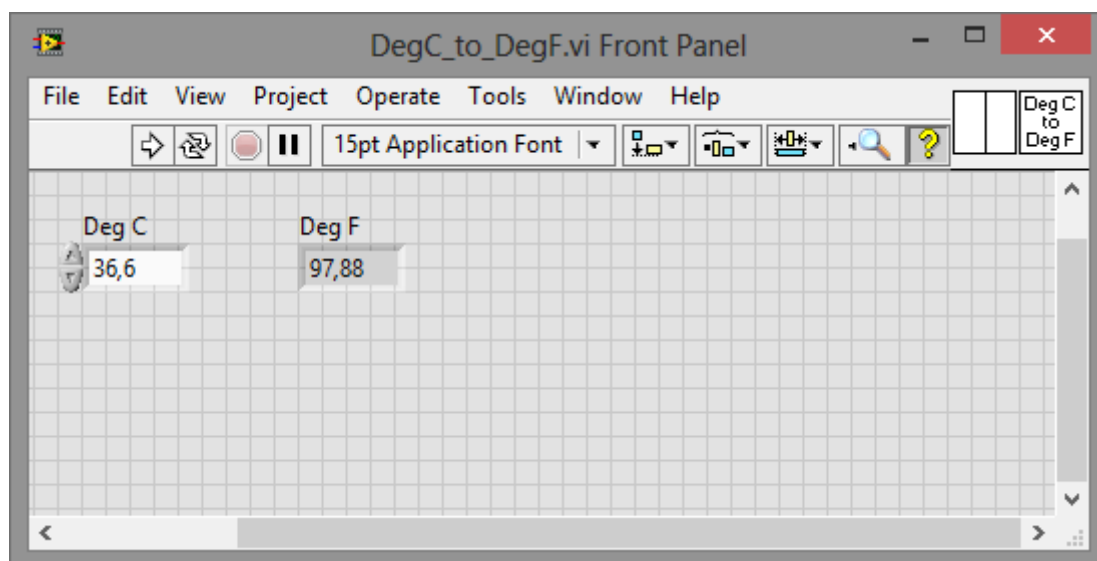
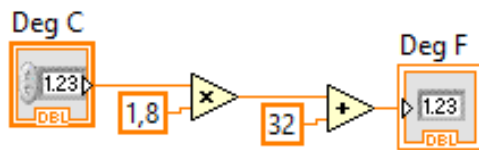
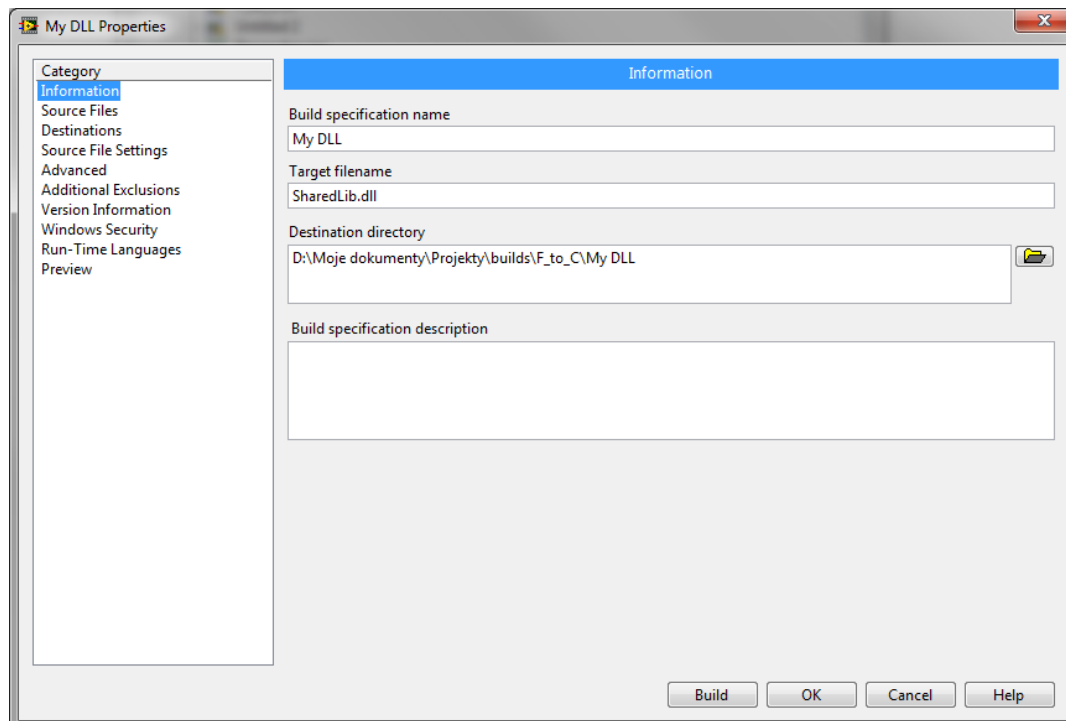
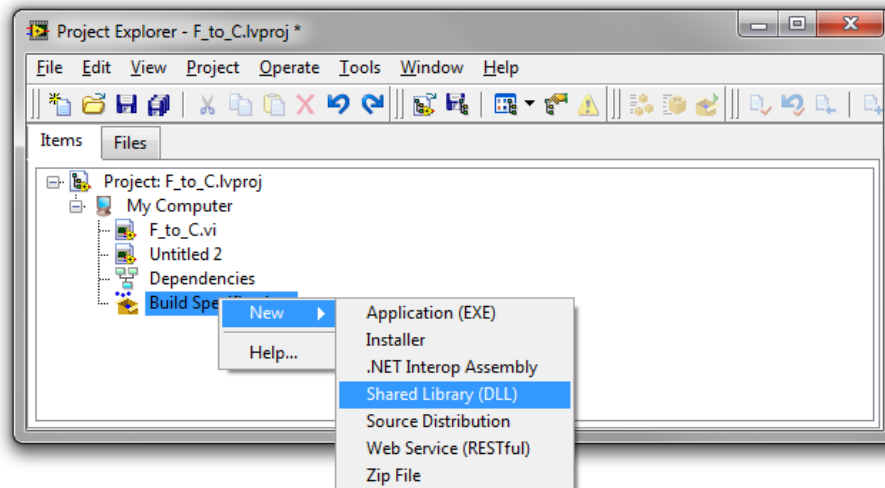


Diagram programu:



Konfiguracja kreatora DLL



Ćwiczenie 26 – slajdy 274 do 275:

Wyświetlanie wielu przebiegów jednocześnie.

Interfejs użytkownika:

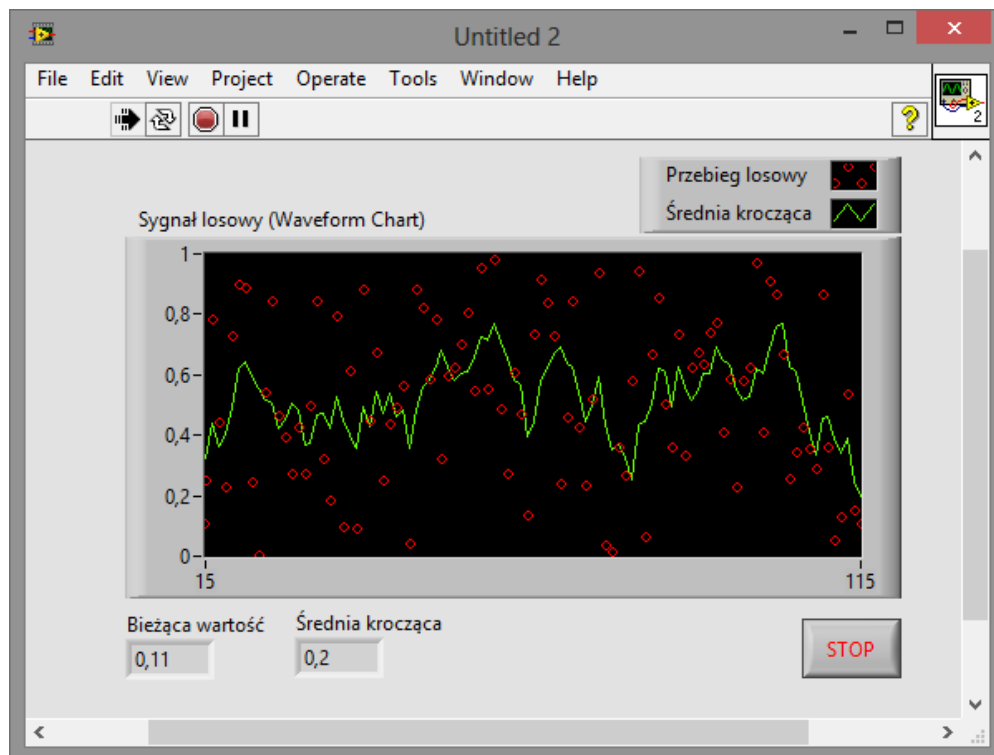
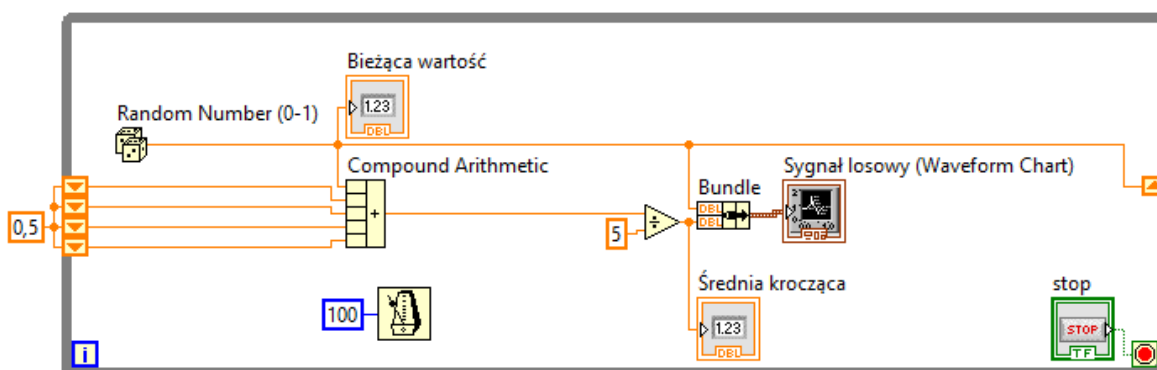


Diagram programu:



Ćwiczenie 28 – slajdy 144 do 145:

Aplikacja generującą losowe dane wraz z systemem analizy i alarmów.

Interfejs użytkownika:

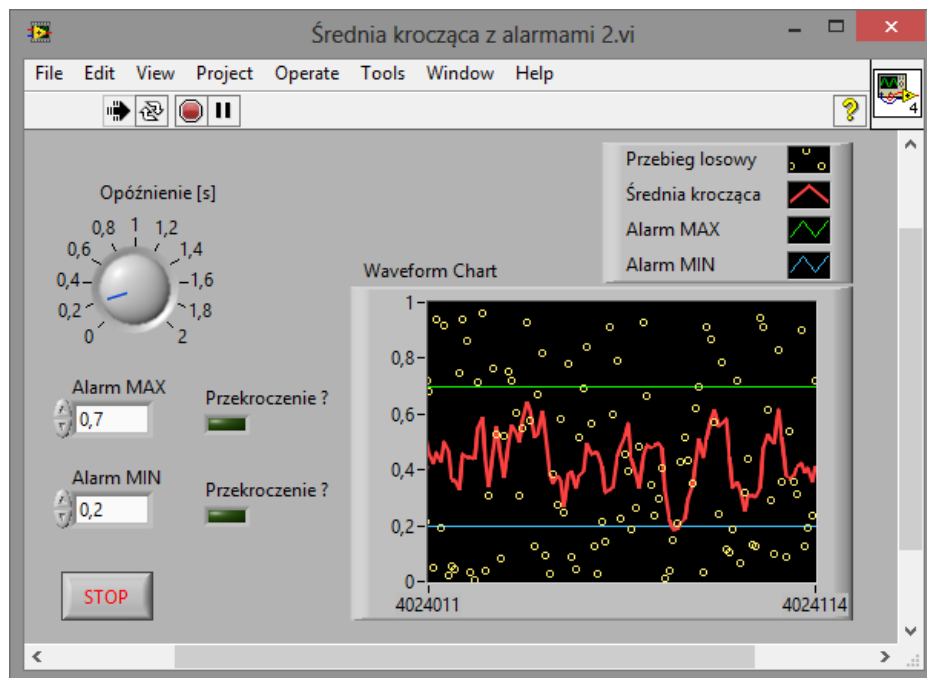
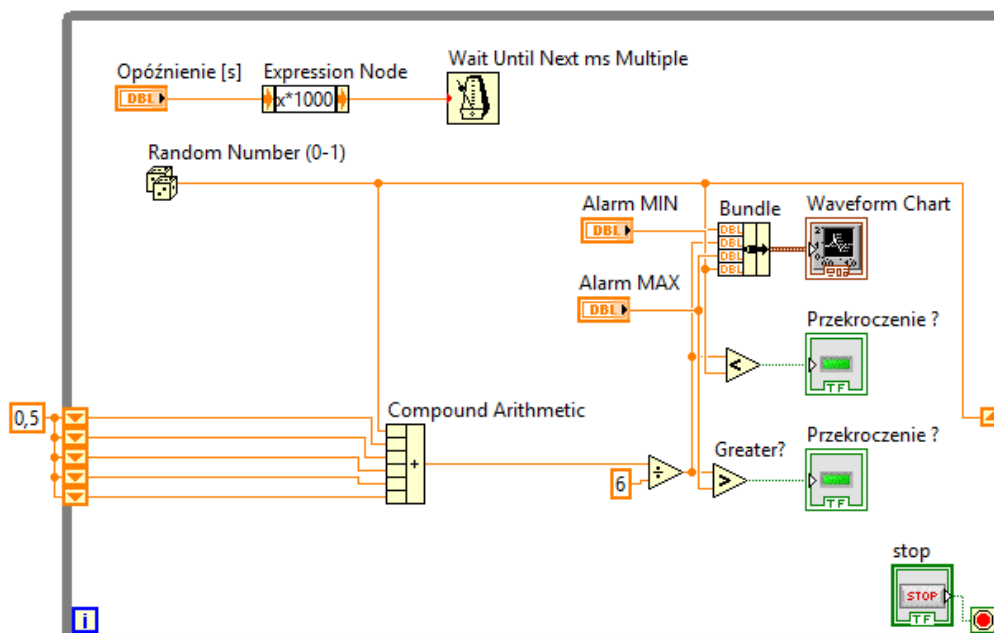


Diagram programu:



Ćwiczenie 29 – slajdy 278 do 280:

Aplikacja wykorzystująca mechanizm kolejek oraz trzy równoległe pętle:

- pierwsza będąca generatorem danych
- druga będąca odbiorcą danych
- trzecia będąca monitorem danych

Interfejs użytkownika:

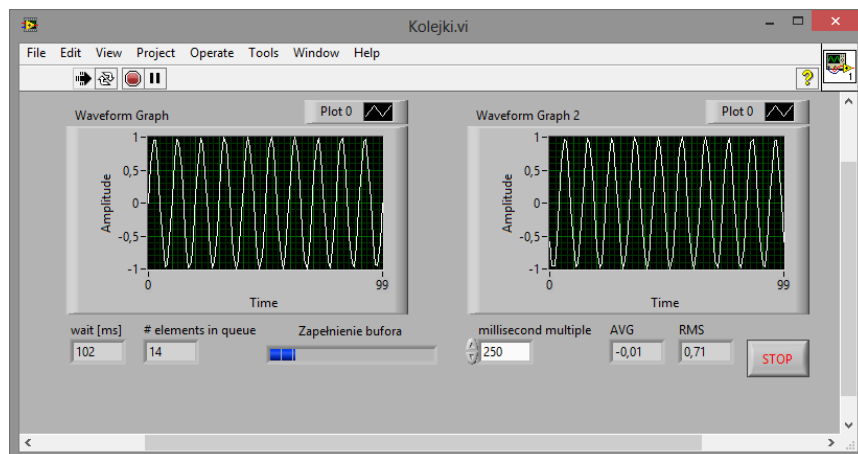
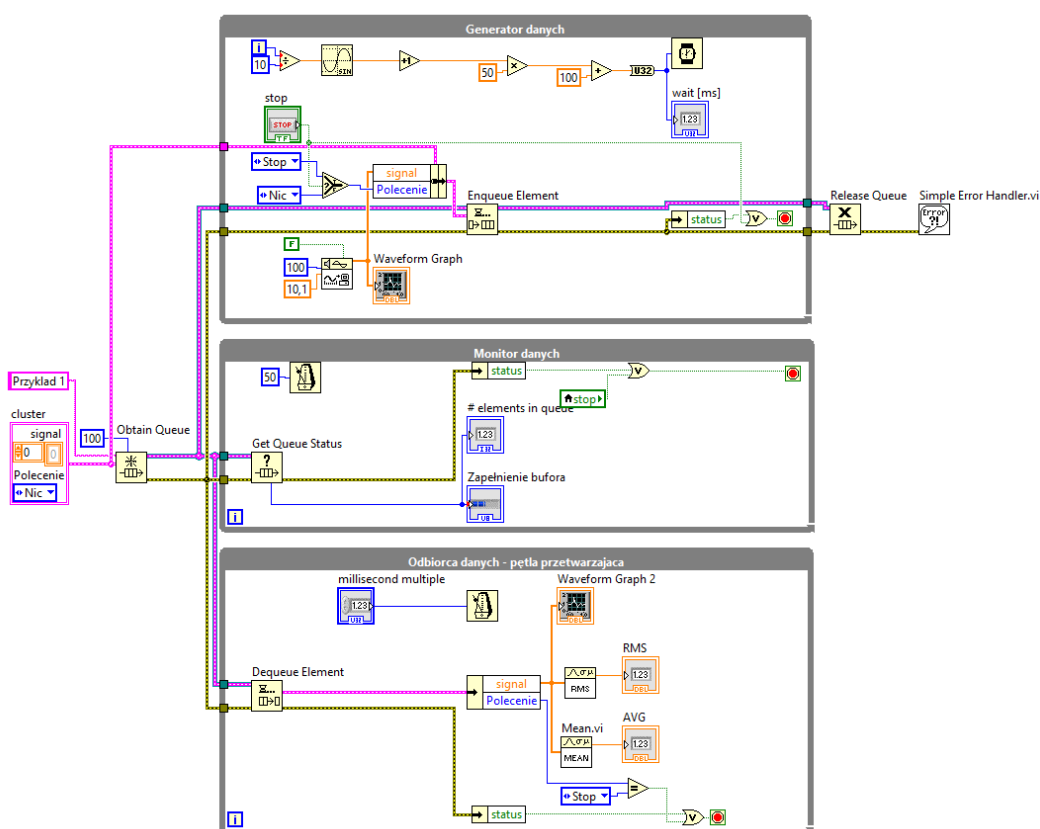


Diagram programu:



Komunikacja sieciowa w LabVIEW – slajdy: 281 do 315.

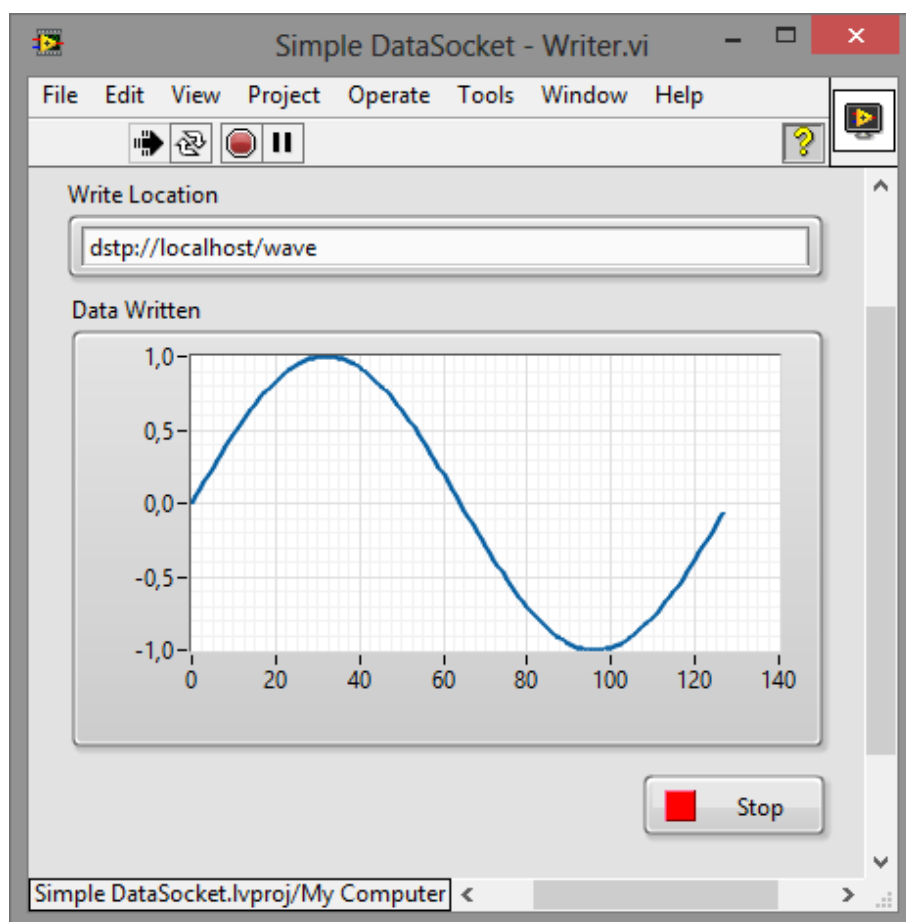
Celem modułu jest zapoznanie się z sposobem budowania aplikacji z wykorzystaniem graficznego języka programowania.

Ćwiczenie 30 – slajdy 281 do 285:

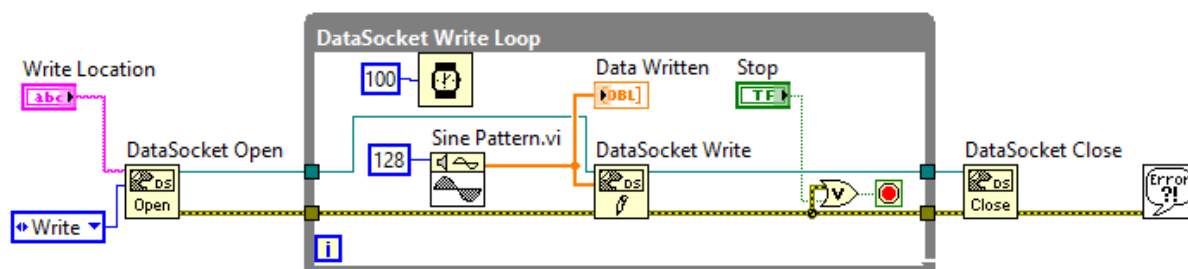
Aplikacja klient – serwer z DSS – zakres ćwiczenia:

- Budowa aplikacji wysyłającej wygenerowany przebieg sinusoidalny z wykorzystaniem technologii DataSocket Server
- Budowa aplikacji odbierającą dane z wykorzystaniem technologii DataSocket Server

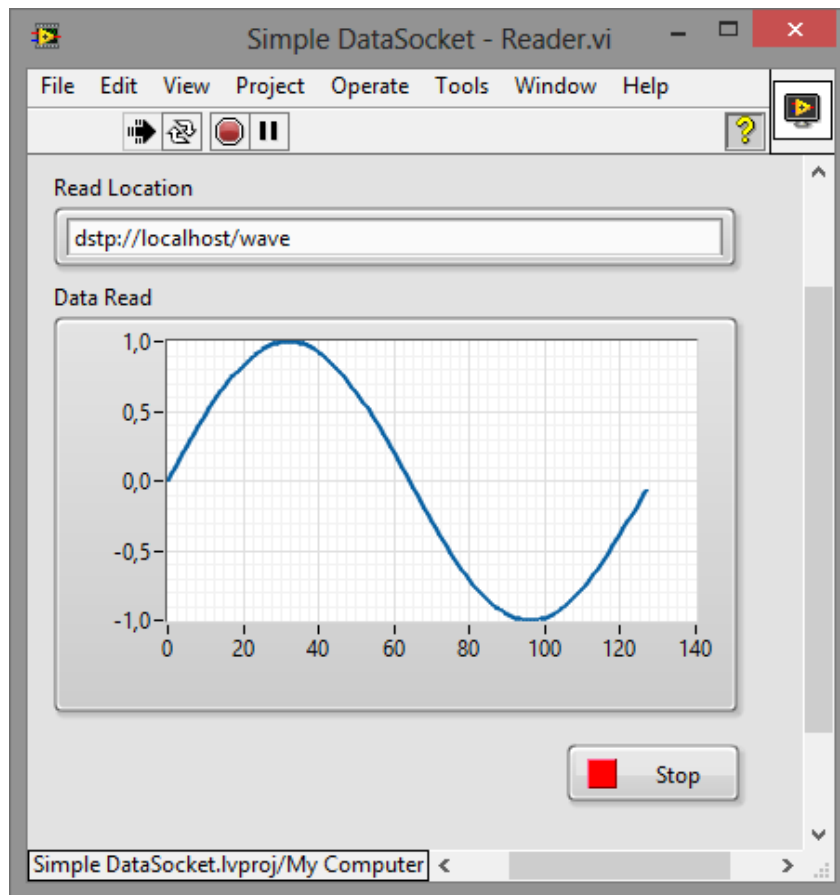
Nadawanie danych - interfejs użytkownika:



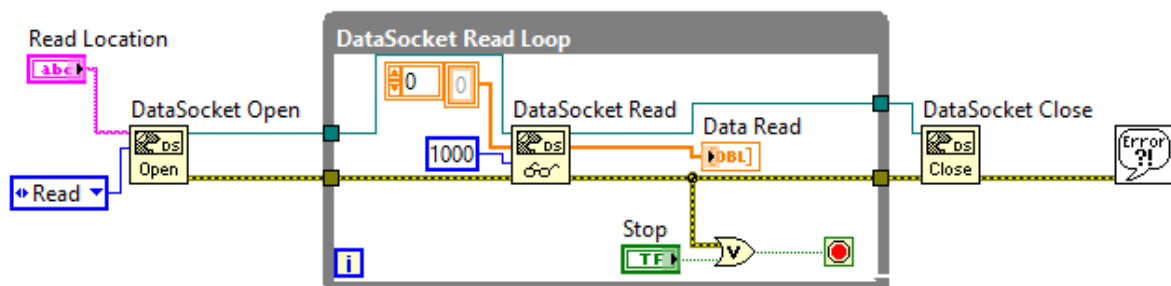
Nadawanie danych - diagram programu:



Odbieranie danych - interfejs użytkownika:



Odbieranie danych - diagram programu:

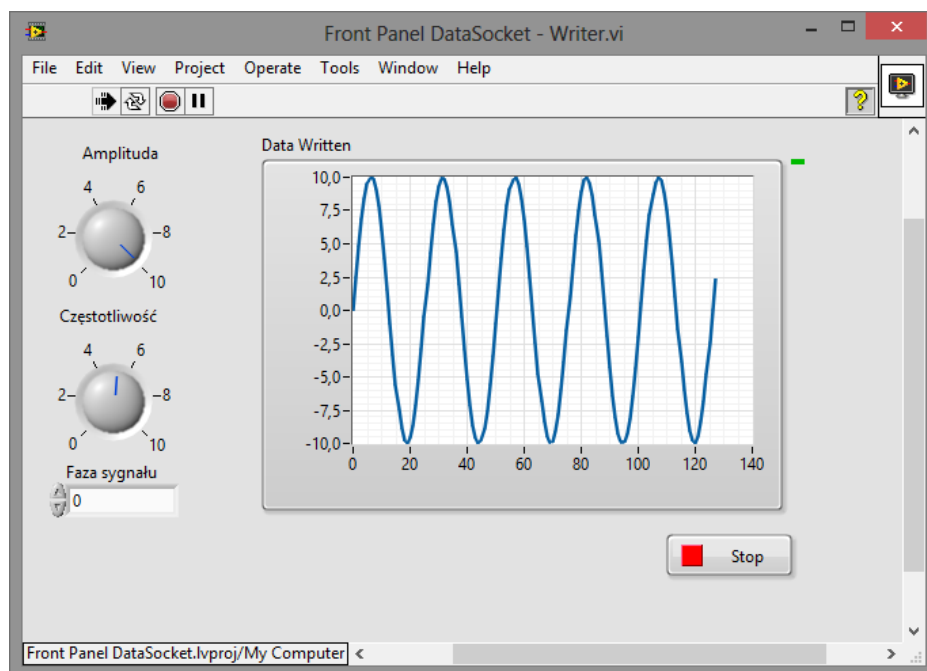


Ćwiczenie 30 – slajdy 286 do 288:

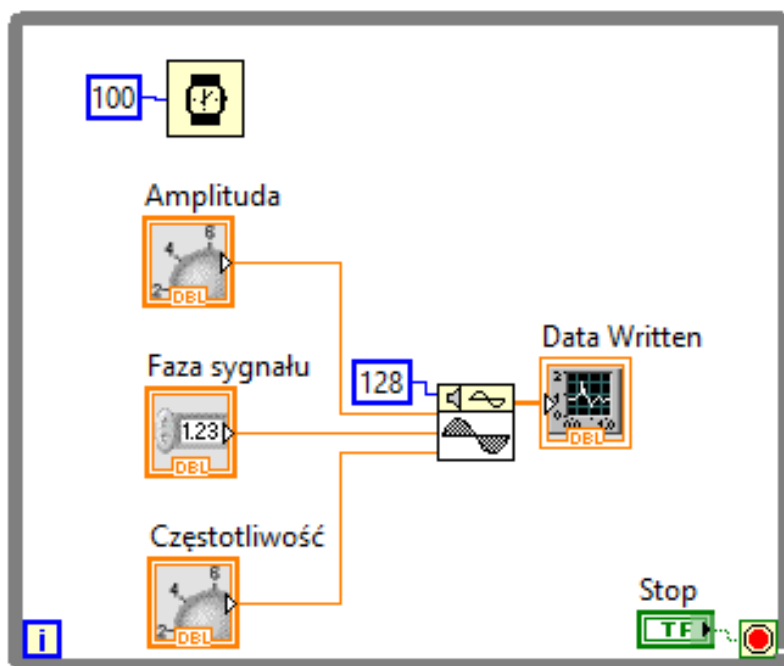
Aplikacja klient – serwer z DSS – zakres ćwiczenia:

- Budowa aplikacji wysyłającej wygenerowany przebieg sinusoidalny z wykorzystaniem technologii DataSocket Server oraz VI Server
- Aplikacja ma mieć możliwość regulacji, częstotliwości, fazy sygnału oraz amplitudy
- Budowa aplikacji odbierającej dane z wykorzystaniem technologii DataSocket Server oraz VI Server

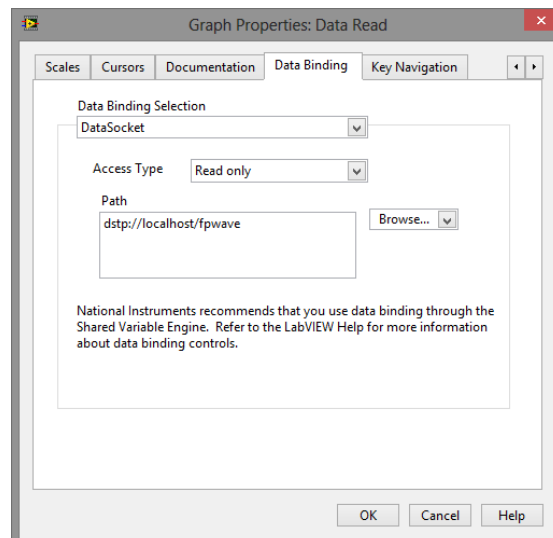
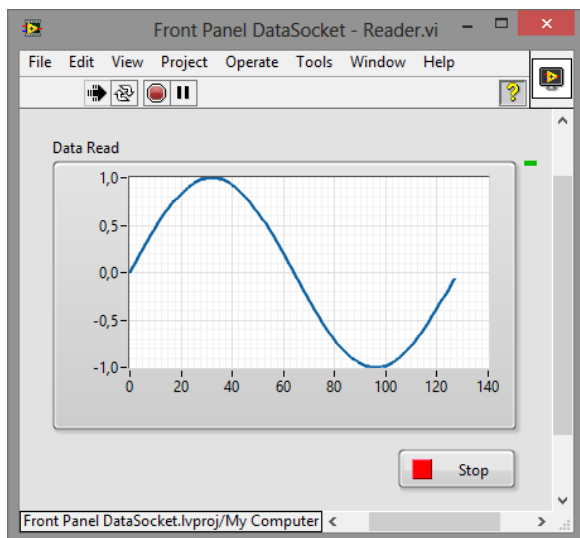
Nadawanie danych - interfejs użytkownika:



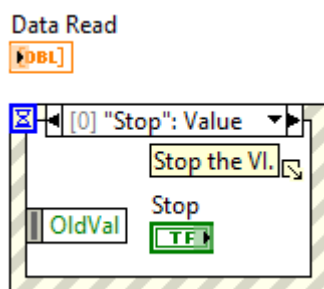
Nadawanie danych - diagram programu:



Odbieranie danych - interfejs użytkownika:



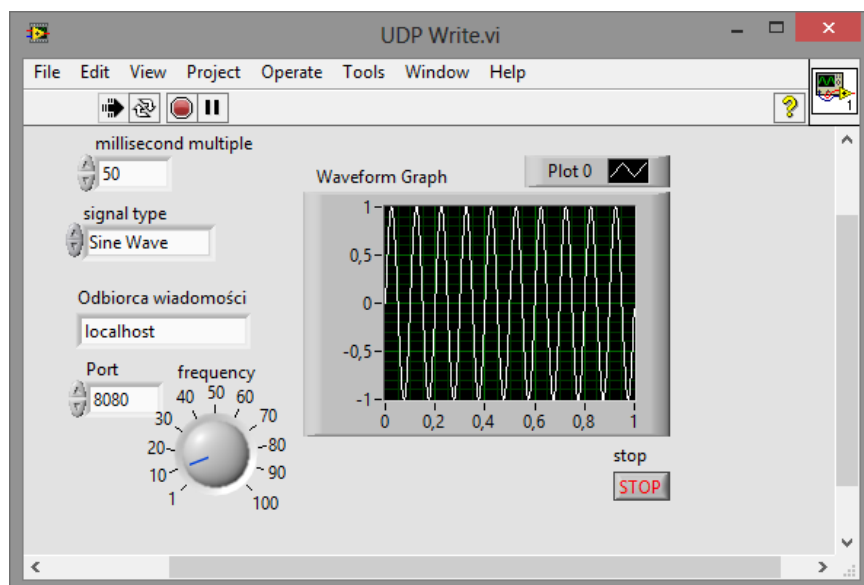
Odbieranie danych - diagram programu:



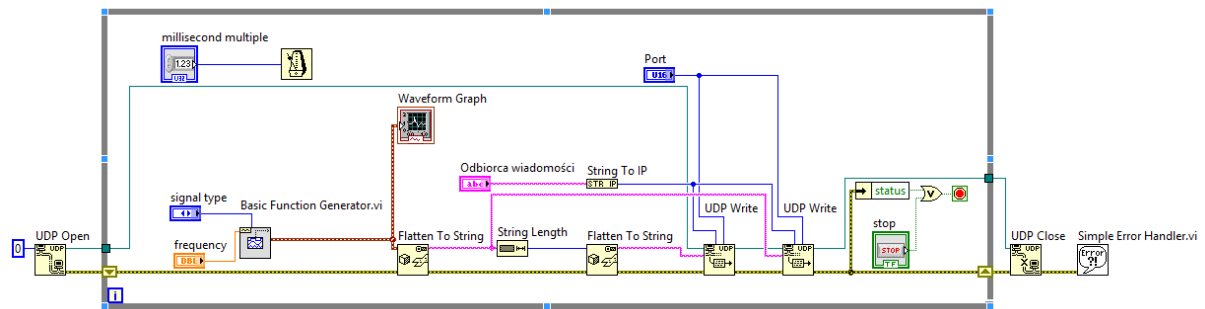
Ćwiczenie 31 – slajdy 289 do 292:

Aplikacja klient – serwer wykorzystująca protokół UDP.

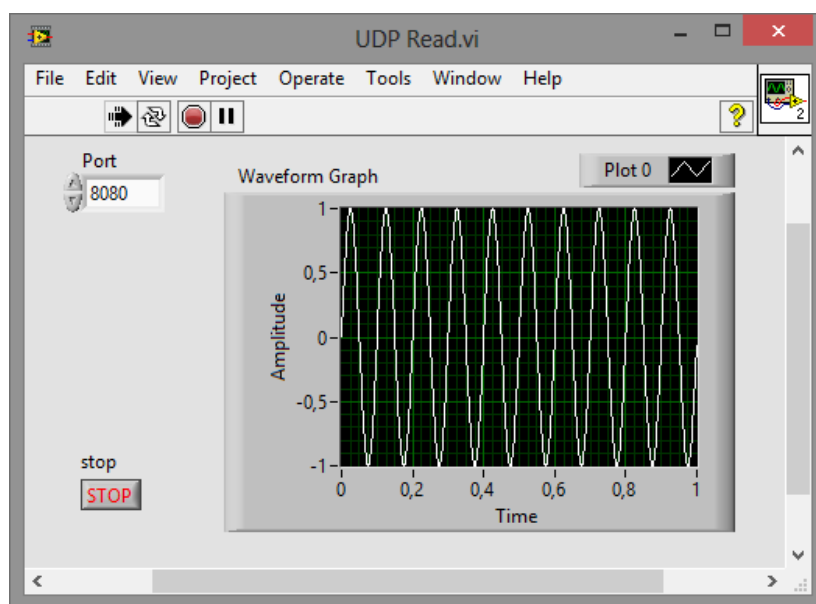
Nadawanie danych - interfejs użytkownika:



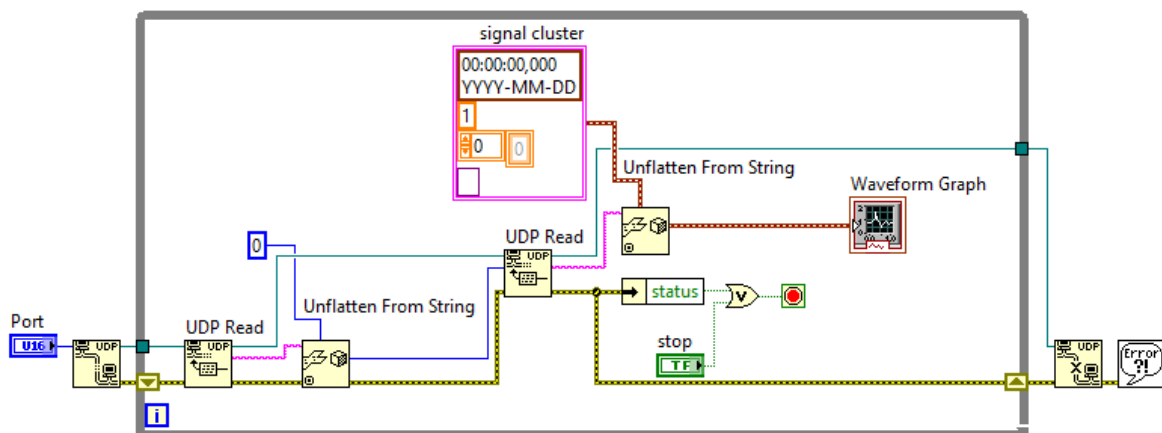
Nadawanie danych - diagram programu:



Odbieranie danych - interfejs użytkownika:



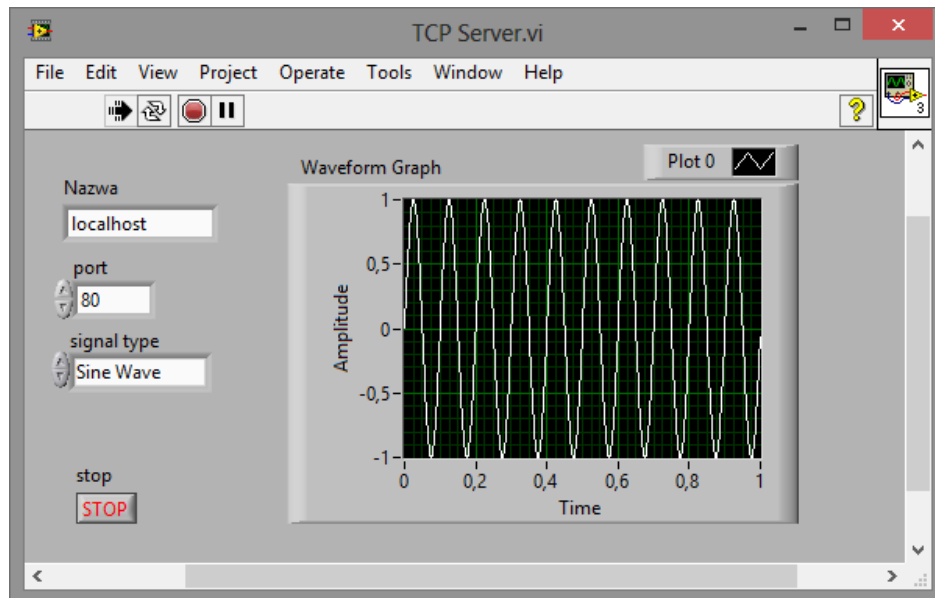
Odbieranie danych - diagram programu:



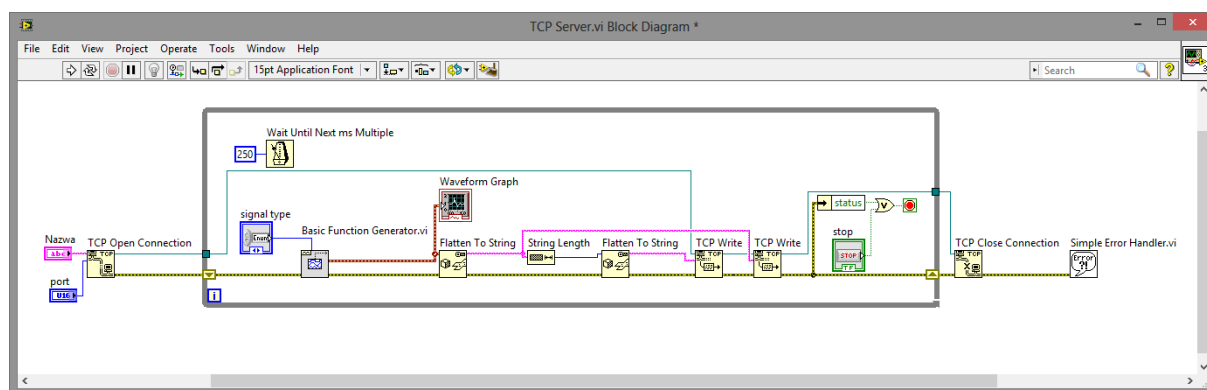
Ćwiczenie 32 – slajdy 293 do 292:

Aplikacja klient – serwer wykorzystująca protokół TCP.

Nadawanie danych - interfejs użytkownika:



Nadawanie danych - diagram programu:

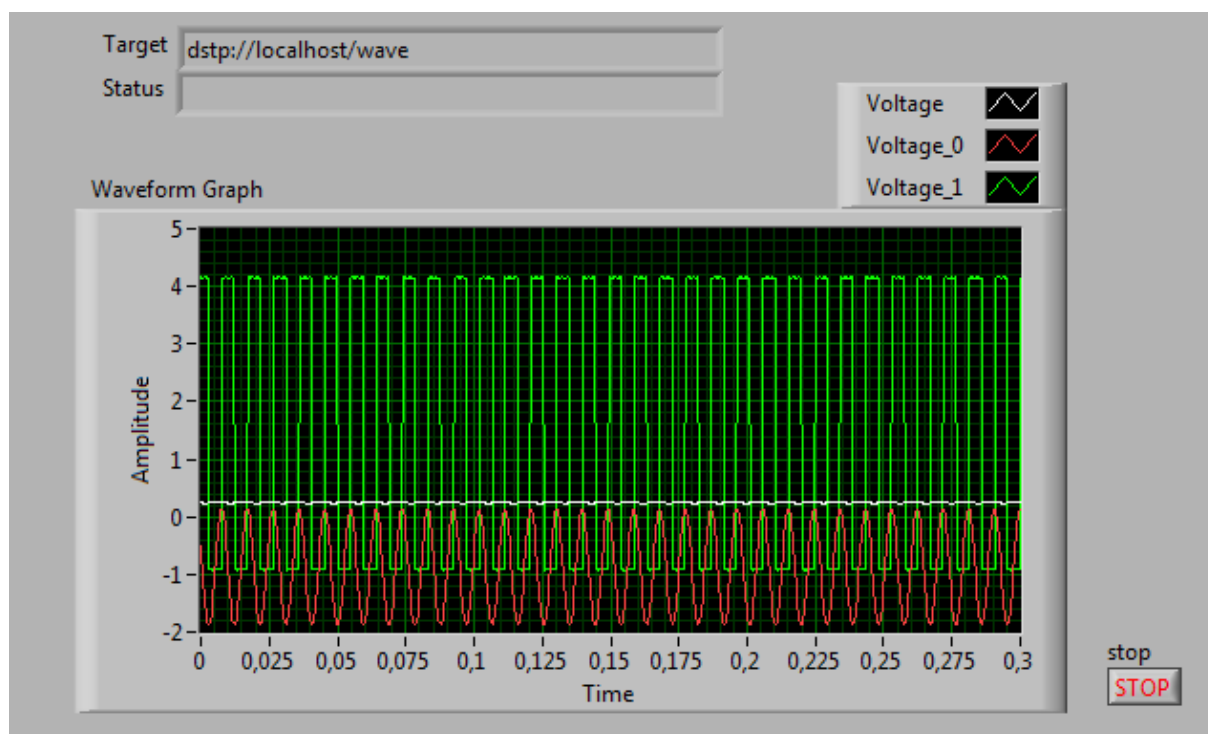


Ćwiczenie 33 – slajdy 295 do 301:

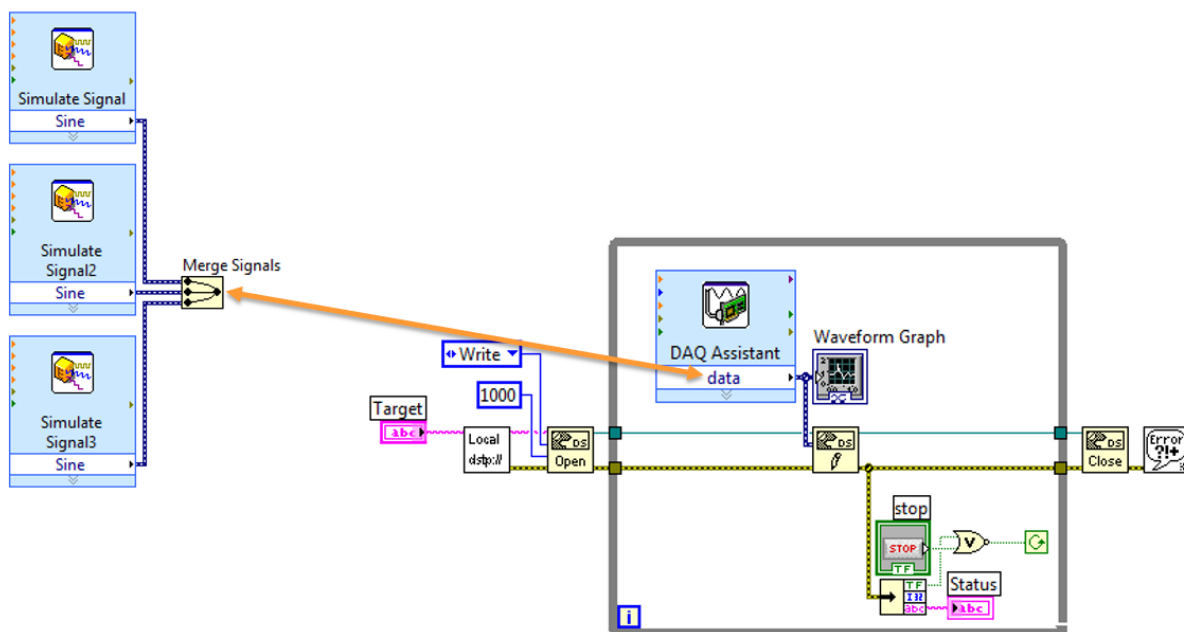
Aplikacja klient – serwer z DSS – zakres ćwiczenia:

- Budowa aplikacji wysyłającej odczytane przebiegi z karty zbierania danych lub z symulatora (użyć funkcji simulate signal i wygenerować 3 przebiegi) za pośrednictwem technologii DataSocket Server
- Budowa aplikacji odbierającej dane z wykorzystaniem technologii DataSocket Server

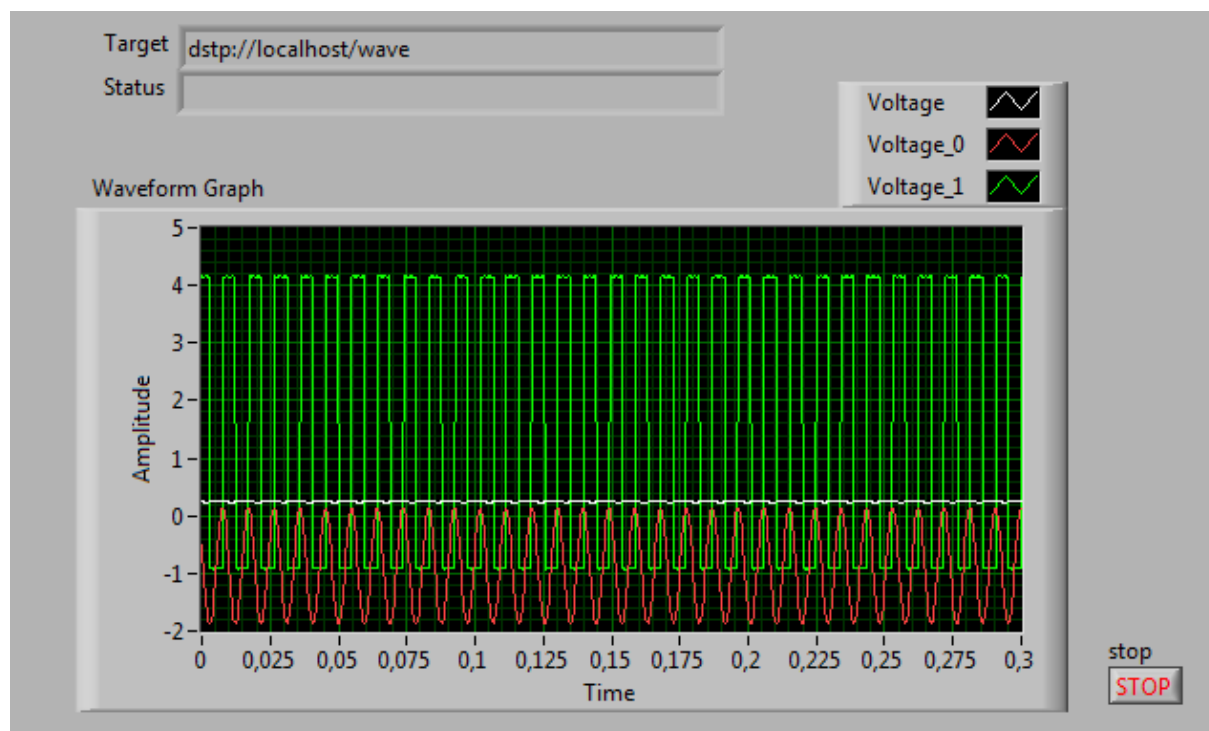
Nadawanie danych - interfejs użytkownika:



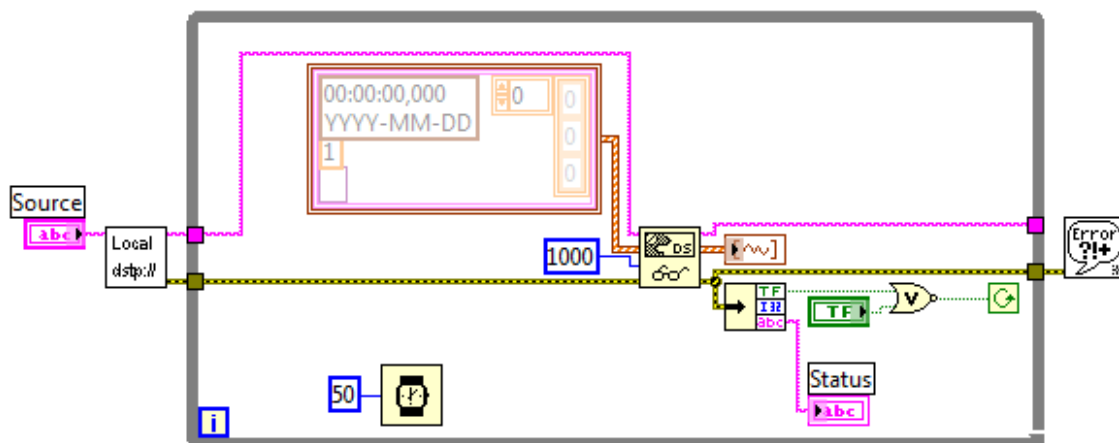
Nadawanie danych - diagram programu:



Odbieranie danych - interfejs użytkownika:

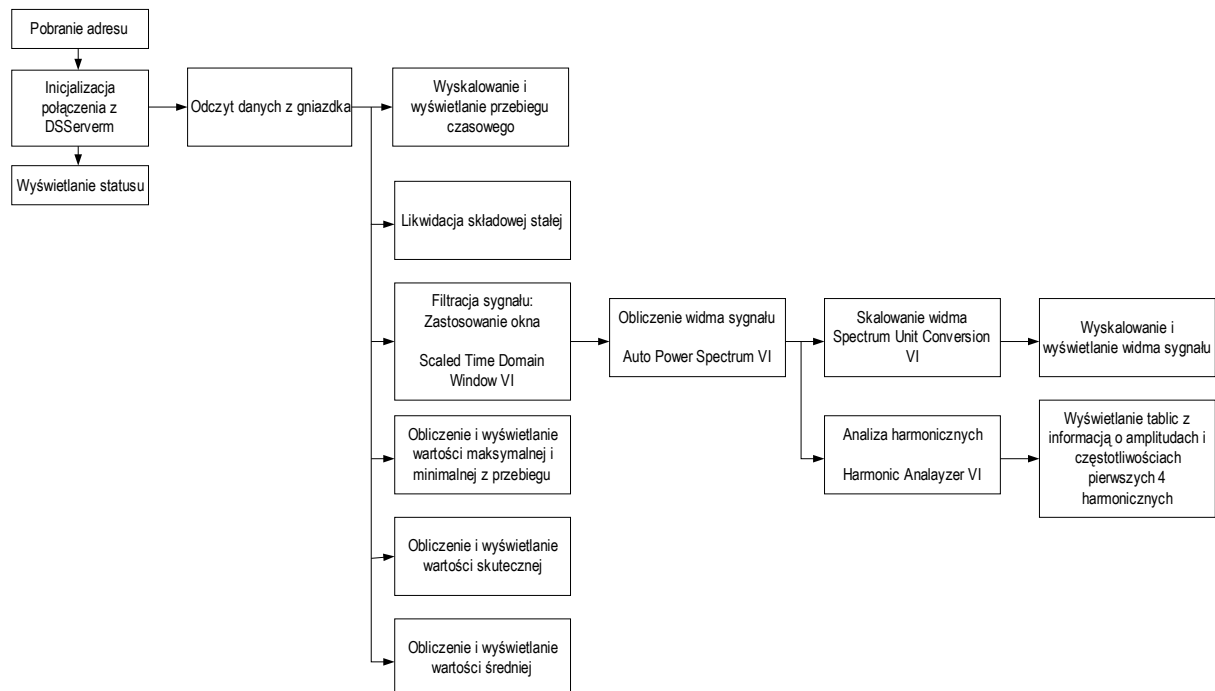


Odbieranie danych - diagram programu:



Ćwiczenie 34 – slajdy 302 do 305:

Wykorzystując, jako źródło sygnału aplikacje zrobione w ćwiczeniu 33 należy zbudować wirtualny przyrząd pomiarowy czytający dane z serwera pomiarowego. Przebieg należy poddać podstawowej analizie częstotliwościowej: widmo sygnału, detekcja harmoniczných oraz analizie parametrów czasowych, tzn. wyznaczyć wartości: średnia, skuteczna, maksymalna i minimalna.



Interfejs użytkownika:

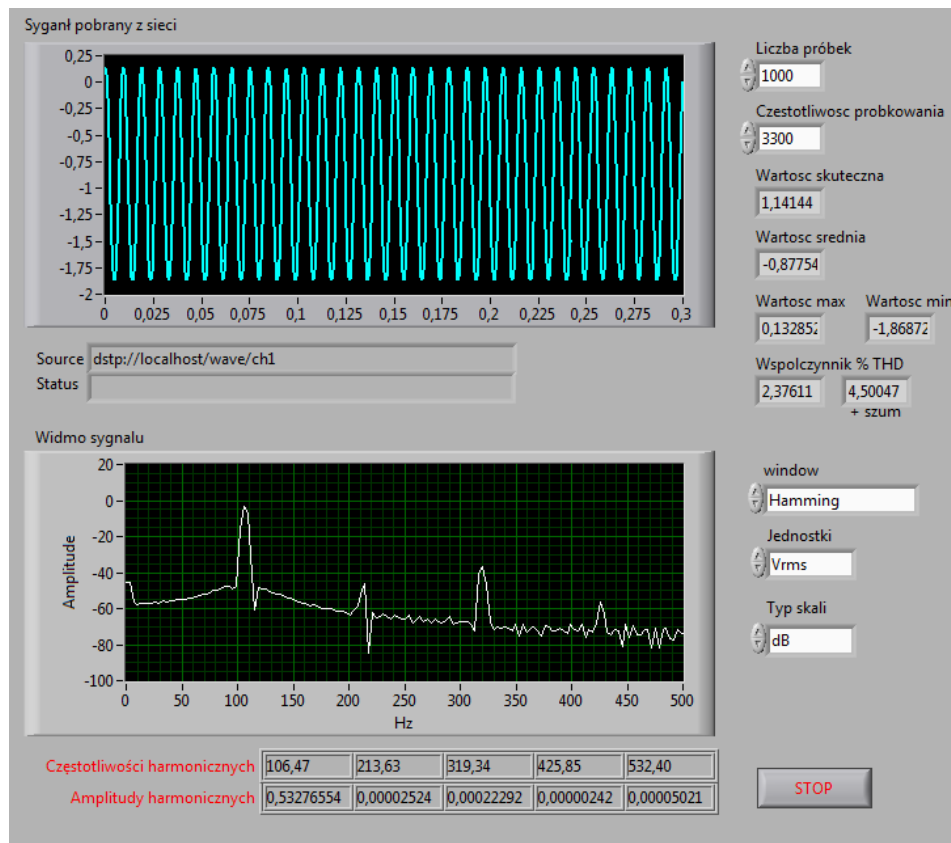
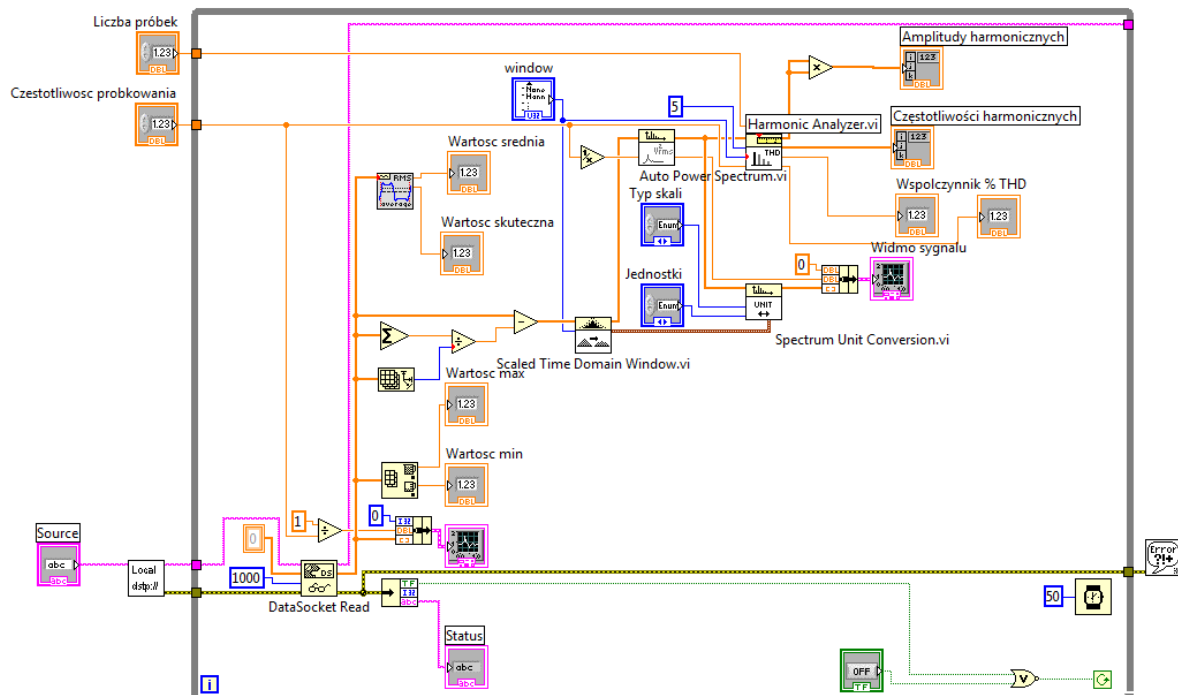


Diagram programu:



Ćwiczenie 35 – slajdy 308 do 309:

Wyświetlanie wielu przebiegów jednocześnie. Prezentacja zjawiska próbkowania sygnału.

Interfejs użytkownika:

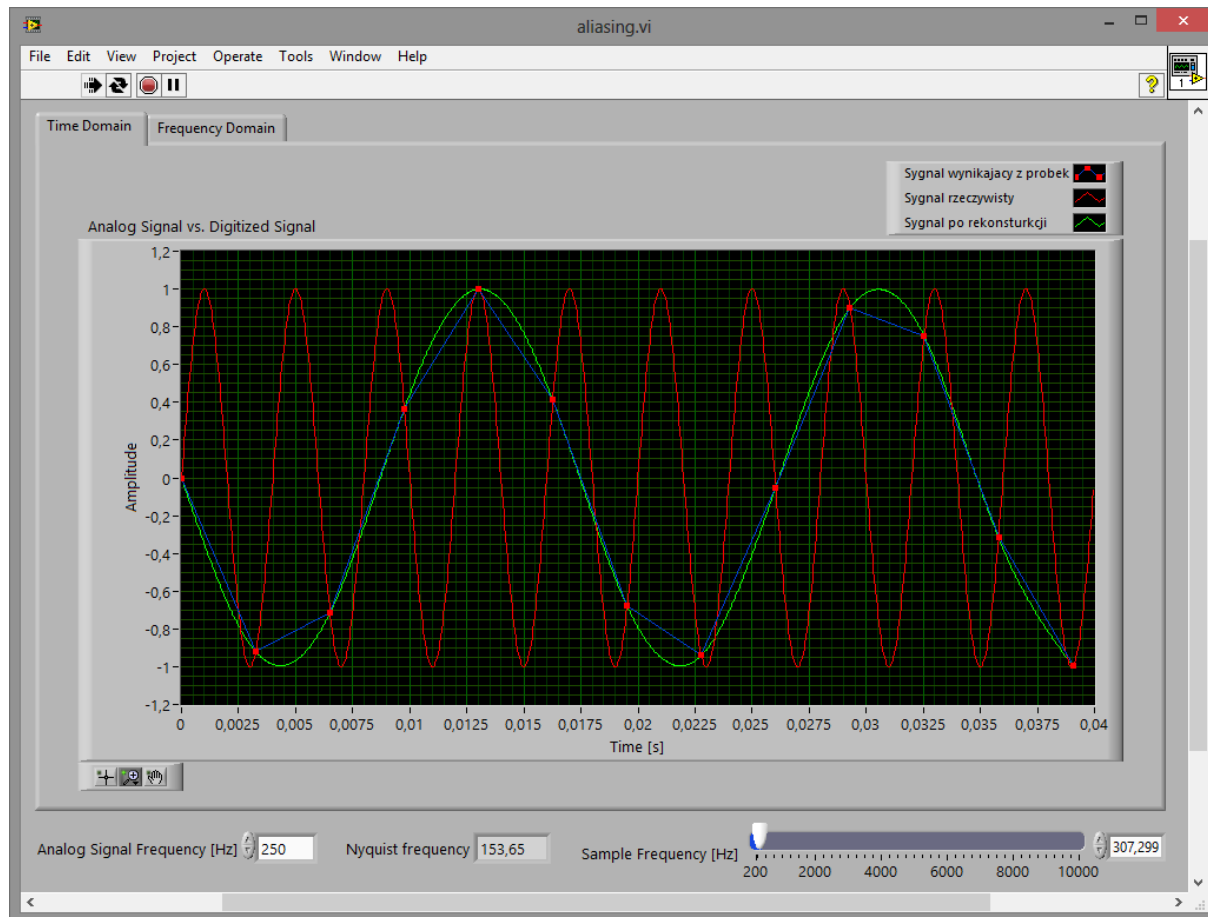
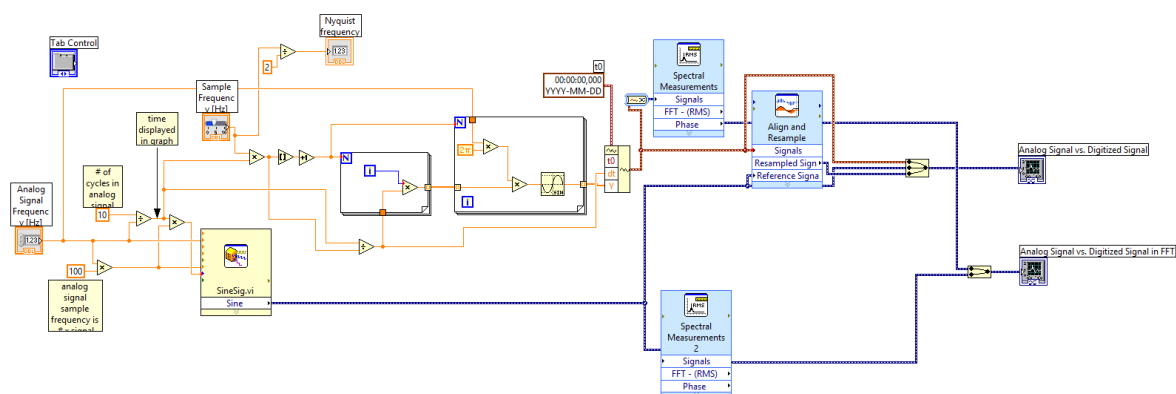


Diagram programu:



Ćwiczenie 36 – slajdy 310 do 311:

Generator przebiegów okresowych.

Interfejs użytkownika:

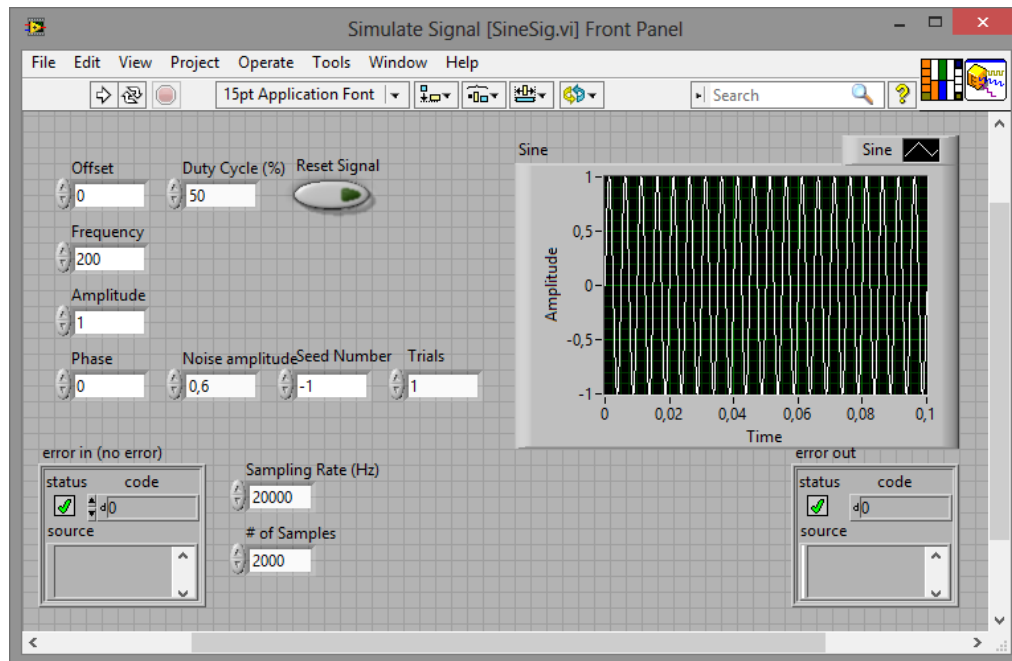
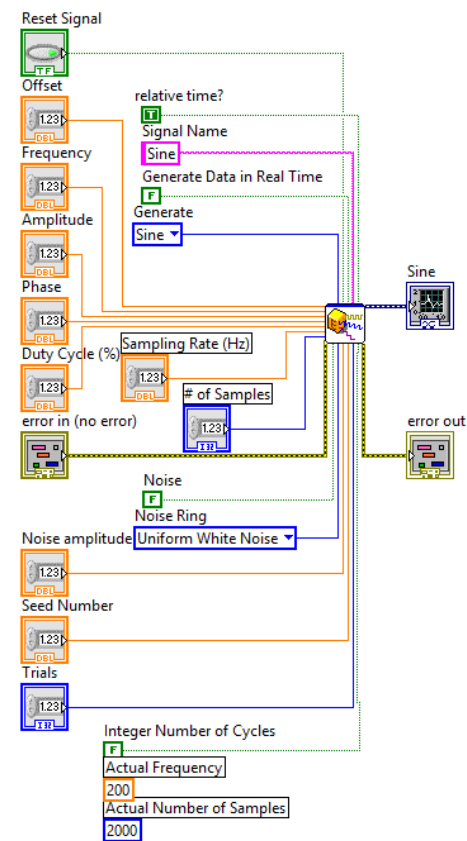
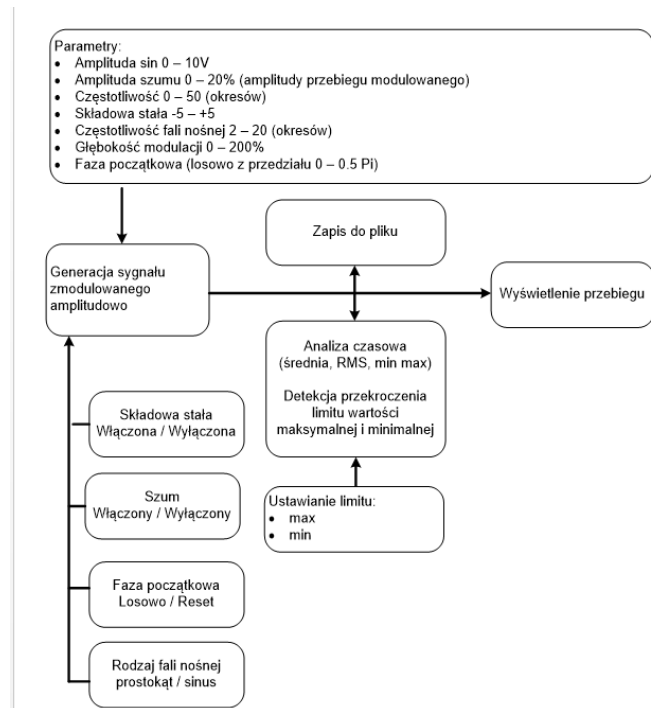


Diagram programu:

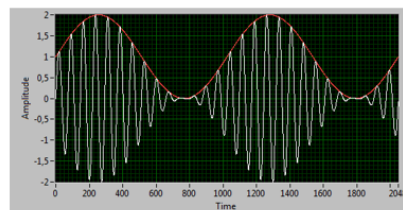


Ćwiczenie 37 – slajdy 312 do 314:

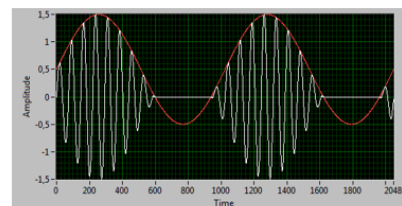
Należy zaprojektować przyrząd wirtualny umożliwiający wygenerowanie przebiegu modulowanego amplitudowo (2000 próbek). Fala nośna – przebieg sinusoidalny lub prostokątny o regulowanej ilości okresów przypadających na jeden okres sygnału modulowanego. Sygnał modulujący (ma być wyświetlony na wykresie) – sinusoidalny o regulowanej amplitudzie i częstotliwości (ilość okresów do wyświetlania). Faza początkowa ma być generowana losowo z przedziału $0 - 0,5 \pi$ lub ustawiana w zakresie $0 - 1 \pi$. Generator ma mieć możliwość regulowania składowej stałej, zaszumienia przebiegu oraz głębokości modulacji. Przebieg należy poddać podstawowej analizie parametrów czasowych, tzn. wyznaczyć wartości: średnia, skuteczna, maksymalna i minimalna. Przyrząd ma być wyposażony w detektor przekroczenia zadanego limitu (wykorzystać diody LED).



Modulacja 100%



Modulacja 150%



Interfejs użytkownika:

