



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

## 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

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

## Ćwiczenie:

# Środowisko programistyczne LabVIEW

Ćwiczenie opracowane w ramach projektu:  
**„Wirtualne Laboratoria Fizyczne nowoczesną metodą nauczania”**  
realizowanego przez **Warszawską Wyższą Szkołę Informatyki**

Autor ćwiczenia: Marcin Godziemba-Maliszewski ([elab@mgmaliszewski.pl](mailto:elab@mgmaliszewski.pl))

Na podstawie oryginalnych materiałów szkoleniowych firmy National Instrument ([www.ni.com](http://www.ni.com))

Projekt jest realizowany przez Warszawską Wyższą Szkołę Informatyki

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Zakres ćwiczenia:

- Zapoznanie ze środowiskiem LabVIEW i językiem G.
- Podstawowe funkcje LabVIEW.
- Wzorce projektowe w środowisku LabVIEW.
- 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.

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Zapoznanie z elementami środowiska LabVIEW

Projekt jest realizowany przez Warszawską Wyższą Szkołę Informatyki

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Gdzie można uzyskać więcej informacji ?

### 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 wspomagany komputerowo”
- **Dariusz Świstulski** „Oprogramowanie wirtualnych przyrządów pomiarowych w LabVIEW”
- **Wiesław Winiecki** „Graficzne zintegrowane środowiska programowe do programowania komputerowych systemów pomiarowo-kontrolnych”

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Tytułem wstępu ...

### Co to jest i co oznacza LabVIEW?

## Laboratory Virtual Instrument Engineering Workbench

- Istnieje od 1980 roku
- Graficzne środowisko programowania
- Stworzone przez inżynierów dla inżynierów
- Używane w laboratoriach badawczo naukowych (CERN – LHC Collimator Project, JPL, FedEx, Siemens Medical, European Southern Observatory ...)

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

## Środowisko LabVIEW

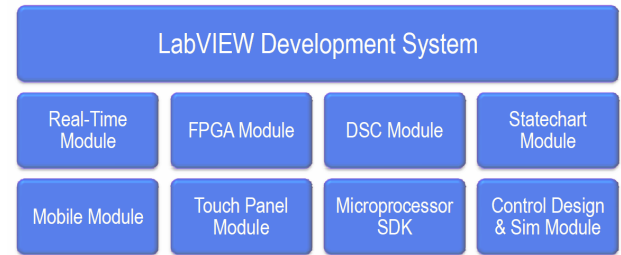
Co sprawia że LabVIEW zyskuje taką popularność?

- Środowisko stworzone przez inżynierów dla inżynierów a nie dla programistów.
- Środowisko wykorzystujące język G (graficzny) prezentujący dany proces bezpośrednio na diagramie.
- Procesy znacznie łatwiej wyobrazić sobie i przedstawić w formie diagramu niż kodu C.
- Łatwiej zobaczyć i zrozumieć błędy w budowanej aplikacji.
- Dostęp do najnowszych technologii spotykanych w przemyśle.
- Działa na różnych platformach: Windows, Linux, Mac, Solaris.
- Duża ilość funkcji (ponad 1200) wspierających analizę i przetwarzanie sygnałów zoptymalizowanych pod kątem wielowątkowości i przetwarzania równoległego.
- Obsługa czasu rzeczywistego

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## LabVIEW – podstawowe moduły

Duża liczba dostępnych modułów od producenta jak i firm trzecich.



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## LabVIEW czy Text-based programming?

Co łatwiej przemawia do inżyniera – zapis tekstowy?

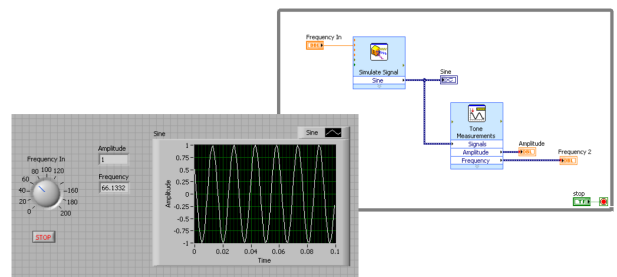
```
void ReadDataFile(void)
{
    FILE *hFile;
    char FileName[260];
    FileSelectPopup ("*.* txt" " Open existing text file",
        VAL_LOAD_BUTTON, 0, 1, 0, 0, FileName);
    hFile = fopen(FileName, "r");
    FillBytes (Buffer, 0, 700, 0);
    fread(Buffer, 1, 699, hFile);
    ResetTextBox (mainPanel, MAINPANEL_TEXTBOX, "");
    SetCtrlVal (mainPanel, MAINPANEL_TEXTBOX, Buffer);
    fclose (hFile);
}

void WriteDataFile(void)
{
    FILE *hFile;
    char FileName[260];
    FileSelectPopup ("*.* txt" " Enter file name to save",
        VAL_SAVE_BUTTON, 0, 1, 0, 0, FileName);
    hFile = fopen(FileName, "w");
    GetCtrlVal (mainPanel, MAINPANEL_TEXTBOX, Buffer);
    fwrite(Buffer, 1, strlen(Buffer), hFile);
    fclose (hFile);
}
```

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Język programowania G

Czy może graficzna reprezentacja działań?



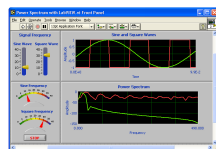
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Z czego składa się program w LabVIEW

Elementy składowe programu:

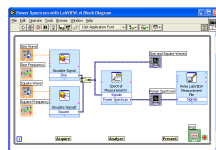
### Panel czołowy (Front Panel)

- Controls (Zadajniki)
- Indicators (Wskaźniki)



### Schemat blokowy (Block Diagram)

- Program towarzyszący panelowi czołowemu
- „Kod programu” zarządzający wykonaniem aplikacji



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Prezentacja błędów

- W LabVIEW prezentacja błędów odbywa się bezpośrednio w trakcie tworzenia diagramu. Błąd sygnalizowany jest w następujący sposób:



- Można uzyskać dodatkową pomoc opisującą problem z połączeniem dwóch elementów diagramu.
- W tekstowych środowiskach programistycznych zrozumienie błędu często wymaga sporego doświadczenia:

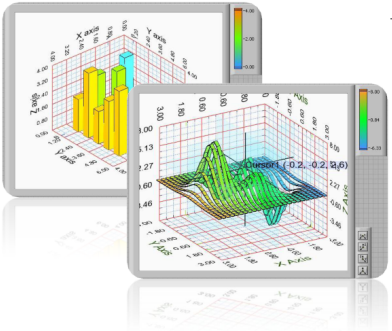
```
: error C2248: 'Rectangle::width' : cannot access private member declared in class 'Rectangle'
cpp(7) : see declaration of 'Rectangle::width'
cpp(5) : see declaration of 'Rectangle'
: error C2248: 'Rectangle::width' : cannot access private member declared in class 'Rectangle'
cpp(7) : see declaration of 'Rectangle::width'
cpp(5) : see declaration of 'Rectangle'
: error C2248: 'Rectangle::width' : cannot access private member declared in class 'Rectangle'
cpp(7) : see declaration of 'Rectangle::width'
```

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”



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

### LabVIEW - wykresy 2D i 3D



**New 3D Plots**

- Contour
- Mesh
- Quiver
- Stem
- Surface
- Comet

**New 2D Plots**

- Pie
- Scatter
- Ribbon
- Waterfall
- Bar
- Compass
- Feather
- Errorbar
- Plotmatrix

13

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### LabVIEW - przechowywanie danych

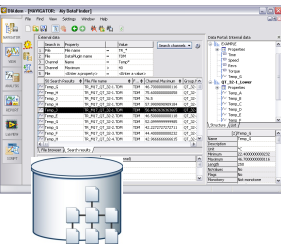
Dane można zapisywać na wiele sposobów i w wielu formatach. Problemem jest wybór odpowiedniego formatu. Format zapisu zdeterminowany jest przez system dalszej obróbki oraz przez ilość danych jakie chcemy zapisać.



14

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Przeszukiwanie danych - DIAdem



Typowy obszar zastosowań (Excel dla inżynierów):


- Wyszukiwanie, analiza i raportowanie danych.
- Automatyczne indeksowanie danych umożliwiające szybkie odnajdywanie interesujących nas danych (indeksowanie atrybutów w każdym z plików wybranych do analizy)
- Możliwość przeszukiwania danych pod kątem plików, grup i kanałów spełniających nasze wymagania

15

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### LabVIEW – automatyzacja zadań (scripting)

- **Możliwość automatyzacji często wykonywanych zadań programistycznych w środowisku.**
- **Można też stworzyć i podpiąć do menu nowe stworzone przez siebie funkcje.**



16

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Aplikacje mobilne w LabVIEW

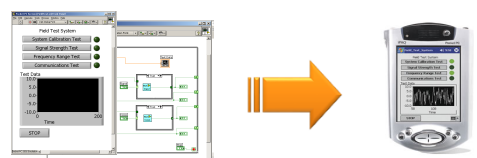
- Moduł PDA zawiera elementy interfejsu użytkownika znane z LabVIEW dla systemów Palm OS oraz Pocket PC.
- Kod jest optymalizowany pod kątem wydajności urządzeń PDA.
- Wbudowany emulator umożliwiający przetestowanie i debugowanie aplikacji.
- Umożliwia współpracę z kartami zbierania danych oraz przyrządami autonomicznymi.
- Umożliwia komunikację za pośrednictwem Bluetooth, WiFi, IrDA, RS232.

17

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Jak zacząć budowanie aplikacji dla PDA?

- Wybór właściwego systemu operacyjnego urządzenia PDA.
- Zbudowanie interfejsu użytkownika.
- Zaimplementowanie logiki programu.
- Kompilacja i uruchomienie aplikacji w symulatorze – weryfikacja poprawności działania.
- Instalacja na urządzeniu.



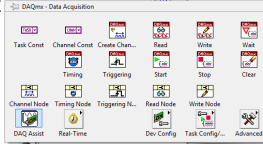
18

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Używanie DAQ w PDA

- Dostęp do funkcji DAW poprzez bibliotekę DAQmx
- Współpraca ze wszystkimi kartami PCMCIA
  - 200 ks/s
  - wielokanałowa akwizycja
  - możliwość synchronizacji i używania triggerów
- Intuicyjna obsługa – jak DAQmx dla PC



### Używanie przyrządów autonomicznych

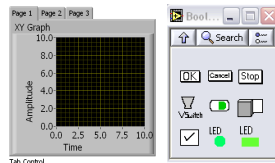
- Możliwość użycia przyrządów pomiarowych wykonanych w postaci kart PCMCIA, np. Multimetr cyfrowy 5 1/2 cyfry (NI 4050)
- Tworzy z PDA przenośne narzędzie (przyrząd) o właściwościach dopasowanych do potrzeb użytkownika.
- Możliwość rejestracji i wysyłania danych pomiarowych (karta SD, WiFi, Bluetooth)



### Ograniczenia

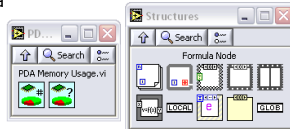
#### Rozdzielczość ekranu

- konieczność przemysłenia i zaplanowania rozmieszczenia jedynie niezbędnych elementów na interfejsie



#### Pamięć

- przemyślana i zoptymalizowana pod względem zużycia pamięci aplikacja



#### Moc obliczeniowa

- aplikacje oparte o wzorzec Event Structure

### Podsumowanie

- Moduł PDA dla LabVIEW umożliwia budowanie aplikacji umożliwiających akwizycję sygnałów oraz mobilne pomiary.
- Można zbudować dowolną aplikację do wspomaganie procesów konserwacji i diagnostyki systemów pomiarowych.
- Możliwość komunikacji za pośrednictwem WiFi, Bluetooth.
- Mobilne urządzenie pomiarowo-diagnostyczne.



### Zapoznanie z elementami środowiska LabVIEW

#### LabVIEW Project

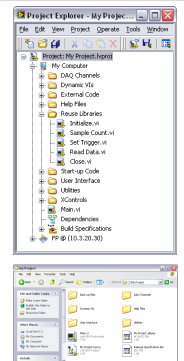
Jako sposób zarządzanie projektem i utrzymanie porządku w plikach projektu oraz na automatyzację procesów wytwarzania aplikacji.

### Zarządzanie plikami programu

LabVIEW Project Explorer pozwala osiągnąć właściwą organizację plików w projekcie

#### Kilka zasad pozwalających uniknąć podstawowych problemów z organizacją plików projektu:

- Utworzyć osobny katalog dla każdego projektu
- Budować drzewo katalogów zgodne z hierarchią komponentów w projekcie
- „Vijaje” nadrzędne powinny być dostępne z głównego katalogu projektu.
- Struktura katalogów i plików powinna odzwierciedlać strukturę projektu



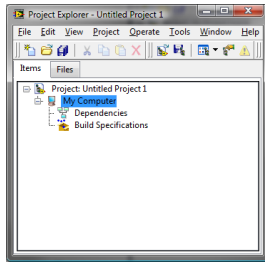


Opracowane w ramach projektu: "Wirtualne Laboratoria Fizyczne nowoczesną metodą nauczania" realizowanego przez Warszawską Wyższą Szkołę Informatyki.

## LabVIEW Project – nowy projekt

### Project Explorer (PE)

- Używać PE do tworzenia i edycji plików projektu
- Domyślne katalogi (gałęzie):
  - **Project Root** (główne drzewo projektu)
  - **My Computer** (elementy składowe projektu)
  - **Dependencies** (biblioteki projektu)
  - **Build Specifications** (zarządzanie dystrybucją projektu)

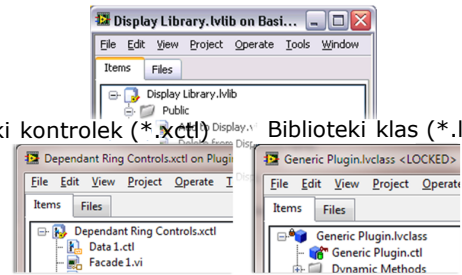


## Odzwierciedlenie architektury systemu

### Biblioteki projektu (\*.lvlib)

### Biblioteki kontrol (\*.xctl)

### Biblioteki klas (\*.lvclass)



## Biblioteki w projektach – kilka faktów

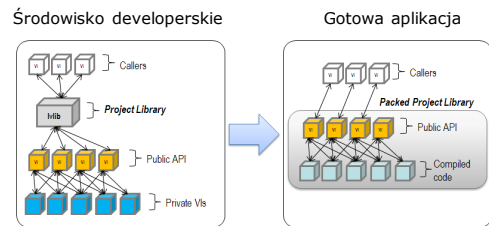
- Budować biblioteki lvlib a nie LLB czy zip będące jedynie kontenerem dla plików
- LVLIB pozwalają na używanie względnych nazw funkcji:

mylibrary.lvlib:Initialize.vi

- **Korzyści ze stosowania bibliotek lvlib:**
  - Przestrzenie nazw (Namespaces)
  - Kontrola wersji
  - Kontrola dostępu do kodu – możliwość zastosowania hasła dostępu do kodu źródłowego
  - Krok w kierunku budowy aplikacji z komponentów
  - Krok w kierunku programowania obiektowego w LabVIEW
  - Przyspieszenie budowania przyszłych aplikacji dzięki wykorzystaniu gotowych komponentów

## Biblioteki projektu

### Publikacja biblioteki jako pojedynczego pliku



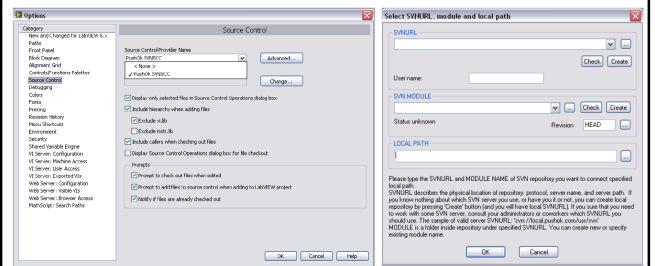
- Upraszcza i przyspiesza budowanie pakietu dystrybucyjnego
- Hierarchia bibliotek umieszczona w pojedynczym pliku

## Source Code Control – zarządzanie wersją

- Perforce
- Microsoft Visual SourceSafe
- IBM Rational ClearCase
- Serena Version Manager (PVCS)
- IonForge Evolution
- Borland StarTeam
- MKS Source Integrity
- Subversion
- CVS
- Inne kompatybilne z pakietem Microsoft Visual Studio

## Source Code Control – zarządzanie wersją

Oprogramowanie SCC musi posiadać uniwersalną wtyczkę (jak do każdego innego środowiska) do LabVIEW pozwalającą na integrację z środowiskiem



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

### Source Code Control – zarządzanie wersją

KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOLECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### LabVIEW Project

**Project Explorer Window – dwa typy katalogów: wirtualne i samo powielające – które użyć?**

**Katalog wirtualny**  
**Katalog samopowielający**

**Katalog wirtualny** – logiczne pogrupowanie elementów projektu nie mające odzwierciedlenia w strukturze katalogów

**Katalog samopowielający** – będący odzwierciedleniem fizycznego katalogu na dysku. Wszystkie elementy dodane do katalogu są automatycznie dodawane do projektu

KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOLECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Katalog wirtualny – o czym nie zapomnieć

- Łatwiejsze dynamiczne ładowanie plików
- Należy pamiętać aby wszystkie potrzebne pliki bibliotek oraz plików vi ładowanych dynamicznie zostały dołączone do pakietu instalacyjnego (lub w projekcie poprawnie skonfigurować dostęp www lub ftp)

KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOLECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### LabVIEW Project – biblioteki projektu

**Dependencies** – katalog zawierający informacje o wszystkich używanych funkcjach i bibliotekach. Jeśli używamy swoich bibliotek koniecznie trzeba je tu umieścić.

KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOLECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### LabVIEW Project – dystrybucja aplikacji

**Build Specification:**

- Application (EXE) – wymagany jedynie LabVIEW Runtime
- Shared Library (DLL) – do użytku z C/C++/Visual Studio i innymi środowiskami
- Installer – kompletny pakiet dystrybucyjny, zawiera pliki, sterowniki i potrzebne biblioteki
- Source Distribution – umożliwia zbudowanie paczki np. dla repozytorium kodu
- Web Service, Zip File

KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOLECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### App Builder Log File

**Ułatwia budowanie i zarządzanie dokumentacją wersji dystrybucyjnej**

KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOLECZNY

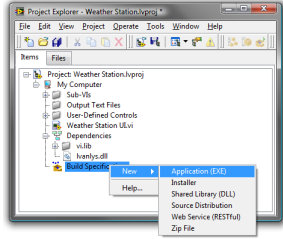
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”



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

## Application Builder – executables

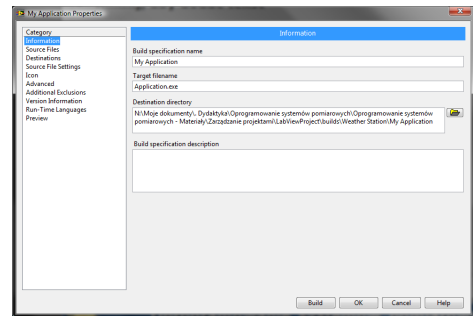
Buduje aplikacje w postaci pliku exe. Do uruchomienia pliku nie jest wymagane LabVIEW. Jest potrzebny darmowy LabVIEW Runtime (do pobrania ze strony producenta) w odpowiedniej wersji.



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Application Builder – executables

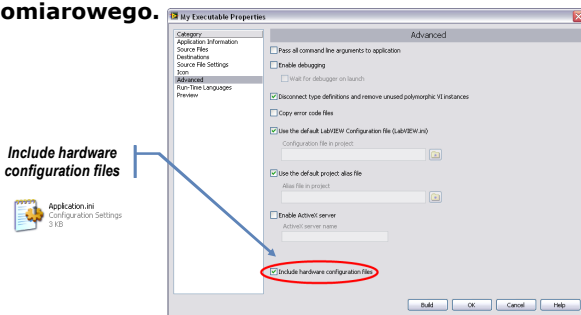
### Podstawowe informacje o pliku exe



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Application Builder – executables

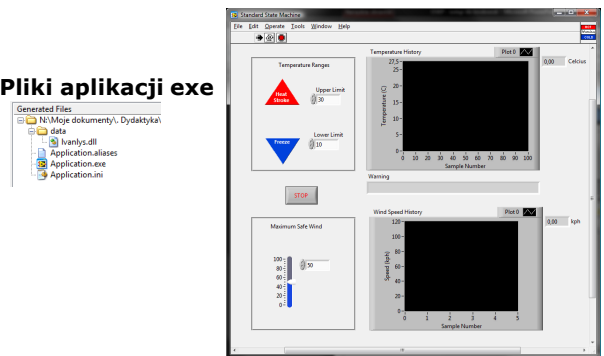
### Możliwość dołączenia plików konfiguracja sprzętu pomiarowego.



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Application Builder – executables

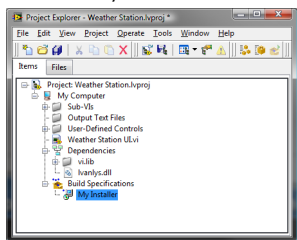
### Pliki aplikacji exe



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Application Builder – tworzenie pakietu dystrybucyjnego

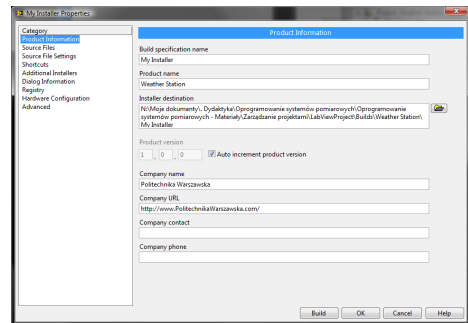
- Tworzy profesjonalny pakiet dystrybucyjny (instalacyjny).
- Można dołączać dowolne pliki, dokumentacja, pliki pdf itp.
- Można wymusić instalację sterowników i usług oraz innych aplikacji i pakietów składowych



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Application Builder – tworzenie pakietu dystrybucyjnego

### Podstawowe informacje a instalowanej aplikacji



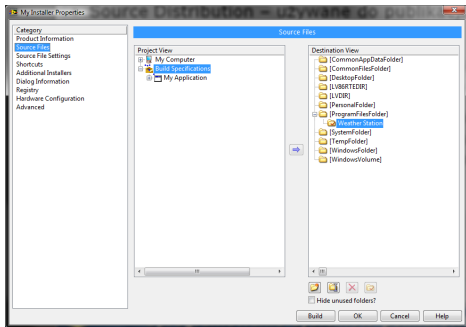
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”



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

**Application Builder – tworzenie pakietu dystrybucyjnego**

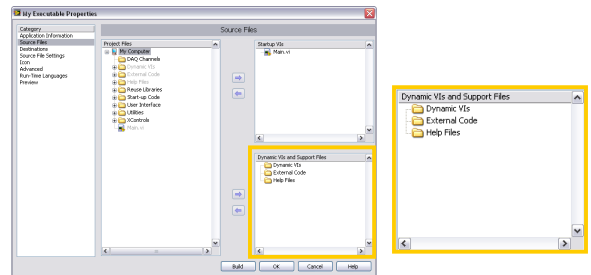
**Instalowanie w domyślnych folderach systemu Windows**



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

**Application Builder – tworzenie pakietu dystrybucyjnego**

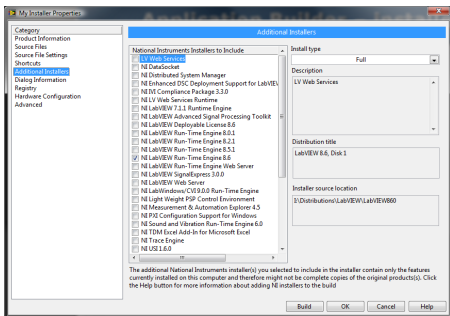
**Nie zapomnieć o dynamicznie ładowanych plikach!  
(pliki vi, biblioteki dll, pliki konfiguracyjne)**



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

**Application Builder – tworzenie pakietu dystrybucyjnego**

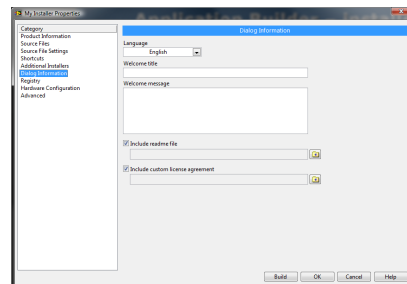
**Możliwość dołączenia bibliotek instalowanych automatycznie**



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

**Application Builder – tworzenie pakietu dystrybucyjnego**

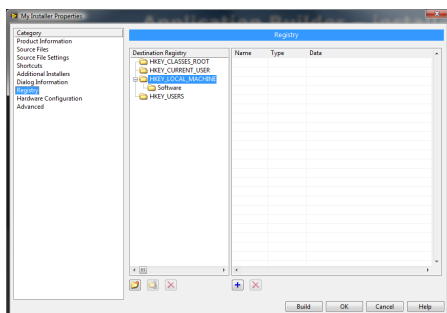
**Konfiguracja powitalnego okna dialogowego  
BRAK wsparcia dla języka polskiego!**



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

**Application Builder – tworzenie pakietu dystrybucyjnego**

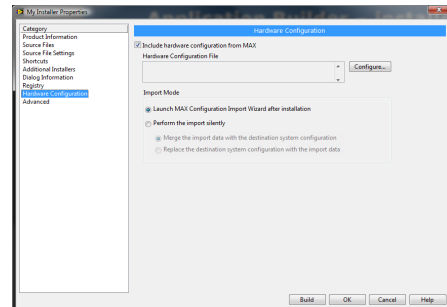
**Możliwość dodawania wpisów do rejestru Windows**



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

**Application Builder – tworzenie pakietu dystrybucyjnego**

**Możliwość dołączania plików z konfiguracją sprzętu**



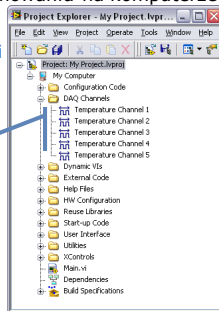
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Application Builder – konfiguracja sprzętowa

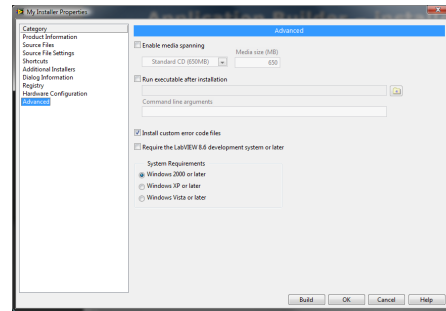
Pozwala na zapisanie i skonfigurowanie odpowiednich zadań w trakcie wykonywania instalacji oprogramowania na komputerze użytkownika.

Należy pamiętać o dodaniu do instalacji programu NI-DAQmx.



### Application Builder – tworzenie pakietu dystrybucyjnego

#### Definiowanie sposobu dystrybucji pakietu instalacyjnego



### Application Builder – pozostałe opcje

- **Source Distribution** – używane do publikacji źródeł programu
- **Web Service** – przygotowanie instalatora usługi sieciowej przeznaczonej do dystrybucji danych
- **Zip File** – paczka zip zawierająca kody źródłowe i/lub inne elementy projektu. Najczęściej używana do tworzenia kopii zapasowych danej wersji programu.

### Weryfikacja instalacji

- Instalator powinien zostać uruchomiony na czystym komputerze który nie posiada zainstalowanych żadnych komponentów i programów pochodzących od producenta National Instrument.
- Po instalacji aplikacja powinna zostać uruchomiona i przetestowana pod kątem poprawności działania, poprawności współpracy z urządzeniami pomiarowymi i kartami zbierania danych.
- Po zakończeniu należy przetestować uninstaller i po odinstalowaniu aplikacji porównać obrazy i konfigurację systemu.

### Dystrybucja oprogramowania

W przypadku dystrybucji na CD/DVD należy pamiętać o zamieszczeniu prawidłowej struktury plików uruchomieniowych instalatora:

- Installer
- Autorun.exe
- Autorun.inf
- Dokumentacja i instrukcja instalacji wraz z pomocą.

### Zarządzanie projektem i drzewem projektu

#### Podsumowanie:

- Struktura katalogów i plików powinna odzwierciedlać strukturę projektu
- Utworzyć oddzielny katalog dla każdej biblioteki
- Uważać na opcję „Zapisz jako” – używać tylko w ostateczności
- Czytać a nie „przeklikiwać” ostrzeżenia i informacje w trakcie zapisywania plików i nie tylko.
- W dużych projektach używać bibliotek
- Używać unikalnych i jednoznacznych nazw funkcji.
- Nie używać więcej niż jednego projektu w tym samym czasie.
- Nie używać kopiowania plików gdy są otwarte pliki projektu.
- Duże aplikacje umieszczać w projektach podzielonych na spójne części



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

## Podstawowe funkcje LabVIEW

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

55

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Program w LabVIEW

**Elementy składowe programu:**

- Panel czołowy (Front Panel)**
  - Controls (Zadajniki)
  - Indicators (Wskaźniki)
- Schemat blokowy (Block Diagram)**
  - Program towarzyszący panelowi czołowemu
  - „Kod programu” zarządzający wykonaniem aplikacji

56

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Panel Czołowy czyli GUI (VI Front Panel)

**Ikona reprezentująca funkcję**

**Pasek narzędzi**

**Zadajnik (Control)**

**Wskaźnik (Indicator) wyświetlacz graficzny**

**Opcje wskaźnika (legenda wykresu)**

**Opcje wskaźnika (legenda wykresu)**

57

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Schemat blokowy (VI Block Diagram)

**Pasek narzędzi**

**funkcja (subVI)**

**Funkcje matematyczne**

**Reprezentacja graficzna wyświetlacza**

**Połączenia elementów**

**Reprezentacja graficzna pętli while**

**Stała numeryczna**

**Funkcja timera**

**Reprezentacja graficzna przełącznika z panelu czołowego**

58

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Express VI, VI i Funkcje

- Express VI: interaktywny VI wyposażony w panel konfiguracyjny
- Standard VI: modułowy VI
- Instrukcje i Funkcje: podstawowy element LabVIEW nie posiada panelu konfiguracyjnego

59

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Paleta kontrolki i funkcji

**Paleta funkcji programu**  
**Functions Palette**  
**(Block Diagram Window)**

**Paleta elementów płyty czołowej**  
**Controls Palette**  
**(Front Panel Window)**

60

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”



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

### Paleta narzędzi (Tools Palette)

Paleta narzędzi wykorzystywana jest do budowania, modyfikowania oraz obsługiwanie: płyty czołowej i diagramu programu.

Narzędzie automatycznego wyboru

- Obsługa płyty czołowej
- Przenoszenie/skalowanie
- Etykiety tekstowe
- „Szpuła” – łączenie funkcji
- Menu kontekstowe
- Przewijanie/przenoszenie zawartości ekranu
- Wstawianie pułapek
- Próbnik
- Kopiowanie kolorów
- Zmiana kolorów

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Pasek narzędzi (Status Toolbar)

- Uruchomienie programu
- Uruchomienie w pętli
- Przerwanie działania
- Paauza
- Ustawienia tekstu
- Wyrównanie elementów płyty czołowej
- Rozmieszczenie elementów płyty czołowej
- Zmiana położenia elementów płyty czołowej w osi Z
- Przeskalowanie elementów płyty czołowej
- Śledzenie wykonania programu (krok po kroku)
- Funkcja Step Into
- Funkcja Step Over
- Funkcja Step Out
- Przywrócenie wartości
- „Posprzątanje” vi

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Typy danych - reprezentacja graficzna

Zmienna logiczna – reprezentowana na diagramie jako element graficzny

Literaly typów danych

- Zmienna logiczna
- Typ całkowitoliczbowy
- Typ zmiennoprzecinkowy
- Ciąg znaków
- Typ zespolony
- Obraz graficzny, plik
- Tablica jednowymiarowa typu zmiennoprzecinkowego
- Tablica dwuwymiarowa typu zmiennoprzecinkowego
- Tablica n-wymiarowa typu zmiennoprzecinkowego
- Struktura danych (ang. Cluster)

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Automatyczna konwersja typów

#### Zasady automatycznej konwersji:

- LabVIEW wybiera reprezentację używającą więcej bitów.
- Przy tej samej ilości bitów (reprezentacja ze znakiem i bez) LabVIEW wybierze reprezentację bez znaku a nie ze znakiem.

Możliwość wymuszenia konwersji typów

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Tips & Tricks – przydatne skróty

Przydatne skróty klawiaturowe:

Windows	Sun	Linux	MacOS	
<Ctrl-R>	<◀-R>	<M-R>	<⌘-R>	Uruchom VI
<Ctrl-F>	<◀-F>	<M-F>	<⌘-F>	Znajdź obiekt
<Ctrl-H>	<◀-H>	<M-H>	<⌘-H>	Wyświetl Context Help
<Ctrl-B>	<◀-B>	<M-B>	<⌘-B>	Usuń błędne połączenia
<Ctrl-W>	<◀-W>	<M-W>	<⌘-W>	Zamknij aktywne okno
<Ctrl-E>	<◀-E>	<M-E>	<⌘-E>	Przełączanie Diagram/Panel

- Tools Palette -> <shift>-right-click
- Increment/Decrement z <shift> znacznie szybciej
- Wiring: spacebar -> zmiana kierunku połączeń
- Wiring: right mouse click -> usunięcie rysowanego połączenia

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Tips & Tricks – zamiana terminali

**Aby zamienić kolejność podłączeń należy wcisnąć CTRL a następnie lewy przycisk myszy na terminalu**

**Niestety działa tylko na terminalach o dwu wejściach. Terminale muszą być podpięte.**

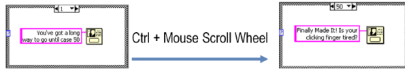
WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Tips & Tricks – przewijanie struktur

Aby „przewijać” zawartość struktur należy wcisnąć CTRL i za pośrednictwem rolki można przewijać zawartość.

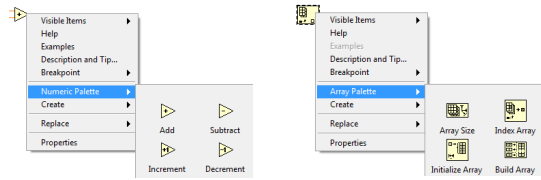


Działa z:

- Case Structures
- Event Structures
- Stacked Sequence Structures
- Diagram Disable Structures

### Tips & Tricks – dodawanie elementów

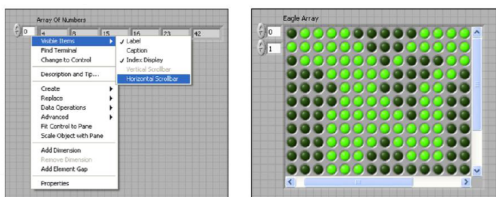
Szybkie wstawianie nowych elementów z podobnej grupy za pomocą prawego klawisza myszy.



### Tips & Tricks – tablice i przewijanie

Włączenie przewijania zawartości dużych tablic na interfejsie znacznie ułatwia pracę.

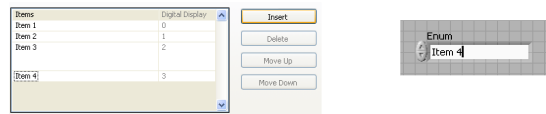
Visible Items >> Horizontal Scrollbar (or Visible Items >> Vertical Scrollbar)



### Tips & Tricks – wstawianie nowych pól w listach

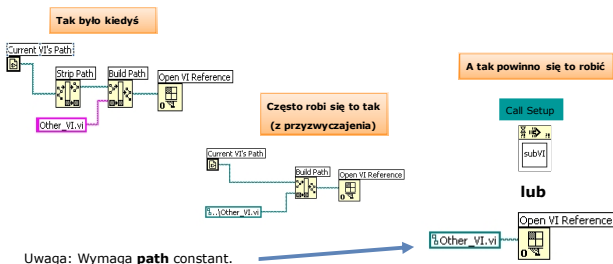
Aby wpisać jakąś wartość do pola listy należy wcisnąć CTRL gdy jest się w polu listy co przełączy automatycznie użytkownika do trybu Text Tool.

Dodanie nowego pola: Shift + Enter



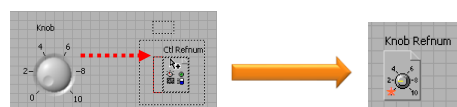
### Tips & Tricks – używanie ścieżek do plików

LabVIEW automatycznie szuka plików w bieżącym katalogu aplikacji



### Tips & Tricks – Szybkie tworzenie referencji z kontrolą typu

Aby utworzyć referencję z kontrolą typu bez konieczności przechodzenia przez szereg okienek konfiguracyjnych należy utworzyć na interfejsie Referencje a następnie obiekt interfejsu (np. knob) i przeciągnąć go do referencji. Aby zachować oryginalny element na interfejsie należy użyć CTRL przy przeciąganiu.

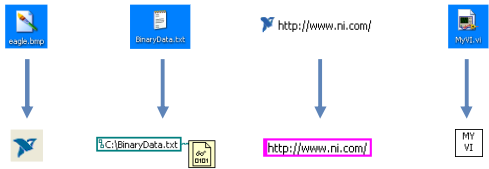


Opracowane w ramach projektu: "Wirtualne Laboratoria Fizyczne nowoczesną metodą nauczania" realizowanego przez Warszawską Wyższą Szkołę Informatyki.

### Tips & Tricks – przeciągnij i upuść

**Aby zaoszczędzić czas można użyć technologii przeciągnij i upuść do:**

- umieszczenia ikonki
- wpisania do zmiennej Path nazwy i ścieżki do pliku.
- Wpisania adresu url do zmiennej
- Wstawienie subVI

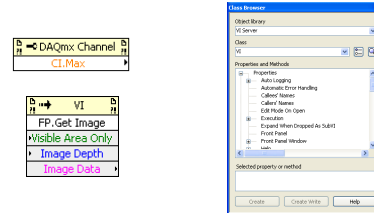


„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Tips & Tricks – szybki dostęp właściwości

**Aby szybko ustawić interesujące nas właściwości można:**

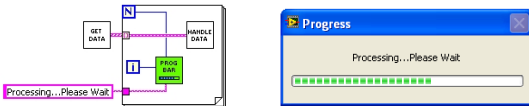
- użyć Class Browera (View->Clas Browser)
- Skrotu Ctrl+Shift + B
- Przeciągnąć daną metodę bezpośrednio do VI



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Tips & Tricks – wskaźnik postępu

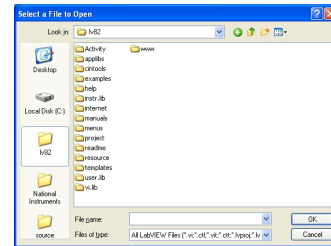
- Gotowy element: Progress Bar.vi
- Możliwość uruchomienia paska postępu dopiero po upływie określonego czasu.
- Automatyczna aktualizacja postępu.
- Wskazane dla długotrwałych obliczeń i operacji.



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Tips & Tricks – włączenie MRU

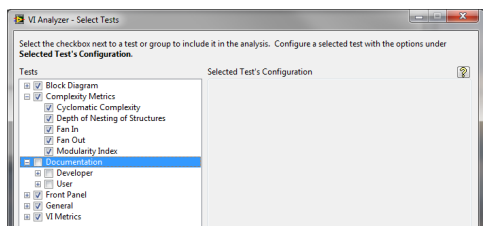
- W pliku LabVIEW.ini należy zmienić wpis „MRUFolders=TRUE”
- Szybki dostęp do 5 ostatnio używanych lokalizacji



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Tips & Tricks – automatyczna analiza budowanego VI

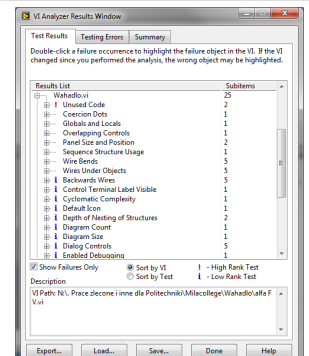
- Narzędzie VI Analyzer pozwala na sprawdzanie kodu (rewizja kodu) pod względem wydajności, dokumentacji i innych właściwości.
- Możliwość zapisania ustawień rewizji kodu dla przyszłych wywołań



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Tips & Tricks – wynik analizy

**Wypisuje szczegółowy raport dotyczący pojedynczego pliku lub całego projektu z rozbiem na wszystkie pliki i subVI-je.**



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Tips & Tricks – porównywanie dwóch VI

**Narzędzie: Tools -> Merge VIs**

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Tips & Tricks – używanie Stop Terminal

**Często pętle wykonywane są do końca chociaż można przerwać po wykonaniu oczekiwanej czynności, np.:**

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Wzorce projektowe i dobre praktyki w LabVIEW

Projekt jest realizowany przez Warszawską Wyższą Szkołę Informatyki

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Jak pisać dobre programy

**LabVIEW bez planu = katastrofa**  
**lub spaghetti code**

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Podstawowe architektury

Programy w LabVIEW za zwyczaj buduje się w oparciu o:

- Pojedynczą pętlę  
Architektura aplikacji wykorzystywana w prostych projektach gdzie w ramach jednej pętli wykonywane są określone zadania.
- Równoległe pętle  
Architektura aplikacji wykorzystywana w projektach gdzie zadania w ramach pętli wykonywane są w oddzielnym wątku. Zastosowanie takiej architektury programu umożliwia zrównoleżenie czynności – np.: akwizycja danych, obsługa interfejsu użytkownika, przetwarzanie i archiwizacja danych.

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Wzorce projektowe - Single Loop Architecture

**Przykład aplikacji wykorzystującej jedną pętlę**

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

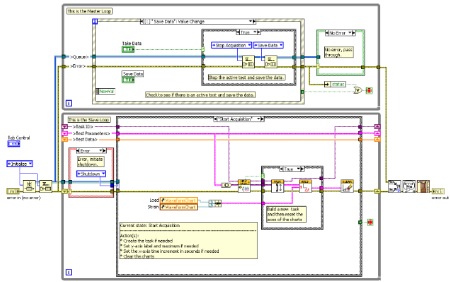
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”



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

### Wzorce projektowe - Parallel Loop Architecture

**Przykład aplikacji z dwoma równoległymi i niezależnymi pętlami.**



85

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Wzorce projektowe – Event Structure

**Aplikacja nastawiona na interakcję z użytkownikiem.**

Aplikacja wykrywa zdarzenia wygenerowane przez użytkownika i podejmuje akcje. Interfejs użytkownika nie powinien konsumować zbyt wiele zasobów systemowych. Wywoływane zdarzenia nie są zależne od siebie.

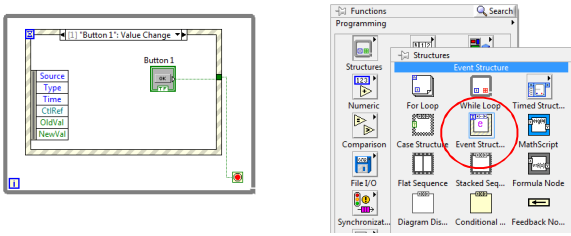
**Rozwiązanie: Wzorec oparty o strukturę zdarzeń.**

Możliwość blokowania interfejsu na czas wykonania obsługi zdarzenia i przekazanie wyników za pośrednictwem rejestrów w pętli głównej programu, bądź odblokowanie interfejsu na czas wykonania funkcji obsługi zdarzenia gdy zdarzenia są niezależne od siebie. Możliwość zastosowania kolejki do przekazywania wyników.

86

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Program oparty o strukturę zdarzeń

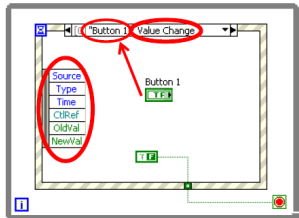


87

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Zdarzenia

- System operacyjny zgłasza zdarzenia zaistniałe w aplikacji i systemie.
- Event structures przechwytyje zarejestrowane zdarzenia i wykonuje kod przypisany do obsługi zdarzeń
- Pełna informacja o zdarzeniu wewnątrz case
- Kolejkowanie zdarzeń
- Blokada na czas wykonania zdarzenia



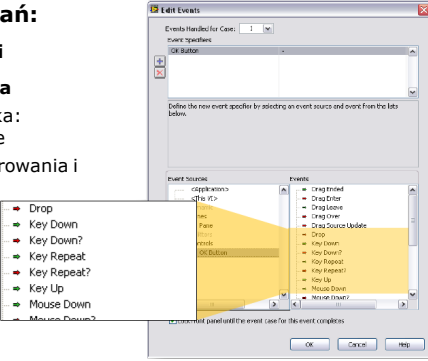
88

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Dostępne zdarzenia

**Kolejność działań:**

- Wybór kontrolki
- Wybór zdarzenia
  - Zielona strzałka: powiadomienie
  - Czerwona: filtrowania i oczekiwanie



89

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Wzorce projektowe – State Machine

Rozwiązanie przeznaczone dla aplikacji które muszą wykonywać się w oparciu o określony stan aplikacji. Użycie tego wzorca pozwala na decydowanie o stanie aplikacji po określonym stanie poprzedzającym. Można użyć dodatkowo struktury zdarzeń aby móc obsługiwać interakcję z użytkownikiem. Rozwiązanie przeznaczone dla systemów nie posiadających zbyt wielu stanów.

90

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Program oparty o maszynę stanów

91

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Możliwe sekwencje zdarzeń

- Sekwencja prosta**

- Sekwencja dynamiczna**

92

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Sekwencja dynamiczna - przykład

Soda costs 50 cents

93

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Zastosowanie pętli i struktur

Case structure – struktura danych dla każdego stanu programu

Logika przejścia do następnego stanu

94

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Wzorce projektowe – producer/consumer

**Wzorec używany w aplikacjach składających się z wątku produkującego dane oraz wątku który zajmuje się przetworzeniem i wizualizacją tych danych.**

Najczęstsze zastosowanie to wątek zajmujący się akwizycją sygnału oraz wątek zajmujący się ich obróbką, zapisem i wyświetlaniem. Przepływ danych pomiędzy pętlami zrealizowany jest zazwyczaj w oparciu o kolejkę. Kolejka może mieć określoną długość lub być „nieskończona” – rozwiązanie wygodne gdy akwizycja produkuje więcej danych niż wątek przetwarzający jest w stanie obrócić. Istnieje możliwość rozpraszania zadań na poszczególne maszyny – co ułatwia skalowalność aplikacji.

95

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Producer/Consumer - działanie

- Główna pętla uruchamia podrzędne pętle
- Możliwość wykonywania asynchronicznej pętli głównej i pętli podrzędnych
- Komunikacja i synchronizacja pomiędzy pętlami podrzędnymi – jeśli konieczna
- Kolejkowanie danych pomiędzy pętlą produkującą a przetwarzającą

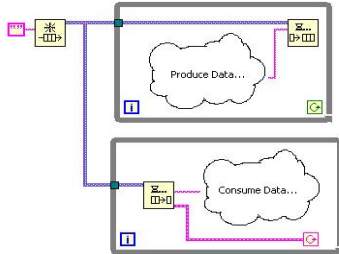
96

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Producer/Consumer - przykład

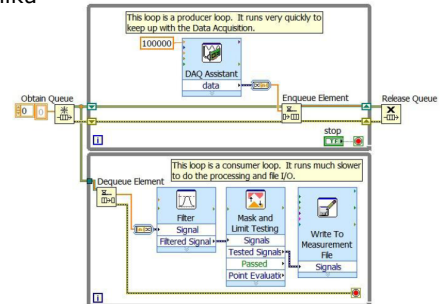
Przykład wzorca wykorzystującego pętlę do „produkcji” danych i pętli do ich przetwarzania z wykorzystaniem kolejki



WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI  
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Producer/Consumer - przykład

Przykład programu zbierania danych oraz jego filtracji i zapisu do pliku



WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI  
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Wzorce projektowe – „oo” programming

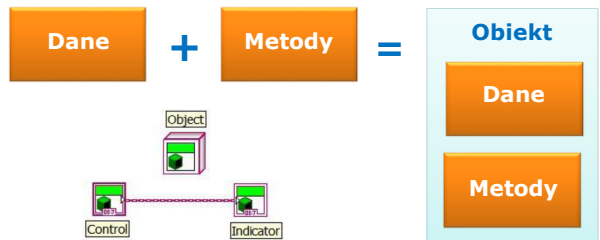
Programowanie obiektowe w LabVIEW stosowane jest w przypadkach gdy aplikacja musi wyprodukować bliżej nieokreśloną liczbę obiektów o określonych właściwościach.

Znacznie komplikuje budowanie aplikacji ale niektóre zadania tylko tak można rozwiązać.

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI  
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Obiektowość – znana z innych języków

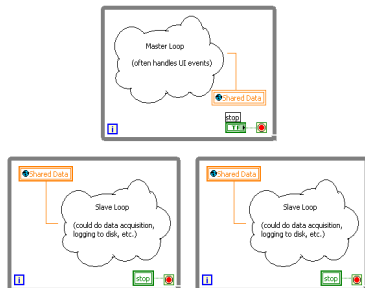
- W LabVIEW często nazywany specjalnym clusterem
- Dowlone dane zdefiniowane przez użytkownika
- Możliwość dziedziczenia obiektów



WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI  
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Pętle Master / Slave

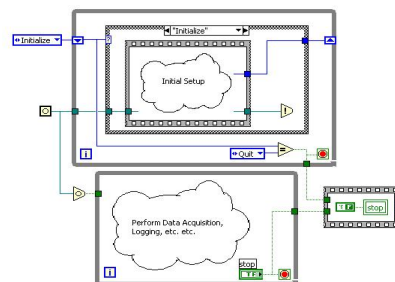
Używane w aplikacjach gdzie główny proces startuje kolejne asynchroniczne wykonania kodu



WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI  
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Master / Slave synchronizacja

Przykład programu z synchronizacją startu



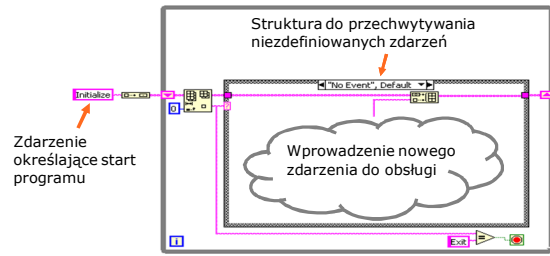
WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI  
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

## Kolejki – Queued Message Handler

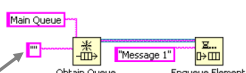
- Często traktowane jako alternatywny wzorzec podobny do struktury zdarzeń.
- Możliwość przesyłania więcej informacji niż w przypadku typowej maszyny stanów
- Używany głównie do przechwytywania zdarzeń w aplikacji i umieszczanie ich w kolejce zadań lub komunikatów dostępnych dla całej aplikacji. Rozwiązanie wygodne do synchronizowania działań w systemach wielowątkowych czy rozproszonych.
- Możliwość implementacji sieciowej co znacznie ułatwia synchronizację działań aplikacji sieciowych.

## Kolejki – Queued Message Handler



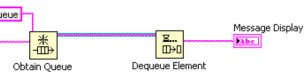
## Kolejkowanie danych

### Dodawanie elementów do kolejki

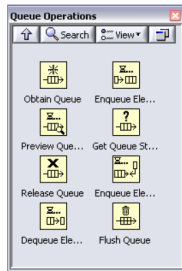


Typ danych wprowadzanych do kolejki:  
tekst, klastery, tablica, liczba ...

### Pobieranie elementów z kolejki



Pobieranie danych z kolejki może oczekiwać (wstrzymać działanie programu) na wypłynięciu do kolejki lub przejść dalej po określonym przez użytkownika czasie



## „Daemony” – usługi działające w tle



Używane jako aplikacje nadzorujące działanie danego programu czy systemu. Na ogół działają w tle i są niewidoczne dla użytkownika, ale mogą uruchamiać programy z interfejsem użytkownika lub informować użytkownika o pewnych zdarzeniach.

BSD Daemon Copyright 1988 by Marshall Kirk McKusick. All Rights Reserved.

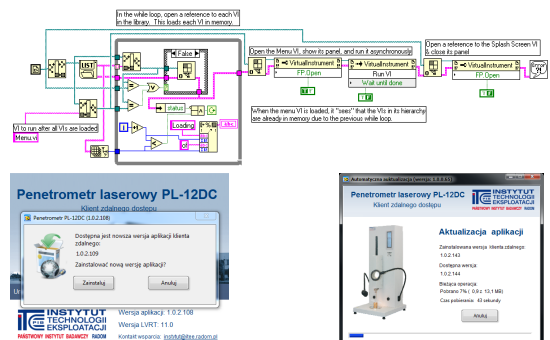
## Splash Screen (popularny Luncher)

Wykorzystywane do zastąpienia standardowego komunikatu o ładowaniu i uruchamianiu aplikacji własnym ekranem startowym.

Jest to zabieg kosmetyczny ale poprawia ogólne wrażenie o aplikacji. Informuje zazwyczaj użytkownika o nazwie, wersji i postępie ładowania aplikacji.

Zamyka się po zakończeniu procesu ładowania aplikacji.

## Splash Screen (Luncher) - przykład





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

## Podsumowanie

### Dlaczego powinno się używać wzorców projektowych?

- Są one wypracowane przez wiele osób i sprawdzone w wielu projektach o konkretnym zastosowaniu.
- Skracają czas wykonania aplikacji.
- Polepszają jakość kodu oraz często mają wpływ na wydajność i konsumpcję zasobów sprzętowych.

### Warto poświęcić chwilę na znalezienie i dostosowanie wzorca do swoich.

## 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 krok po kroku - budowa prostej aplikacji typu klient-serwer.

## Wstęp do programowania w środowisku LabVIEW

- Pierwszy program w języku G
- Przydatne informacje przyspieszające tworzenie programów w środowisku LabVIEW
- Przegląd najważniejszych funkcji matematycznych

## Pierwszy program

### Ćwiczenie do wykonania:

Napisać program realizujący dodawanie i odejmowanie dwóch liczb A i B.

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

## 1. Umieszczamy elementy na panelu

Elementy typu: Zadajnik (Numeric Control) i wskaźnik (Numeric Indicator)

Płyta czołowa - Front Panel

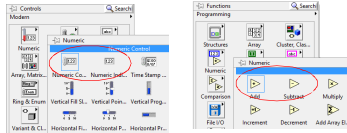
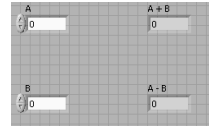
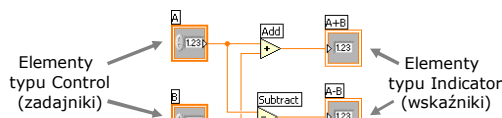


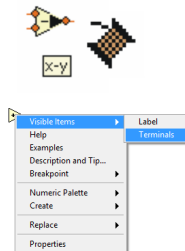
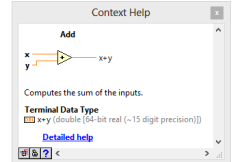
Diagram programu - Block Diagram



## 2. Tworzymy schemat blokowy

Po najechnaniu „szpulka” na terminal zostaje on podświetlony a po naciśnięciu przycisku myszy przyłączony zostaje początek ścieżki

Ctrl+H – pomoc kontekstowa



Włączenie podglądu terminali





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

### Tworzymy schemat blokowy

Narzędzie "Connect Wire"

Zaznaczanie połączeń poprzez kliknięcie

pojedyncze      podwójne      potrójne

Automatyczne łączenie końców

„Układanie” połączeń

lub z paska narzędzi

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

119

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Funkcje śledzenia wykonania programu

- Sposób sygnalizacji błędu:
- Wyszukiwanie błędów
  - Naciśnij przycisk startu programu w celu wyświetlenia okna zawierającego opisy błędów
- Śledzenie wykonania aplikacji
  - Włączenie-wyłączenie śledzenia przepływu danych w aplikacji. Wartości wyświetlane są na wyjściach węzłów
- Śledzenie wartości
  - Włączenie ciągłego śledzenia zmiennej (Probe). Wartość wyświetlona jest w okienku.

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

116

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Przepływ danych w programie

- Program **nie wykonuje się!** od lewej do prawej,
- Poszczególne funkcje realizują swoje operacje dopiero gdy na wszystkich wejściach są dostępne dane,
- Dane wyjściowe z węzła pojawiają się równocześnie na wszystkich wyjściach

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

117

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Funkcje matematyczne w LabVIEW

Dostęp do pogrupowanych tematycznie funkcji matematycznych z dwóch miejsc:

**Programming:**

- Numeric
- Boolean
- Comparison
- Array

**Mathematics:**

- Numeric
- Elementary
- Linear Algebra
- Fitting
- Interpolation & Extrapolation
- Integration & Differentiation
- Probability & Statistics
- Optimization
- Differential Equation
- Geometry
- Polynomial
- Script & Formulas

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

118

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Funkcje matematyczne - Numeric

Mnogosc funkcji – grupa konwersji, liczb zespolonych ...

Grupa podstawowa

Grupa konwersji typów

Manipulacja danymi

Predefiniowane stałe

Liczby zespolone

Skalowanie danych

Fixed-point

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

119

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Funkcje matematyczne - Boolean

Funkcje logiczne pozwalające realizować podstawowe operacje logiczne nawet i proste schematy elektroniki cyfrowej

Boolean

And, Or, Exclusive Or, Not, Compound ...

Not And, Not Or, Not Exclusiv..., Implies

And Array El., Or Array El., Num to Array, Array to Num, Bool to (0,1)

True Constant, False Constant

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI

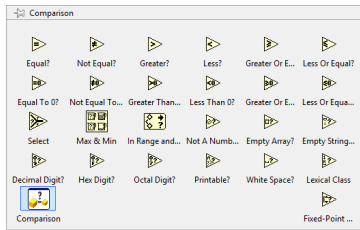
120

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

## Funkcje matematyczne - Comparison

Funkcje pozwalające realizować warunki logiczne

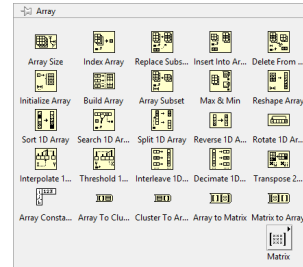


„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

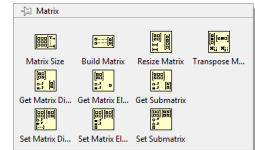
## Funkcje matematyczne - Array

Funkcje pozwalające realizować operacje na tablicach i macierzach – omówione szerzej w dalszej części.

Operacje na tablicach 1,2 i n wymiarowych



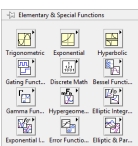
Operacje na macierzach



Operacje z pozostałych grup np. numeric dostosowują się automatycznie (polimorfizm funkcji) w momencie operacji na tablicach

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Funkcje matematyczne - Elementary

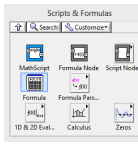


W grupie funkcji podstawowych znajdują się:

- Funkcje trygonometryczne
- Funkcje eksponentialne
- Funkcje hiperboliczne
- Funkcje jednostkowe
- Funkcje związane z matematyką dyskretną
- Funkcje Bessela
- Funkcje gamma
- Funkcje hipergeometryczne
- Całki eliptyczne, eksponentialne
- Funkcje do obliczania błędów
- Funkcje Eliptyczne i paraboliczne

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Funkcje matematyczne – skrypty i formuły



Najbardziej zaawansowaną grupą narzędzi matematycznych są elementy takie jak MathScript

```
input variable (optional)
input variable (optional)
input variable (optional)
error in (optional)
SumA = eye(size(A));
for i = 1:n
    SumA = SumA + A*(factorial(i)); Delta = output variable (optional)
end
Delta = SumA - expm(A); error out
```

Służące do wykonywania obliczeń w znanym języku tekstowym z Matlab'a. Drugim modułem wywołującym program Matlab do obliczeń jest moduł MATLAB Script Node.

Wygodnym narzędziem jest też Formula Node interpretująca skrypt w języku C.

```
input variable (optional)
int(32 y;
if(x>=0)
y = 1;
else y = -1;
); output variable (optional)
```

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Grupowanie danych – struktury (clusters)

Co to są struktury danych (clusters)?

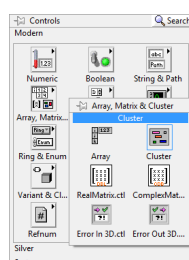
- Są to struktury pozwalające zgrupować różne dane i przesyłać je razem (porządek na diagramie).
- Są odpowiednikami struct w C i pozwalają grupować dane różnych typów.
- Dane muszą być jednego rodzaju Indicator lub Control



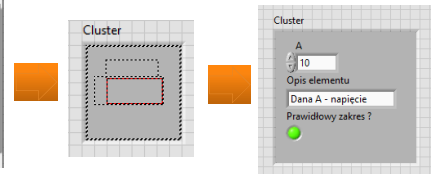
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Cluster

Budowanie elementu typu cluster:



Umieszczamy kontrolki jakie chcemy aby znalazły się w strukturze



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Struktura do wyświetlania błędów

**Element Error cluster -> Controls»Array & Cluster**

The image shows two 'Error Cluster' control panels. The left one has 'status' (green checkmark), 'code' (0), and 'source' (empty). The right one has 'status' (red X), 'code' (0), and 'source' (empty). A wiring diagram shows a 'true' value connected to the 'error in (no error)' input of an 'Error Cluster' control, which is then connected to the 'error out' output.

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI  
127

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Informacja o błędzie

- **Status** - wartość typu Boolean zawierająca TRUE jeśli w aplikacji wystąpił błąd.
- **Code** - zawiera numer błędu.
- **Source** - zawiera miejsce wystąpienia błędu.

The diagram shows a control with three fields: 'status' (Boolean), 'code' (Number), and 'source' (Text).

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI  
128

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Obsługa błędów - przykład

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

The diagram shows a 'File Path' control connected to an 'Open/Create/Replace File' function. The 'error in' output of the function is connected to an 'Error' control. The 'error out' output of the 'Error' control is connected to a 'Bundle By Name' control, which outputs 'Podano liczbę ujemną'.

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI  
129

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Wyświetlanie informacji o błędzie

**Dialog z informacją o błędzie w aplikacji.**

Simple Error Handler -> Functions » All Functions » Time and Dialog palette

The diagram shows an 'Error Cluster' control connected to an 'Error' dialog control. The 'error?' output of the dialog is connected to the 'status' field of the error cluster. The 'code out' and 'source out' outputs of the dialog are connected to the 'code' and 'source' fields of the error cluster.

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI  
130

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Obsługa błędów

**Diagram aplikacji wczytującej dane.**

The diagram shows a 'File Path' control connected to an 'Open/Create/Replace File' function. The 'error in' output of the function is connected to an 'Error' control. The 'error out' output of the 'Error' control is connected to a 'Simple Error Handler' control. The 'status' output of the 'Simple Error Handler' is connected to a 'status' control.

Można też sterować pętlą ...  
(zatrzymaj jeśli pojawił się błąd)

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI  
131

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Operacje na zbiorach

The background features the text 'WIRTUALNE LABORATORIA FIZYCZNE' and 'OPERACJE NA ZBIORACH' in a stylized font.

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

WARSAWSKA WYŻSZA SZKOŁA INFORMATYKI  
132


„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Tablica wartości stałych


#### Tworzenie tablicy wartości stałych

Z grupy funkcji Array wybieramy Array Constant




Następnie dodajemy typ danych jaki ma być przechowywany w tablicy (wkładamy go w pole Array Constant).

Tablica z elementami typu double




Tablica z elementami typu boolean




Następnie określamy rozmiar tablicy poprzez rozciągnięcie za jeden z rogów do wymaganej ilości pól.

Tablica 1 elementowa



Tablica 5 elementowa

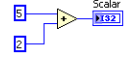


**133**

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Tablice – operacje na tablicach

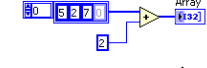
Skalar + Skalar



wynik skalar


Użycie funkcji polimorficznej do wykonania operacji na tablicy

Tablica + Skalar




wynik tablica

Tablica + Tablica



wynik tablica

Liczba elementów tablicy



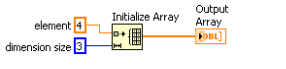
size(s)

**134**

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

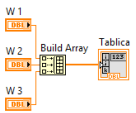
### Tablice – operacje na tablicach

Zainicjowanie tablicy n elementami o określonej wartości



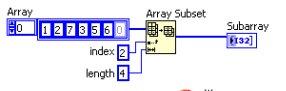
Output Array

Utworzenie tablicy z 3 wartości skalarnych



Tablica

Wybranie n elementów od indeksu i z tablicy



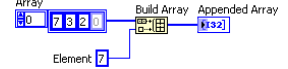
Subarray

**135**

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

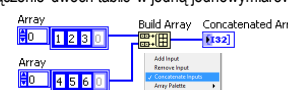
### Tablice – operacje na tablicach

Dodawanie elementu na końcu tablicy



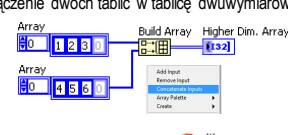
Appended Array

Łączenie dwóch tablic w jedną jednowymiarową



Concatenated Array

Łączenie dwóch tablic w tablicę dwuwymiarową – wyłączona opcja Concatenate Inputs



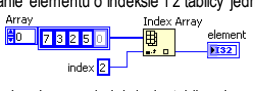
Higher Dim. Array

**136**

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

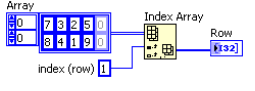
### Tablice – operacje na tablicach

Pobieranie elementu o indeksie i z tablicy jednowymiarowej



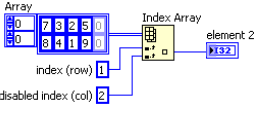
element

Pobieranie wiersza o indeksie i z tablicy dwuwymiarowej



Row

Pobieranie elementu o indeksie (r,c) z tablicy dwuwymiarowej



element 2

**137**

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Tworzenie tablic – przykładowy program

#### Tworzenie tablic Build Array (płyta czołowa)



**138**

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”



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

### Tworzenie tablic

#### Tworzenie tablic Build Array (diagram).

Liczba wejść ustalana jest dynamicznie

Elementy wejściowe są różnych rozmiarów (tablica, skalar), dlatego opcja łączenia wejść jest domyślnie zaznaczona i nieaktywna. W przypadku łączenia tablic, użytkownik może zdecydować czy na wyjściu ma być złożenie czy tablica wielowymiarowa o podanych wierszach.

139

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Podstawowe operacje na plikach

140

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### LabVIEW - przechowywanie danych

Dane można zapisywać na wiele sposobów i w wielu formatach. Problemem jest wybór odpowiedniego formatu. Format zapisu zdeterminowany jest przez system dalszej obróbki oraz przez ilość danych do zapisu.

LabVIEW zapewnia obsługę wielu domyślnych formatów oraz obsługę plików binarnych i tekstowych umożliwiającą zaprogramowanie praktycznie dowolnego formatu.

141

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Odczyt i zapis danych do pliku

W LabVIEW użytkownik ma do dyspozycji wiele funkcji począwszy od popularnego CSV do przechowywania danych tabelarycznych (SpreadSheet) poprzez specjalizowane:

- TDM – specjalny samo opisujący się format danych przeznaczony do streamingu danych pomiarowych
- Meas File – wewnętrzny tekstowy format do zapisu sygnałów pomiarowych
- ZIP – skompresowane archiwa plików
- XML – tekstowy plik XML przeznaczony raczej do statycznych danych takich jak pliki konfiguracyjne
- WaveForm File – dynamiczne struktury danych otrzymywane np. z kart pomiarowych.

142

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Odczyt i zapis danych do pliku

#### Schemat działania:

LabVIEW upraszcza pisanie kodu do minimum – można pominąć instrukcje open i close. Operacje te zostaną wykonane automatycznie przy próbie dostępu i zakończeniu ścieżki pliku.

143

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Zapis danych

#### Zapis danych tekstowych do pliku – przykład 1

Jeżeli nie podamy na wejściu funkcji ścieżki do pliku funkcja Open lub w przypadku nie używania funkcji Open funkcja Write wyświetli okno z zapytaniem o plik. Komunikat można podać na wejście – prompt funkcji

144

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Odczyt danych

#### Odczyt danych tekstowych z pliku – przykład 2

145

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Instrukcje sterujące i pętle

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

146

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Pętla for loop

Przykład pętli for loop w LabVIEW

Schemat działania

Pętla for w notacji języka C

```
for(i = 0 ; i < n ; i++ )
{
    //treść pętli
}
```

Pseudo Code

```
N=100;
i=0;
Until i=N:
    Repeat (code; i=i+1);
End;
```

147

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Pętla for

#### Pętla for w konwencji języka G (LabVIEW)

Zmienna **n** definiuje liczbę iteracji wykonywanych w ramach pętli

Zmienna **P** określa ilość wątków jeżeli istnieje możliwość równoległego wykonania obliczeń

Zmienna **i** przechowuje numer bieżącej iteracji

Treść pętli

Terminal zatrzymania warunkowego

148

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Pętla for loop – przykładowy program

1. Dodanie tablicy (Array) z Controls palette

2. Dodanie elementów do tablicy

„Rozciągamy tablicę” do minimum 10 elementów

Używamy opcji Add Dimension – jeżeli chcemy zrobić tablicę dwuwymiarową (2D arrays)

149

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Pętla for i tablica – przykładowy program

#### Aplikacja wykorzystująca pętlę for oraz tablicę jednowymiarową

Pętla ustawiona na 15 iteracji

Zatrzymanie pętli po 10 wartości

Indeksowanie włączone

Indeksowanie wyłączony

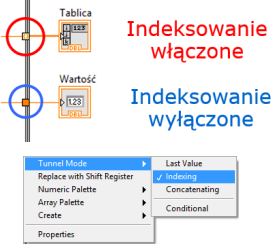
150

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Pętla for i wyjście danych z pętli

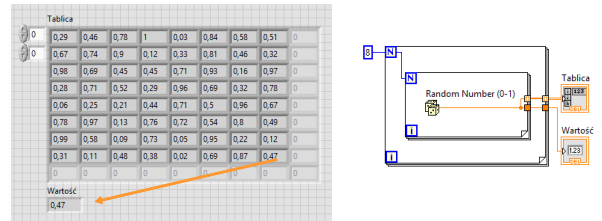
Sposób wyjścia danych z pętli określamy poprzez kliknięcie myszą na terminalu wyjściowym i wybierając jedną z dostępnych opcji:



- Last Value - ostatnia wartość
- Indexing – włączone auto indeksowanie ( na wyjściu jest budowana tablica)
- Concatenating – dołączanie do istniejącej np. tablicy
- Conditional – wyjście warunkowe

### Pętla for i tablica dwuwymiarowa

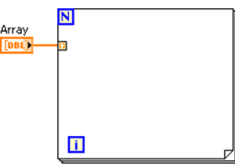
#### Aplikacja wykorzystująca pętlę for oraz tablicę dwuwymiarową



### Tablice – auto indeksowanie wejścia

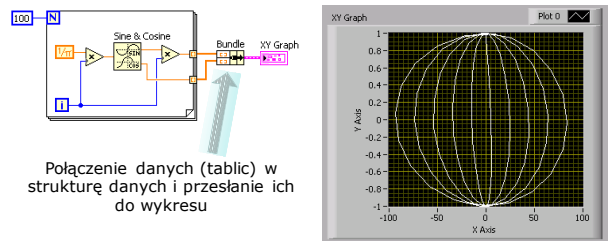
Tak samo jak auto indeksowanie wyjścia jest możliwe auto indeksowanie danych wejściowych z tablicy.

Jeżeli na wejściu podpięte są dwie tablice o różnej wielkości, to LabVIEW zawsze wybierze jako liczbę N wielkość mniejszej tablicy



### Pętla for loop – przykład użycia

#### Wyświetlanie wielu przebiegów - Przykład 2.



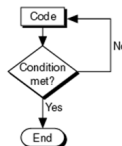
Połączenie danych (tablic) w strukturę danych i przesłanie ich do wykresu

### Pętla While

Pętla while loop w LabVIEW



Schemat działania

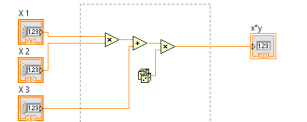
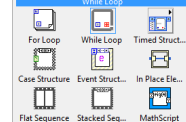


Pętla while w notacji języka C

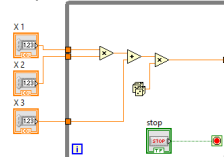
```
do {
} while (Condition)
Pseudo Code
Repeat (code);
Until Condition met;
End;
```

### Dodawanie pętli while na schemacie

1. Wybranie pętli While Loop
2. Otoczenie elementów które mają wykonywać się w pętli



3. Uzupełnienie schematu o elementy sterujące



W tym przypadku jest to przycisk STOP zatrzymujący wykonywanie się pętli

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

### Wyrażenie warunkowe w pętli while

Domyślna wartość

Warunek stopu      Licznik iteracji      Warunek kontynuacji

..Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego" 157

### Wejście i wyjście danych z pętli

**Wyjścia danych z pętli nazywane są tunelami i przybierają kolor zgodny z typem danych.**  
Wyróżniamy:

- tunele wejściowe (pętla zostanie wykonana dopiero gdy dane wpłyną do pętli)
- wyjściowe (zostaną udostępnione dopiero po zakończeniu pętli)

..Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego" 158

### Instrukcje sterujące – case structure

**Case structure – prosta konstrukcja pozwalająca wykonać fragment kodu w zależności od wartości wyrażenia sterującego – w tym przypadku wartości logicznej (true-false)**

..Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego" 159

### Instrukcje sterujące – case structure

**Konstrukcja case structure może być sterowana różnymi elementami – od logicznych przez numeryczne aż po tekstowe.**

Pozycja default oznacza wykonanie kodu dla pozycji spoza listy zaprogramowanych wartości

..Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego" 160

### Instrukcje sterujące – select

**Prosta instrukcja sterująca „select” umożliwiająca wybieranie ścieżki przyływu danych (dowolnych: numeryczne, tekstowe, struktury aż po dynamiczne strumienie) za pośrednictwem selektora binarnego (true, false)**

..Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego" 161

### Instrukcje sterujące – formula node

**Wykorzystanie Formula node jako złożonej instrukcje warunkowej**

..Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego" 162

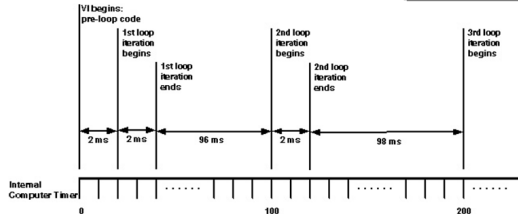
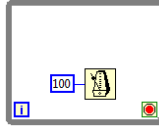
Opracowane w ramach projektu: "Wirtualne Laboratoria Fizyczne nowoczesną metodą nauczania" realizowanego przez Warszawską Wyższą Szkołę Informatyki.

### Funkcje sterujące opóźnieniem wykonania

#### Funkcja Wait Until Next ms Multiple



Functions » Time & Dialog palette



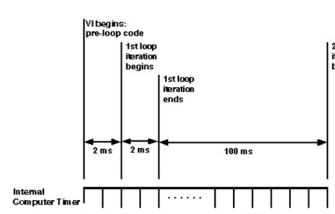
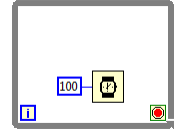
### Funkcje sterujące opóźnieniem wykonania

#### Funkcja Wait (ms) oraz Time Delay



Functions » Time & Dialog palette

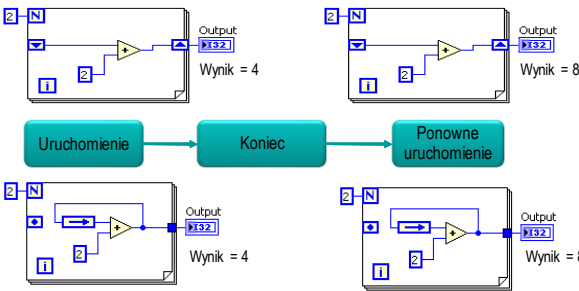
Functions » Time & Dialog palette



### Rejestry przesuwne i sprzężenia

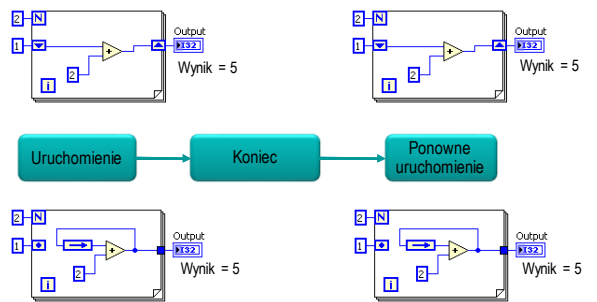
Rejestr przesuwny umożliwia dostęp do danych z poprzedniej lub z poprzednich iteracji pętli.

#### Brak inicjalizacji wartości rejestrów



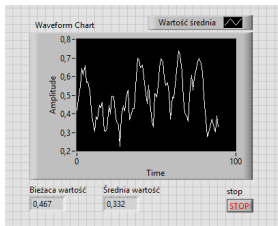
### Rejestry przesuwne i sprzężenia

#### Wstępna inicjalizacja wartości rejestrów



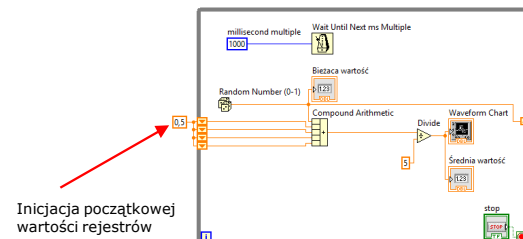
### Rejestry przesuwne – przykład 1

Generacja przebiegu losowego z uśrednianiem - demonstracja rejestrów przesuwnych (plyta czołowa).



### Rejestry przesuwne – przykład 1

Generacja przebiegu losowego z uśrednianiem - demonstracja rejestrów przesuwnych (diagram).

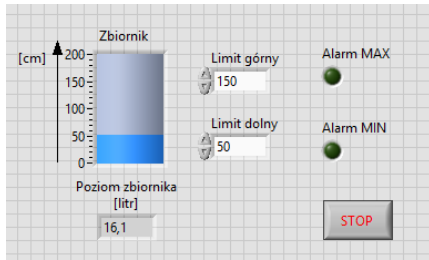




Opracowane w ramach projektu: "Wirtualne Laboratoria Fizyczne nowoczesną metodą nauczania" realizowanego przez Warszawską Wyższą Szkołę Informatyki.

## Pętle i instrukcje sterujące – przykład 2

Zbudować interfejs aplikacji jak na slajdzie

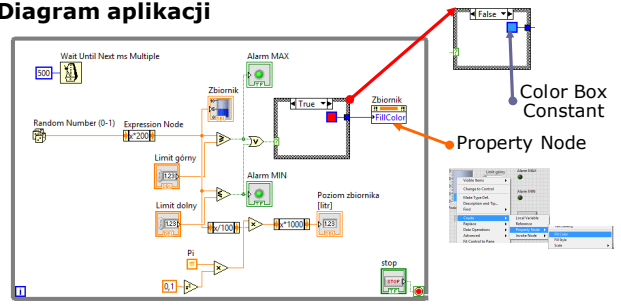


KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY 169

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Pętle i instrukcje sterujące – przykład 2

Diagram aplikacji



KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY 170

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Ciągi tekstowe



KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY 171

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Ciągi tekstowe

Specjalne pole do obsługi ścieżek do plików i katalogów zaopatrzone w okno dialogowe do wyboru ścieżki

Specjalny format wyjściowy: Path. Wymaga konwersji na string

Normal display

Password display

'\` code display

Hex display

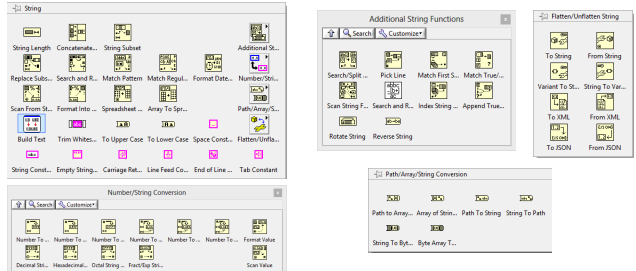
Specjalna kontrolka zawierająca ciągi tekstowe do wyboru przez użytkownika

KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY 172

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Funkcje związane z ciągami tekstowymi

LabVIEW udostępnia szereg funkcji do operacji na ciągach tekstowych począwszy od łączenia, przeszukiwania aż po konwersję na typy liczbowe.

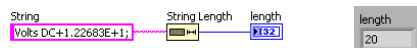


KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY 173

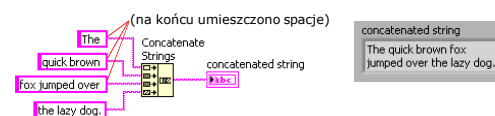
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Podstawowe operacje na ciągach tekstowych

Długość ciągu tekstowego



Łączenie ciągów tekstowych



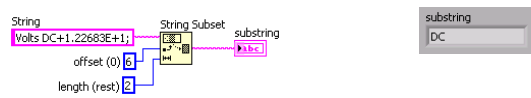
KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY 174

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

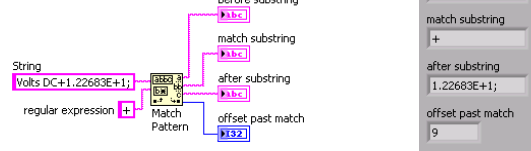
Opracowane w ramach projektu: "Wirtualne Laboratoria Fizyczne nowoczesną metodą nauczania" realizowanego przez Warszawską Wyższą Szkołę Informatyki.

### Podstawowe operacje na ciągach tekstowych cd.

Podciąg z ciągów tekstowych

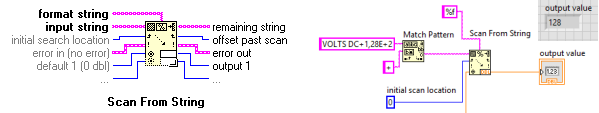


Wyszukiwanie wzorca



### Podstawowe operacje na ciągach tekstowych cd.

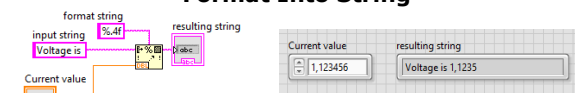
Wyszukiwanie danych z łańcucha znakowego i konwersja do zmiennej numerycznej



### Operacje na łańcuchach znakowych cd.

Budowanie odpowiedzi tekstowej

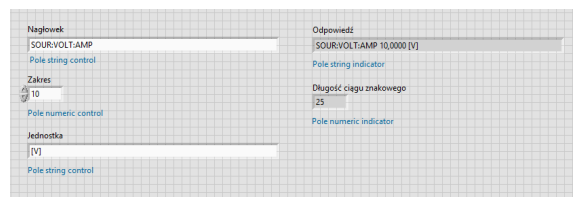
Format Into String



Ciąg formatujący jest zbudowany w oparciu o zasady znane z języka C. Umożliwia konwersję danych numerycznych na ciąg znaków reprezentujący wartość z uwzględnieniem formatowania (dane numeryczne, daty, liczby szesnastkowe itp.).

### Formatowanie złożonych łańcuchów

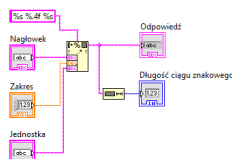
Ćwiczenie 1 - płyta czołowa



### Formatowanie złożonych łańcuchów

diagram połączeń

Funkcja **Format Into String** z polimorficznym wejściem rozszerzona na 3 elementy



WIRTUALNE  
LABORATORIA  
FIZYCZNE

POJĘCIE FUNKCJI W LABVIEW

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

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

### Co to jest funkcja (SubVI)?

**Funkcja zapisana w pseudo kodzie**

```
function average (in1, in2, out)
{
    out = (in1 + in2)/2.0;
}
```

**Wywołanie funkcji - pseudo kod**

```
main
{
    average (point1, point2, pointavg)
}
```

**SubVI - Block Diagram**

**Wywołanie SubVI - Block Diagram**

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Konektory i terminale w SubVI

**Ikona z terminalami**

terminal wejściowy i wyjściowy z funkcji

**Ikona**

Ikona reprezentuje graficznie funkcję (SubVI'a) na diagramie

**Widok konektorów**

Terminale reprezentują wejścia i wyjścia danych z funkcji (SubVI'a)

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Pojęcie funkcji - przykład

**Ćwiczenie do wykonania:**

1. Budowa programu przeliczającego stopnie Celsjusza na Fahrenheita.
2. Stworzenie SubVI z programu z punktu 1
3. Zbudowanie aplikacji wykorzystującej SubVI.

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Program przeliczający stopnie C na F

**Umieszczenie na panelu kontrolki typu:**

1. Numeric Control (Deg C)
2. Numeric Indicator (Deg F)

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Program przeliczający stopnie C na F

**Budowa schematu blokowego:**

**Zmiana sposobu wyświetlania kontrolki**

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Budowa SubVI

**Zapisanie stworzonego programu (DegC\_to\_DegF.vi)**

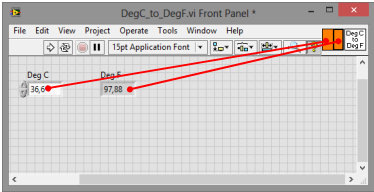
**Utworzenie ikony dla SubVI**

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Budowa SubVI

**Podłączanie konektorów (narzędzie szpulka)**



187

..Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego"

### Aplikacja wykorzystująca SubVI

**Budowa interfejsu, potrzebne elementy:**

1. Numeric -> Numeric Control
2. Boolean -> Vertical Toggle Switch
3. Numeric -> Thermometer

Włączona opcja Digital Display



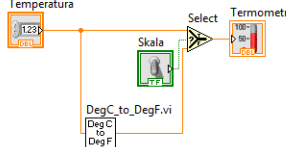
188

..Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego"

### Aplikacja wykorzystująca SubVI

**Budowa schematu blokowego:**

**Wstawienie SubVI do diagramu**  
All Functions » Select a VI...  
<LUB>  
Przeciagnąć ikonkę z drugiego okna na schemat

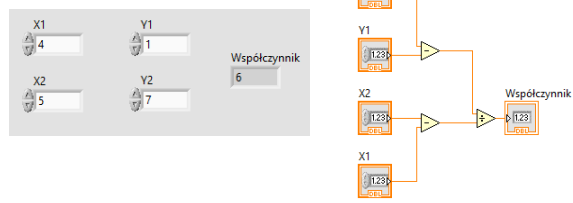


189

..Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego"

### Budowa SubVI - Przykład 2

**Budowa SubVI**



190

..Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego"

### Graficzna reprezentacja SubVI

**Ikona domyślna**



**Ikona zmodyfikowana**

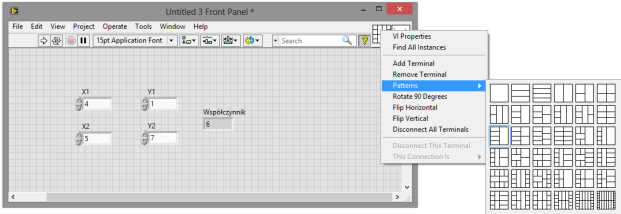


191

..Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego"

### Konektory

**Wybieramy wzorec konektorów najlepiej odwzorowujący strukturę danych wejściowych i wyjściowych z funkcji**



192

..Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego"

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

### Konektory – wejścia \ wyjścia

**Podłączamy dane wejściowe i wyjściowe**

193

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Generacja przebiegów i ich wyświetlanie

194

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Gotowe kontrolki do wyświetlania wykresów

Środowisko LabVIEW oferuje dużą liczbę gotowych kontrolkek umożliwiających wyświetlanie wykresów 2D i 3D. Każdą z kontrolkek można dostosować graficznie do własnych potrzeb

Jeżeli istnieje potrzeba zbudowania wykresu który nie jest dostępny pod postacią gotowej kontrolki z pomocą przychodzi obiekt Image który pozwala wyrysować praktycznie dowolny wykres.

195

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Wykresy 2D i 3D

#### Wykresy 3D

- Contour
- Mesh
- Quiver
- Stem
- Surface
- Comet
- Pie
- Scatter
- Ribbon
- Waterfall
- Bar

#### Wykresy 2D

- Compass
- Feather
- Errorbar
- Plotmatrix

196

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Wizualizacja danych – przykład 1

Generacja przebiegu losowego z uśrednieniem  
demonstracja wykresu Waveform Chart (plyta czołowa)

197

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Wizualizacja danych – przykład 1

Generacja przebiegu losowego z uśrednieniem  
demonstracja wykresu Waveform Chart (diagram)

198

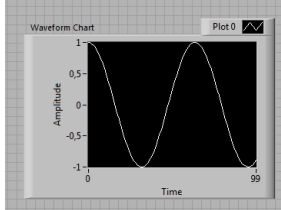
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”



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

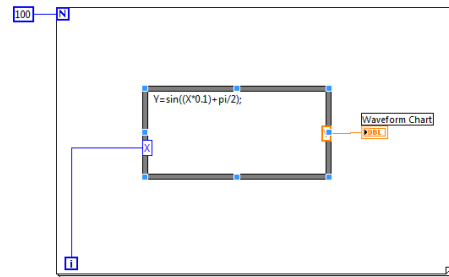
### Wizualizacja danych – przykład 2

Generacja przebiegu sinusoidalnego opisanego wzorem  
 $Y(t)=\sin((t*0.1)+\pi/2)$   
demonstracja wykresu Waveform Chart (plyta czołowa)

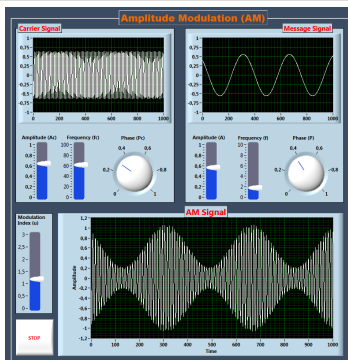


### Wizualizacja danych – przykład 2

Diagram: wykorzystanie Formula Node do wygenerowania sygnału sinusoidalnego



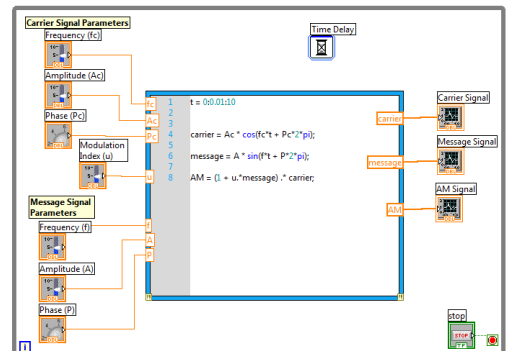
### Wizualizacja danych – przykład 3



Przykład wykorzystania Wykresów Waveform Grpah oraz modułu MathScript do prezentacji zjawiska modulacji sygnału AM

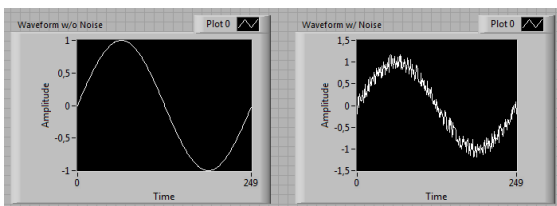
### Wizualizacja danych – przykład 3

Diagram



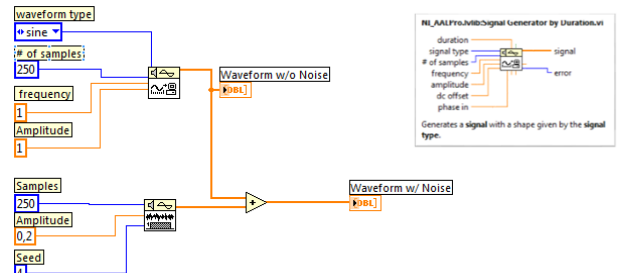
### Generacja sygnału

Wykorzystanie standardowych blozków do generacji sygnałów



### Generacja sygnału

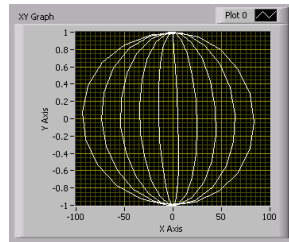
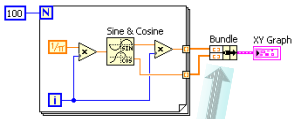
Wykorzystanie standardowych blozków do generacji sygnałów



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

### Wyświetlanie dwóch przebiegów

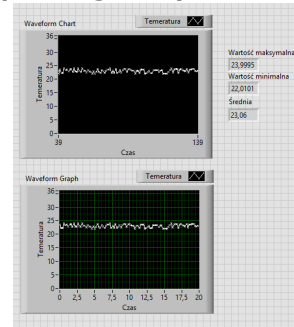
**Generacja sygnału i wyświetlanie wielu przebiegów z wykorzystaniem kontrolki XY Graph.**



Połączenie danych (tablic) w strukturę danych i przesłanie ich do wykresu

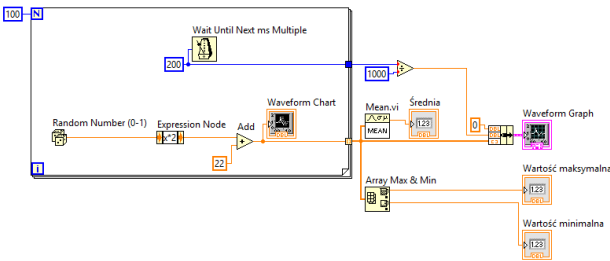
### Wizualizacja danych – przykład 4

**Skalowanie przebiegu – wyskalowanie osi X**



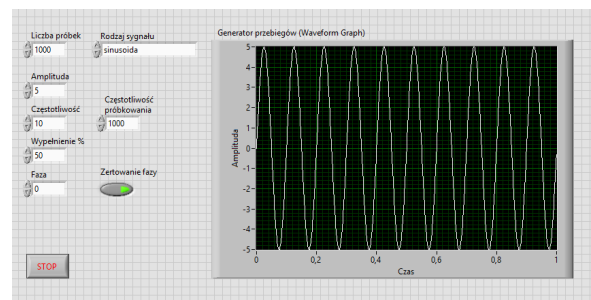
### Wizualizacja danych – przykład 4

**Diagram programu**



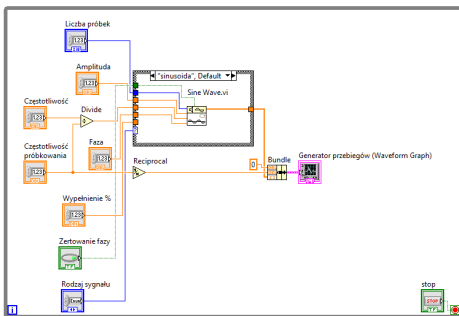
### Generacja sygnału i jego wyświetlanie

**Generator przebiegów okresowych (płyta czołowa).**



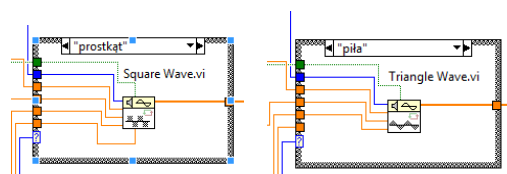
### LabVIEW w przykładach

**Diagram dla przebiegu sinusoidalnego.**



### LabVIEW w przykładach

**Pozostałe przebiegi (diagram elementu case).**



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

## Komunikacja sieciowa w LabVIEW

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

211

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Komunikacja sieciowa – model TCP/IP

212

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Komunikacja w środowisku LabView

- Komunikacja niskiego i wysokiego poziomu**

<p><b>Data Transfer</b></p> <ul style="list-style-type: none"> <li>• Shared Variable</li> <li>• OPC</li> <li>• DSTP</li> <li>• Modbus</li> <li>• EPICS</li> <li>• CANopen</li> <li>• PROFIBUS</li> <li>• Ethernet/IP</li> </ul>	<p><b>Deterministic Data Transfer</b></p> <ul style="list-style-type: none"> <li>• Time-Triggered Variable</li> <li>• EtherCAT</li> </ul>	<p><b>Remote Procedure Calls</b></p> <ul style="list-style-type: none"> <li>• VI Server</li> <li>• LabVIEW Web Services</li> </ul>
<p><b>Miscellaneous</b></p> <ul style="list-style-type: none"> <li>• Custom Protocols with DLLs</li> <li>• Remote Panels</li> </ul>		<p><b>Networked Object Modules</b></p> <ul style="list-style-type: none"> <li>• ActiveX</li> <li>• .NET</li> </ul>

**Low Level**

• TCP • UDP • Bluetooth • Serial • IrDA

213

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Komunikacja na „niskim poziomie”

- Umożliwia dostosowanie do własnych potrzeb.
- Zgodna z większością standardów sieciowych.
- Szybka i wydajna (oczywiście zależy od implementacji).
- Czasochłonna w implementacji i optymalizacji działania.

214

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Komunikacja wysokiego poziomu

- Upraszcza proces kodowania aplikacji
- Dostęp do sieci realizowany jest poprzez funkcje wysokiego poziomu, zoptymalizowane pod kątem wydajności w danym zastosowaniu.
- Duża ilość przykładów pozwalających na rozwiązanie większości zadań bez konieczności wynalezienia koła od nowa.
- Są to sprawdzone i szeroko stosowane rozwiązania w przemyśle.

215

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Jak wybrać właściwy protokół?

- **Identyfikacja potrzeb aplikacji i systemu:**
  - Przejrzystość kodu,
  - Kryterium wydajnościowe,
  - Skalowalność aplikacji
  - Zgodność z określonymi standardami,
  - Dostępność.
- **Właściwy wybór wzorca znacznie upraszcza i przyspiesza proces wytwarzania aplikacji.**
- **Zły wybór API i protokołu może znacznie skomplikować projekt i sposób wykonania, a tym samym odbija się na jakości kodu i na wydajności**

216

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

## Protokoły niskiego poziomu

- Protokół UDP
- Protokół TCP
- TCP – Simple TCP/IP Messaging Protocol
- Bluetooth

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

217

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Protokół UDP

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

218

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## UDP - Datagramowy Protokół Użytkownika

- **Zalety:**
  - Asynchroniczny
  - Bezpołączeniowy
  - Wydajny
  - Wspiera multicasting
  - Skalowalny
- **Wady:**
  - Brak gwarancji dostarczenia
  - Dane muszą być konwertowane do string
  - W przypadku przesyłania skomplikowanych danych wymaga dużego nakładu programowania do zbudowania konwerterów z jednej postaci na string i w drugą stronę.
  - Nie współpracuje z cienkim klientem.

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

219

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## UDP - działanie

Zasada działania komunikacji opartej o UDP jest bardzo prosta – problemy zaczynają się gdy zachodzi potrzeba przesłania skomplikowanych struktur danych. Działanie komunikacji UDP polega na:

- Określeniu adresu IP i portu odbiorcy wiadomości.
- Nadawca otwiera port i wysyła dane.

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

220

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Kiedy używać UDP

- Do przesyłania danych strumieniowych
- W aplikacjach wykorzystujących multicast
- Tam gdzie potrzeba szybkiego transferu
- Własny protokół
- Do komunikacji LabVIEW z oprogramowaniem firm trzecich (często stosowane ☹ )
- Komunikacja w sieciach WAN.

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

221

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Protokół TCP

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

222

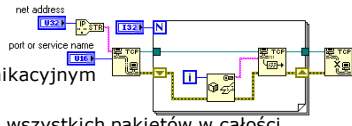
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Protokół TCP

#### Zalety:

- Strumieniowy
- Jest standardem komunikacyjnym
- Wysoka wydajność
- Gwarancja dostarczenia wszystkich pakietów w całości.

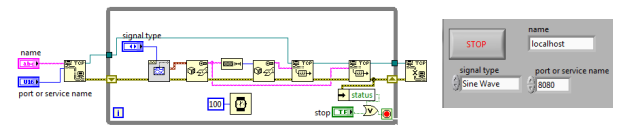


#### Wady:

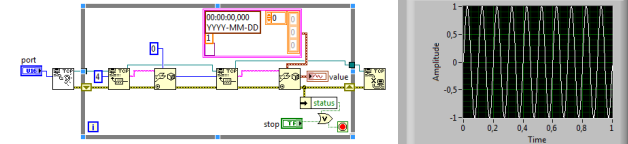
- Nastawiony na komunikację typu klient-serwer (jeden w jeden), w przypadku użycia wielu klientów wymaga pisania specjalnego kodu.
- Konieczność tłumaczenia wysyłanych i odbieranych danych z i do ciągów tekstowych.
- Oprogramowanie komunikacji sieciowej opartej o TCP wymaga znacznych nakładów pracy gdy chcemy przesyłać rozbudowane struktury danych.

### Przykład użycia

#### Nadawca:



#### Odbiorca:



### Jak działa TCP?

#### Polega ona na:

- Nawiązaniu połączenia z odbiorcą wiadomości adresowanym adresem IP oraz portem.
- Zaakceptowaniu bądź odrzuceniu połączenia przez odbiorcę.
- Wysłaniu lub odczytaniu danych.
- Zakończeniu połączenia.

**Połączenie TCP może znajdować się tylko w jednym z kilku predefiniowanych stanów.**

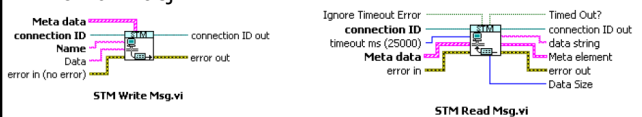
### Kiedy używać TCP?

- Do przesyłania danych strumieniowych
- W aplikacjach wymagających potwierdzenia dostarczenia pakietów, np. wszelkiego rodzaju aplikacje sterujące.
- Konieczność implementacji własnego protokołu.
- Do komunikacji LabVIEW z oprogramowaniem firm trzecich (często stosowane ☺).
- Komunikacja w sieciach WAN.

### TCP – Simple TCP/IP Messaging Protocol

### TCP – Simple TCP/IP Messaging Protocol

- Oparty na protokole TCP model pozwalający na przesyłanie komunikatów pomiędzy aplikacjami klienta i serwera.
- Wykorzystywany do współdzielenia danych pomiędzy serwerem a klientem.
- Gotowe funkcje w LabVIEW wspierające ten typ komunikacji.





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

**Bluetooth**

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY 229

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

**Komunikacja bluetooth**

- Brak konieczności używania dodatkowego osprzętu sieciowego.
- Funkcja rozpoznawania urządzeń będących w zasięgu.
- Wykorzystywane w LabVIEW do obsługi i diagnozy urządzeń
- Wireless Personal Area Network – sieć bezprzewodowa służąca do przesyłania danych pomiędzy ręcznym urządzeniem a stacjonarnym komputerem bądź do komunikacji z przyrządem pomiarowym w celu diagnostyki.

KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY 230

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

**Niskopoziomowy bluetooth**

- LabVIEW posiada protokół RFCOMM pozwalający na wirtualną szeregową komunikację pomiędzy urządzeniami.
- Prosty binarny protokół komunikacyjny.
- Programowanie podobne do programowania komunikacji opartej o TCP

KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY 231

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

**Protokoły wysokiego poziomu**

- Network-Published Shared Variable
- Datasocket server
- VI Server
- WebServices

KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY 232

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

**Network-Published Shared Variable**

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY 233

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

**NPSV - właściwości**

- Używany do komunikacji pomiędzy komponentami w LabVIEW
- Skalowalny
- Zintegrowany ze wszystkimi komponentami w LabVIEW
- Abstrakcyjny mechanizm komunikacji – ukryty przed programistą
- Umożliwia przesyłanie prawie wszystkich typów danych.
- Każda aplikacja napisana w środowiskach NI może być klientem.

KAPITAŁ LUDZKI WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY 234

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”



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

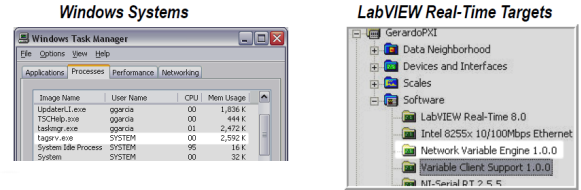
### NPSV – właściwości cd.

- Umożliwia komunikację z wieloma klientami naraz.
- Podłączanie bezpośrednio do funkcji lub kontrolki na interfejsie użytkownika
- Funkcjonalność przemysłowego serwera OPC.
- Współpracuje z aplikacjami napisanymi dla platform czasu rzeczywistego.
- Współpraca z LabVIEW DataLogging i Supervisory Control Module
- Wady:
  - Opatentowany przez NI – problem z użyciem w innych aplikacjach lub przy integracji systemów.
  - Nie jest zoptymalizowany pod kątem streamingu danych.
  - Konsumuje nieco więcej zasobów niż TCP/UDP

### NPSV – jak działa?

Dostępny jako serwis w systemie publikujący zasoby

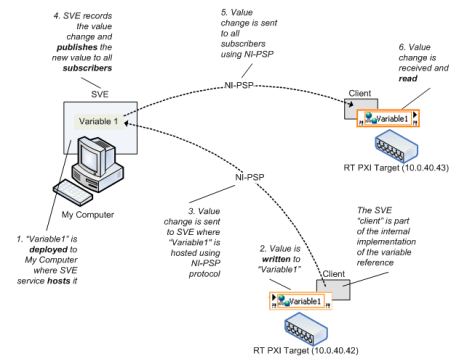
- Tagsrv.exe – silnik npsv.
- Działa także w systemach czasu rzeczywistego
- Wymaga co najmniej 32 MB pamięci operacyjnej – zlecane 64MB ☺



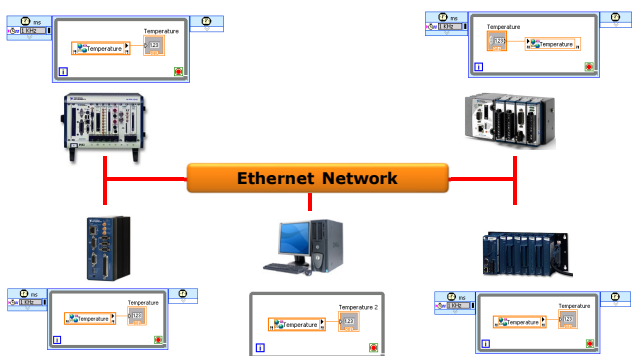
### Shared Variable – NI-PSP

- NI-PSP – pewna odmiana NPSV polegająca na tym że zasoby są publikowane a aplikacja klienta je subskrybuje.
- Używa TCP
- Zaprojektowane w celu zwiększenia efektywności komunikacji w stosunku do NPSV.
- Zasubskrybowane uaktualnienia są pakowane do jednej paczki i wysyłane do subskrybenta – oszczędność pasma.
- Uaktualnienia wysyłane jedynie dla elementów których wartości uległy zmianie w stosunku do ostatniej przesyłki.
- Zaprojektowana z myślą o dużych rozproszonych systemach pomiarowych współpracujących z HMI/SCADA
- Możliwość integracji z serwerami OPC i Modbus – oprogramowanie firm trzecich

### NI-PSP – jak działa ?

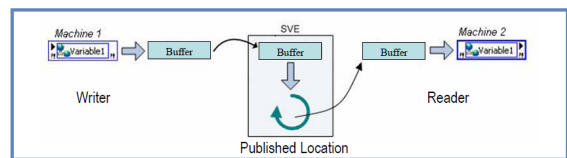


### NI-PSP – jak działa ?



### NI-PSP - buforowanie

- Dostępne jako opcja przy konfiguracji elementu
- Możliwość stworzenia wielu buforów
- Używane w celu zabezpieczenia przed utratą danych i spójnością danych w przypadku zmiany wydajności łącza czy zwiększenia obciążenia urządzenia

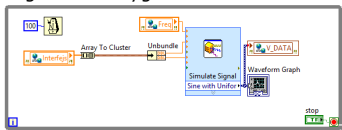




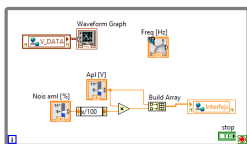
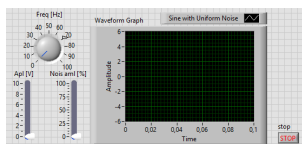
Opracowane w ramach projektu: "Wirtualne Laboratoria Fizyczne nowoczesną metodą nauczania" realizowanego przez Warszawską Wyższą Szkołę Informatyki.

## NI-PSP – przykład aplikacji klient-serwer

Aplikacja serwera- generator sygnału sterowana z klienta



Aplikacja klienta – sterowanie parametrami generatora i odbiór wygenerowanych sygnałów



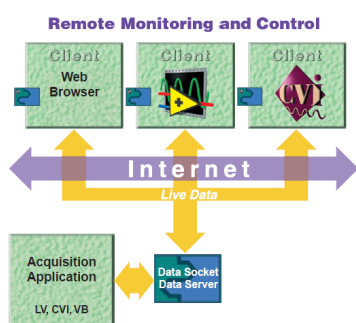
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## Datsocket server

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## Zdalny dostęp do aplikacji w LabVIEW



Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## Technologia DataSocket Server

**DataSocket** popularnie nazywany gniazdkami w wykonaniu National Instruments jest technologią opartą o standard TCP/IP umożliwiającą wymianę danych pomiędzy aplikacjami za pośrednictwem sieci Ethernet.

**DataSocket** w środowiskach NI składa się z dwóch elementów:

- **DataSocket API** – będące interfejsem służącym do wymiany danych różnego typu pomiędzy aplikacją a DataSocket Serverem lub bezpośrednio programującym dostęp do interfejsu sieciowego.
- **DataSocket Server** – jest zarządzaną aplikacją (serwerem) za pośrednictwem której następuje komunikacja z wykorzystaniem protokołu TCP/IP

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## DataSocket API

**Dostępne protokoły – adresowane na zasadzie adresów url:**

- **HTTP** – współpraca z serwerami www
- **FTP** – wykorzystywany do przesyłania plików w systemach pomiarowych
- **Local files** – używany do dostępu do plików na zdalnych maszynach
- **OPC – OLE for Process Control** – współpraca z przemysłowymi urządzeniami sieciowymi oraz systemami pomiarowymi
- **DSTP – DataSocket Transfer Protocol** – umożliwia przesyłanie dowolnych danych pomiędzy aplikacjami pomiarowymi

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

## DataSocket API

**API dostępne pod postacią:**

- kontrolka ActiveX,
- biblioteka w języku C dla LabWindows/CVI,
- zestaw programów środowiska LabView – VI,
- biblioteka do języka Java (?)

**Działanie opiera się o właściwe użycie funkcji:**

- Open
- Read
- Write
- Close

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

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

### DataSocket Server (DSS)

#### DSS - Oprogramowanie pośredniczące w transmisji

Logo: KAPITAŁ LUDZKI, WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI, UNIA EUROPEJSKA FUNDUSZ SPOŁECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Nadawanie danych

**Wymagana jest współpraca 3 elementów:**

- nadawcy (*publisher*, lub *writer*),
- serwera DataSocket,
- abonenta (*subscriber*, lub *reader*).

**Lokalizację nadawcy oraz uprawnienia klientów określa się za pomocą narzędzia:**

**DataSocket Server Manager**

Logo: KAPITAŁ LUDZKI, WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI, UNIA EUROPEJSKA FUNDUSZ SPOŁECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### DataSocket Server Manager

#### Umożliwia konfigurację domyślnych właściwości serwera DSS.

Logo: KAPITAŁ LUDZKI, WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI, UNIA EUROPEJSKA FUNDUSZ SPOŁECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### DataSocket Server – nadawanie danych

Logo: KAPITAŁ LUDZKI, WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI, UNIA EUROPEJSKA FUNDUSZ SPOŁECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Zasada budowania aplikacji z DSS

#### Aplikacja wysyłająca dane

#### Aplikacja odbierająca dane

Logo: KAPITAŁ LUDZKI, WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI, UNIA EUROPEJSKA FUNDUSZ SPOŁECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Przykład programu

#### Uruchamianie DSServera - SubVI

Logo: KAPITAŁ LUDZKI, WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI, UNIA EUROPEJSKA FUNDUSZ SPOŁECZNY

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”



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

**VI Server**

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

### VI Server - właściwości

- Niezależny od platformy softwarowej
- Uproszczenie do nieprzystawności komunikacji sieciowej pomiędzy kontrolkami
- Oparty na hierarchii klas
- Zapewnia synchronizację procesów
- Oparty o RPC
- Umożliwia programową kontrolę nad obiektami w VI, oraz dynamiczne ładowanie, edycję i uruchamianie aplikacji LabVIEW.
- Funkcjonalności dostępne przez sieć jak i lokalnie

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

### Najprostsza aplikacja sieciowa

#### Aplikacja nadająca dane

**Interfejs**

**Diagram**

A gdzie jest wysyłanie danych?

**Wysyłanie – opcje wykresu**

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

### Najprostsza aplikacja sieciowa

#### Aplikacja odbierająca dane

**Interfejs**

**Diagram**

A gdzie jest odbieranie danych?

**Odbieranie – opcje wykresu**

Więcej kodu zawiera zatrzymanie działania aplikacji niż samo odbieranie

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

**WEBSERVICES**

Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

### LabVIEW – usługi sieciowe

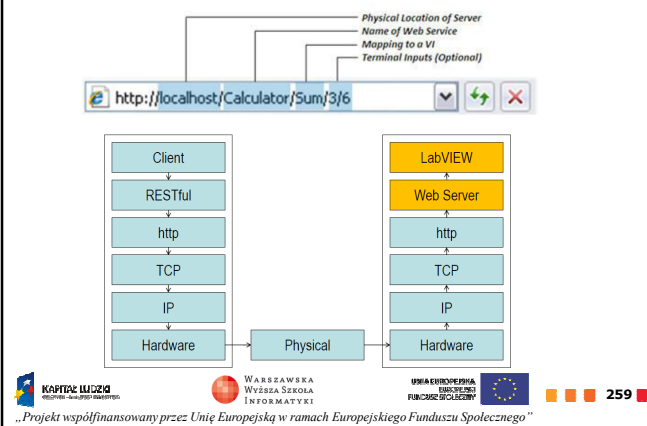
- Umożliwiają integrację z innymi systemami.
- Umożliwiają wywoływanie funkcji oraz vi za pośrednictwem sieci Web.
- Możliwość publikacji jak i subskrypcji usług sieciowych.
- Niezależne od platformy softwarowej.

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego



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

## Usługi sieciowe w LabVIEW – jak działają?



## Usługi sieciowe - przykład

VI Deployed as Web Service

..Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego"

## Usługi sieciowe - właściwości

### Zalety:

- Ogólnie przyjęty standard http – wspierany przez wszystkie urządzenia sieciowe na rynku
- Bezproblemowy w sieciach z zaporą ogniową – łatwość konfiguracji
- RPC z dowolnego obiektu wywoływane za pośrednictwem żądania http,
- Możliwość pracy z wieloma klientami na raz.
- W przypadku LabVIEW wsparcie dla pracy w czasie rzeczywistym

### Wady:

- Nieoptymalizowany pod kątem streamingu danych
- Nieco bardziej obciąża zasoby niż standardowe TCP/UDP

## Podsumowanie

- Wybór sposobu komunikacji powinien nastąpić na podstawie wymagań aplikacji a nie łatwości wykonania mechanizmów komunikacji.
- Każdy mechanizm ma swoje zastosowanie a także wady które powinny zostać uwzględnione na etapie projektowania aplikacji.
- Na NI Developer Zone dostępnych jest wiele zaawansowanych przykładów wraz z opisem przypadków użycia ułatwiających podjęcie decyzji o zastosowaniu danego rozwiązania.

## LabVIEW krok po kroku

- Budowanie biblioteki DLL
- Budowanie aplikacji z wykorzystaniem graficznego języka programowania
- Komunikacja sieciowa przykłady

## Budowanie DLL w LabVIEW

Budowanie biblioteki dll przedstawione zostanie na przykładzie funkcji konwersji temperatury z F na C

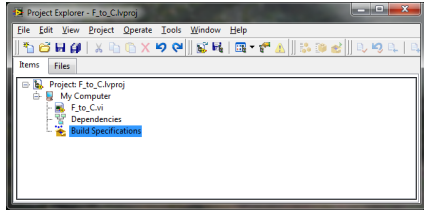
$$\text{Deg C} = (\text{Deg F} - 32.00) / 1.80$$

**Obszar zastosowań:** szybkie zakodowanie obliczeń czy skomplikowanych algorytmów i użycie we własnych aplikacjach napisanych w innych językach programowania.

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

## Utworzenie nowego projektu

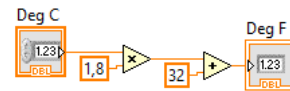
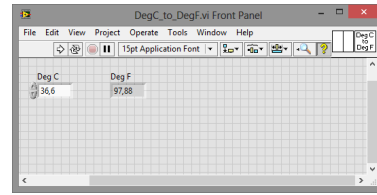
Pierwszym krokiem jest utworzenie nowego projektu oraz dodanie nowego vi-ja



**KAPITAŁ LUDZKI** **WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI** **UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY** 265

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Panel czołowy i diagram

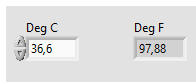


**KAPITAŁ LUDZKI** **WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI** **UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY** 266

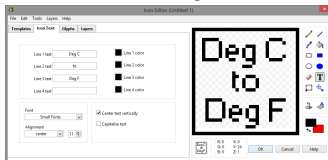
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Budowa SubVI

Zapisanie stworzonego programu (DegC\_to\_DegF.vi)



Utworzenie ikony dla SubVI

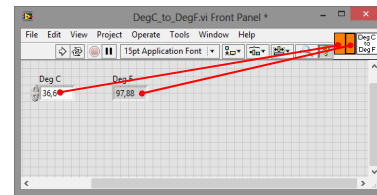


**KAPITAŁ LUDZKI** **WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI** **UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY** 267

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Budowa SubVI

Podłączanie konektorów (narzędzie szpulka)



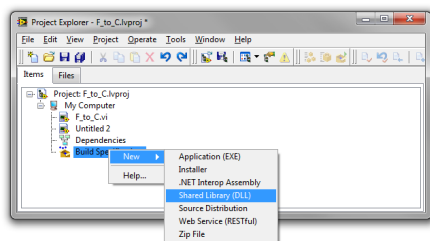
Należy powiązać konektory wejściowe i wyjściowe z odpowiednimi elementami interfejsu

**KAPITAŁ LUDZKI** **WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI** **UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY** 268

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Budowanie DLL

Należy dodać nową specyfikację dystrybucji aplikacji:  
Build Specification -> New -> Shared Library (DLL)

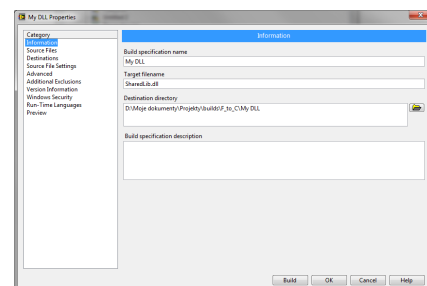


**KAPITAŁ LUDZKI** **WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI** **UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY** 269

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Właściwości DLL

Należy ustawić wszystkie interesujące nas właściwości budowanej biblioteki DLL



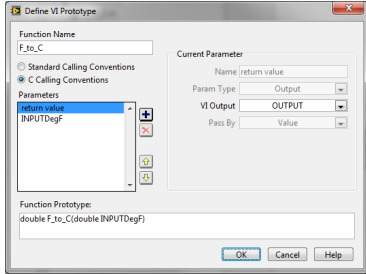
**KAPITAŁ LUDZKI** **WARSZAWSKA WYŻSZA SZKOŁA INFORMATYKI** **UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY** 270

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

## Konfiguracja plików źródłowych dla DLL

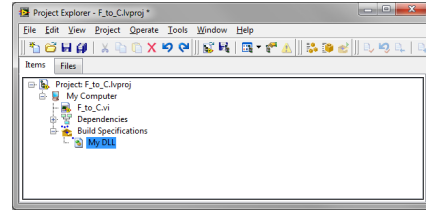
Dla każdej z zbudowanych funkcji jakie mają się znaleźć w wynikowej bibliotece należy zdefiniować (domyślne wartości są nie intuicyjne) nazwę funkcji oraz nazwy parametrów.



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## i biblioteka gotowa ... 😊

Jeżeli chcemy umieścić więcej funkcji w bibliotece DLL należy powtórzyć tą czynność dla każdej dodawanej funkcji z pliku VI.



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## LabVIEW w przykładach

Ćwiczenia do samodzielnego wykonania

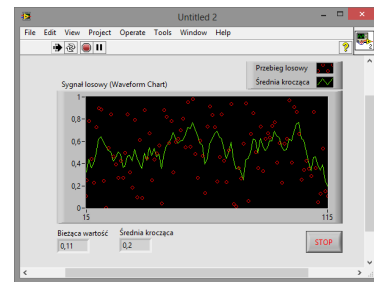
Projekt jest realizowany przez  
Warszawską Wyższą Szkołę Informatyki

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## LabVIEW w przykładach

### Ćwiczenie 1:

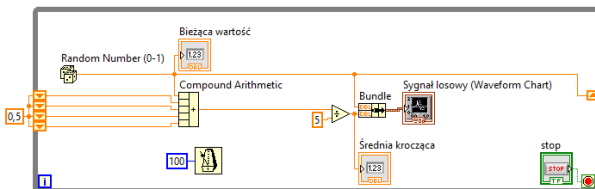
Wyświetlanie wielu przebiegów jednocześnie (płyta czołowa).



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## LabVIEW w przykładach

Wyświetlanie wielu przebiegów jednocześnie (diagram).

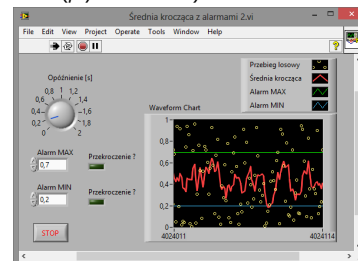


„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## LabVIEW w przykładach

### Ćwiczenie 2:

Zbudować aplikację generującą losowe dane wraz z systemem analizy i alarmów (płyta czołowa).

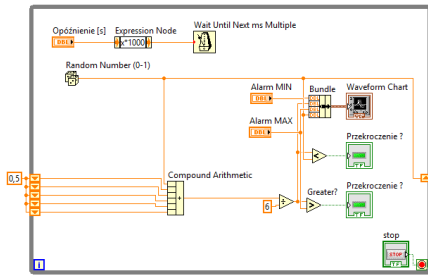


„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### LabVIEW w przykładach

Wyświetlanie wielu przebiegów jednocześnie (diagram).



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### LabVIEW w przykładach

#### Ćwiczenie 3:

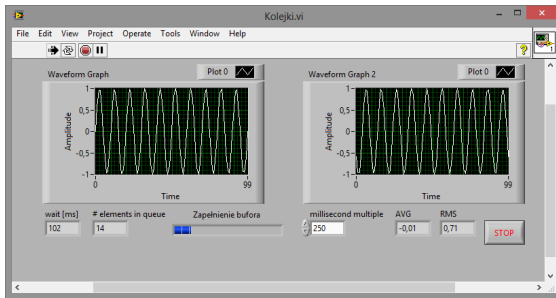
Zbudować aplikację wykorzystującą mechanizm kolejek oraz trzy równoległe pętle:

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

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### LabVIEW w przykładach

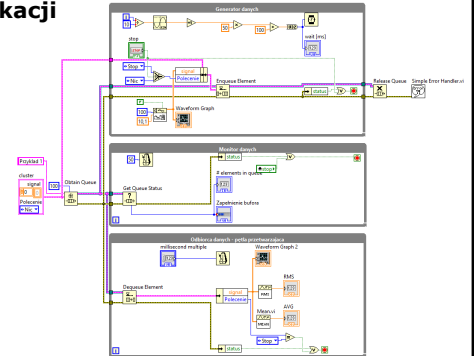
#### Panel czołowy aplikacji



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### LabVIEW w przykładach

#### Diagram aplikacji



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

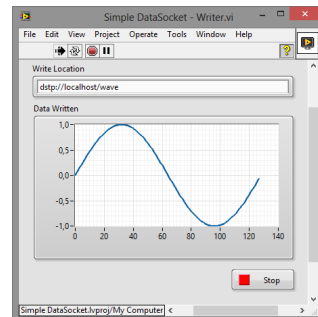
### Aplikacja klient – serwer z DSS

#### Ćwiczenie 4:

- Zbudować aplikację wysyłającą wygenerowany przebieg sinusoidalny z wykorzystaniem technologii DataSocket Server
- Zbudować aplikację odbierającą dane z wykorzystaniem technologii DataSocket Server

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Nadawanie danych - interfejs



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Nadawanie danych - diagram

283

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Odbieranie danych - interfejs

284

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Odbieranie danych - diagram

285

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Aplikacja klient – serwer z DSS

**Ćwiczenie 5:**

- Zbudować aplikację wysyłającą wygenerowany przebieg sinusoidalny z wykorzystaniem technologii DataSocket Server oraz VI Server
- Aplikacja ma mieć możliwość regulacji, częstotliwości, fazy sygnału oraz amplitudy
- Zbudować aplikację odbierającą dane z wykorzystaniem technologii DataSocket Server oraz VI Server

286

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Najprostsza aplikacja sieciowa

#### Aplikacja nadająca dane

**Interfejs**

**Diagram**

287

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Najprostsza aplikacja sieciowa

#### Aplikacja odbierająca dane

**Interfejs**

**Diagram**

**Odbieranie – opcje wykresu**

Więcej kodu zawiera zatrzymanie działania aplikacji niż samo odbieranie

288

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”



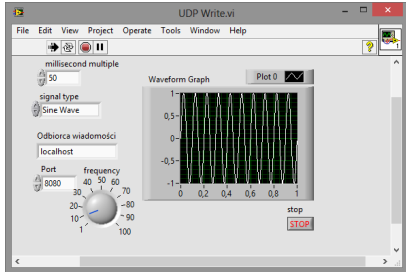


Opracowane w ramach projektu: "Wirtualne Laboratoria Fizyczne nowoczesną metodą nauczania" realizowanego przez Warszawską Wyższą Szkołę Informatyki.

## Komunikacja UDP – aplikacja serwera

### Ćwiczenie 7:

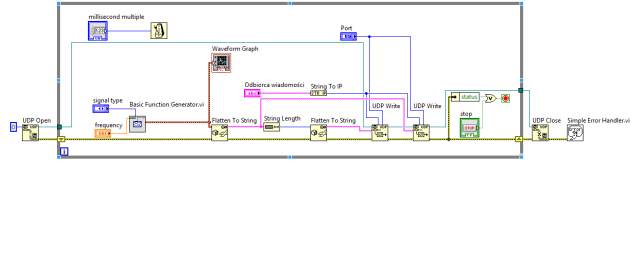
Zbudować aplikację klient-serwer przesyłającą dane opartą o protokół UDP.



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Komunikacja UDP – aplikacja serwera

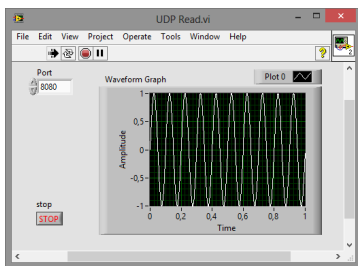
### Diagram



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Komunikacja UDP – aplikacja klienta

### Interfejs aplikacji klienta UDP



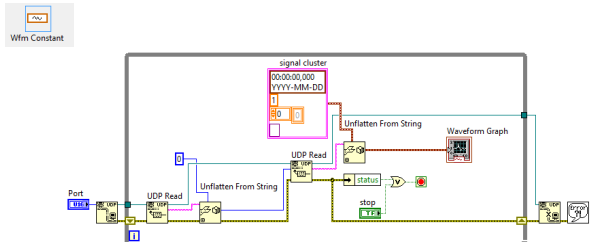
„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Komunikacja UDP – aplikacja klienta

### Diagram

element signal cluster:

Waveform > Analog Waveform > Waveform Constant > Wfm Constant

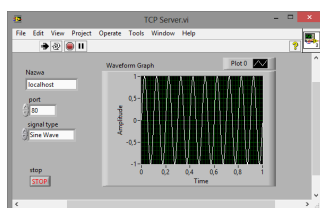


„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Komunikacja TCP

### Ćwiczenie 8:

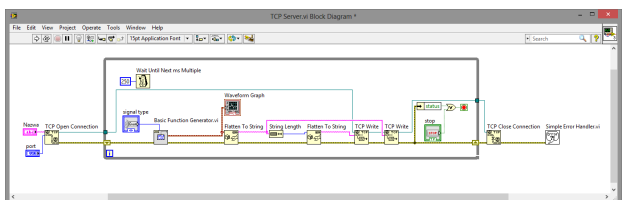
Zbudować aplikację klient-serwer przesyłającą dane opartą o protokół TCP. Na podstawie aplikacji serwera zbudować aplikację klienta.



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Komunikacja TCP

### Diagram



„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

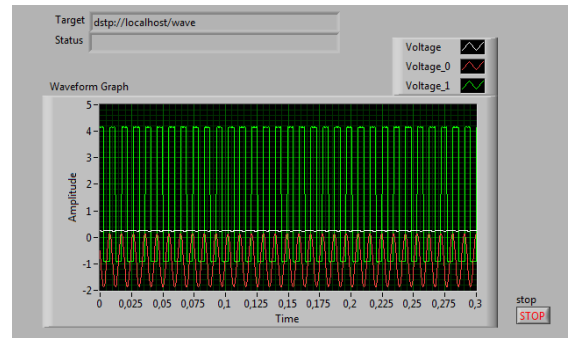
Opracowane w ramach projektu: **"Wirtualne Laboratoria Fizyczne nowoczesną metodą nauczania"** realizowanego przez **Warszawską Wyższą Szkołę Informatyki**.

### Aplikacja klient – serwer z DSS

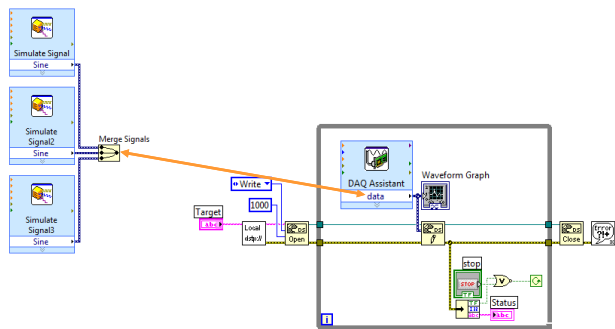
#### Ćwiczenie 9:

- Zbudować aplikację wysyłającą odczytane przebiegi z karty zbierania danych lub z symulatora (użyć funkcji simulate signal i wygenerować 3 przebiegi) za pośrednictwem technologii DataSocket Server
- Zbudować aplikację odbierającą dane z wykorzystaniem technologii DataSocket Server

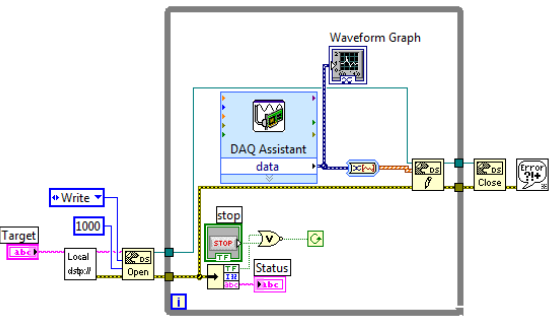
### Przykład – wersja 1



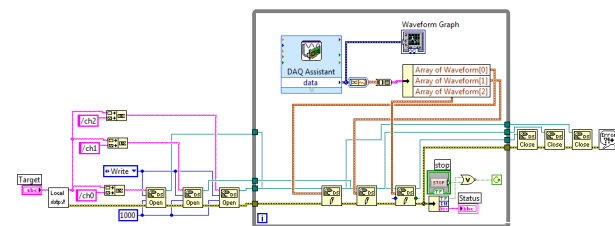
### Przykład – wersja 1 (diagram) ?



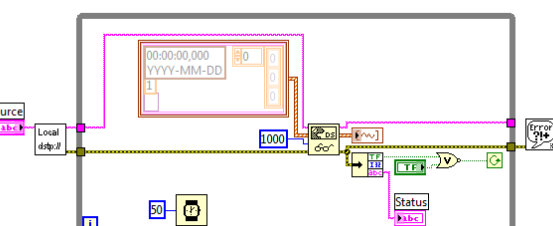
### Przykład – wersja 2 (diagram)



### Przykład – wersja 2 (diagram)



### Odczyt wersja 1



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

### Odczyt wersja 2

301

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Przykładowe zastosowanie

#### Ćwiczenie 10

Wykorzystując jako źródło sygnału aplikacje zrobione w ćwiczeniu 3 należy zbudować wirtualny przyrząd pomiarowy czytający dane z serwera pomiarowego. Przebieg należy poddać podstawowej analizie częstotliwościowej: widmo sygnału, detekcja harmonicznych oraz analizie parametrów czasowych, tzn. wyznaczyć wartości: średnia, skuteczna, maksymalna i minimalna.

302

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Panel czołowy

Frequency (Hz)	Amplitude
50.47	213.63
319.34	425.95
532.40	0.00002392
800.00	0.00000021

303

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Diagram przepływu danych

304

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Diagram

305

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Modulacja AM – ćwiczenie 11

Przykład wykorzystania Wykresów Waveform Grpah oraz modułu MathScript do prezentacji zjawiska modulacji sygnału AM

306

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Modulacja AM

**Diagram**

307

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Generacja sygnałów – ćwiczenie 12

308

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Generacja sygnałów – ćwiczenie 12

309

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Symulacja sygnałów – ćwiczenie 13

310

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Symulacja sygnałów – ćwiczenie 13

311

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Zadanie 14 - Modulacja

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 wyświetlenia). Faza początkowa ma być generowana losowo z przedziału  $0 - 0,5 \text{ Pi}$  lub ustawiana w zakresie  $0 - 1 \text{ Pi}$ . 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)

312

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

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

### Zadanie 14 - Funkcjonalności

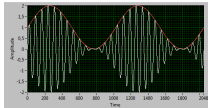
**Parametry**

- Amplituda sin 0-10V
- Amplituda szumu 0-20% (amplitudy przebiegu modulowanego)
- Częstotliwość 0-200 (okresów)
- Składowa stała 5-45
- Częstotliwość fali nośnej 2-20 (okresów)
- Głębokość modulacji 0-200%
- Faza początkowa (kątowa) z przesłasku 0-0.6PI

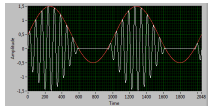
```


    graph TD
      A[Generacja sygnału  
Czynnościowego  
amplitudowo] --> B[Zapis do pliku]
      A --> C[Wyświetlenie przebiegu]
      A --> D[Analiza czasowa  
(Średnia, RMS, min i max)]
      D --> E[Składowa stała  
Wiązania Wyłączona]
      D --> F[Składowe  
Wiązania Włączony]
      D --> G[Sum  
Wiązany/Wyłączony]
      D --> H[Faza początkowa  
Losowo/Przed]
      D --> I[Rodzaje fali nośnej  
prostej i sinus]
      E --> J[Defekcja przekroczenia  
krytyczności  
(maksymalna i minimalna)]
      F --> J
      G --> J
      H --> J
      I --> J
      J --> C
      
```

**Modulacja 100%**



**Modulacja 150%**





313

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

### Zadanie 14 – Panel czołowy



**Amplitude**

4 6  
2 8  
0 10  
1 20

**Częstotliwość (liczba okresów)**

15 20 25 30  
10 5 5 45  
2 50

**Częstotliwość fali nośnej (liczba okresów)**

8 10 12  
6 4 2  
0 20

Sum 5% Składowa stała  
-1 0 1  
-2 1 2  
-3 -3  
-4 -4  
-5 5

Sum: OFF/ON Składowa stała: OFF/ON

**RMS** Max: 1.5 2.99301 Limit MAX: 0  
Średnia: 1.285841 Min: -2.99301 Limit MIN: 0

Faza początkowa: Losowo Typ fali nośnej: prostokat  
0.4 0.5 0.6  
0.2 1 0.7  
0.1 -0.8  
0 -0.9

Głębokość modulacji: 50

Zapisz



314

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”

## Dziękuję za uwagę.

**Pytania?**



elab@mgmaliszewski.pl



315

„Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego”