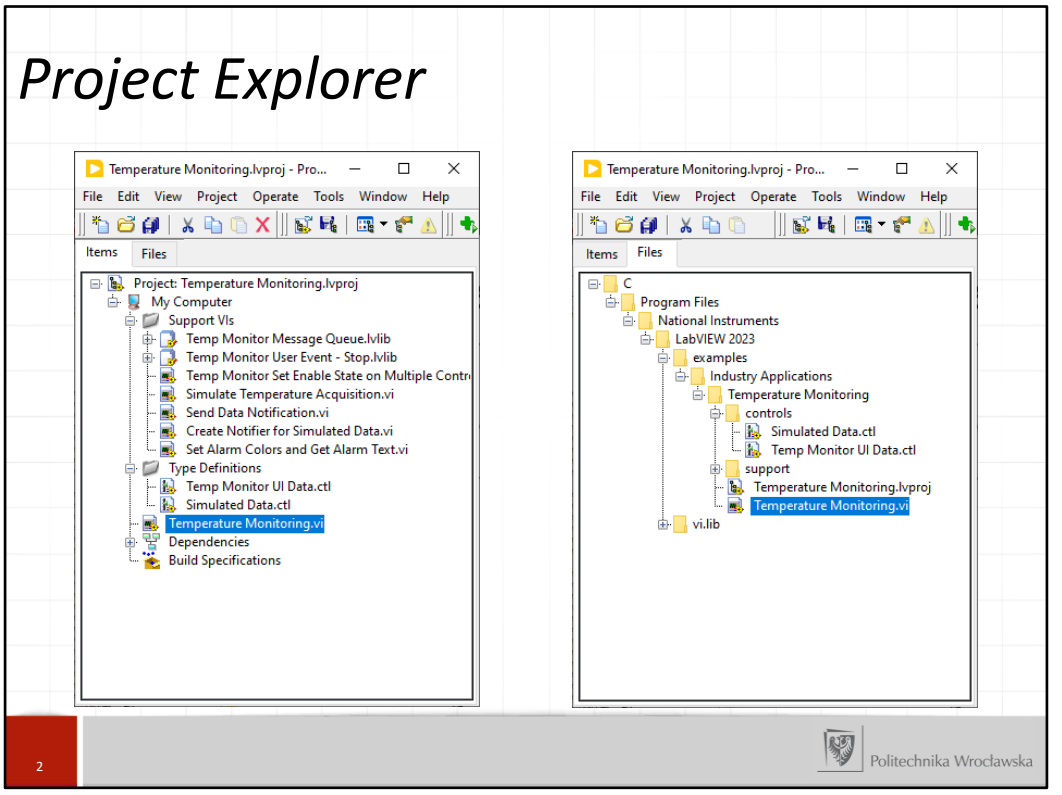
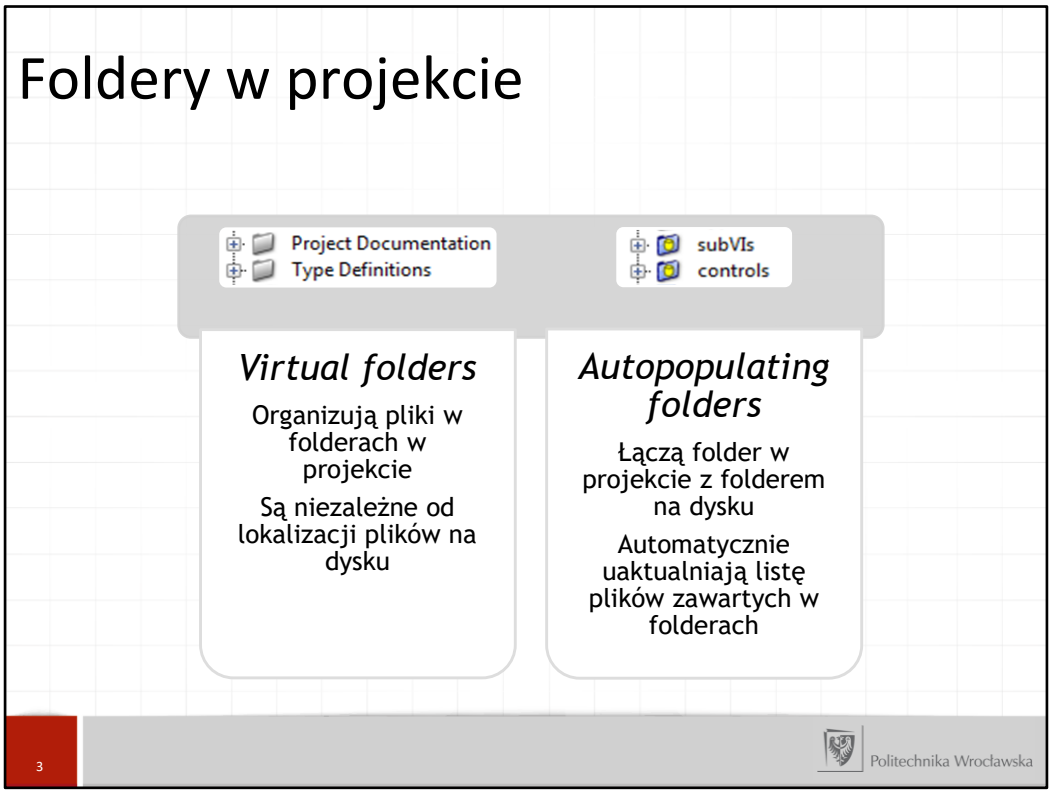
Wykład 4
22.03.2024 r.
Dokumentacja i wizualizacja

Adrian Kaim

Politechnika Wrocławska



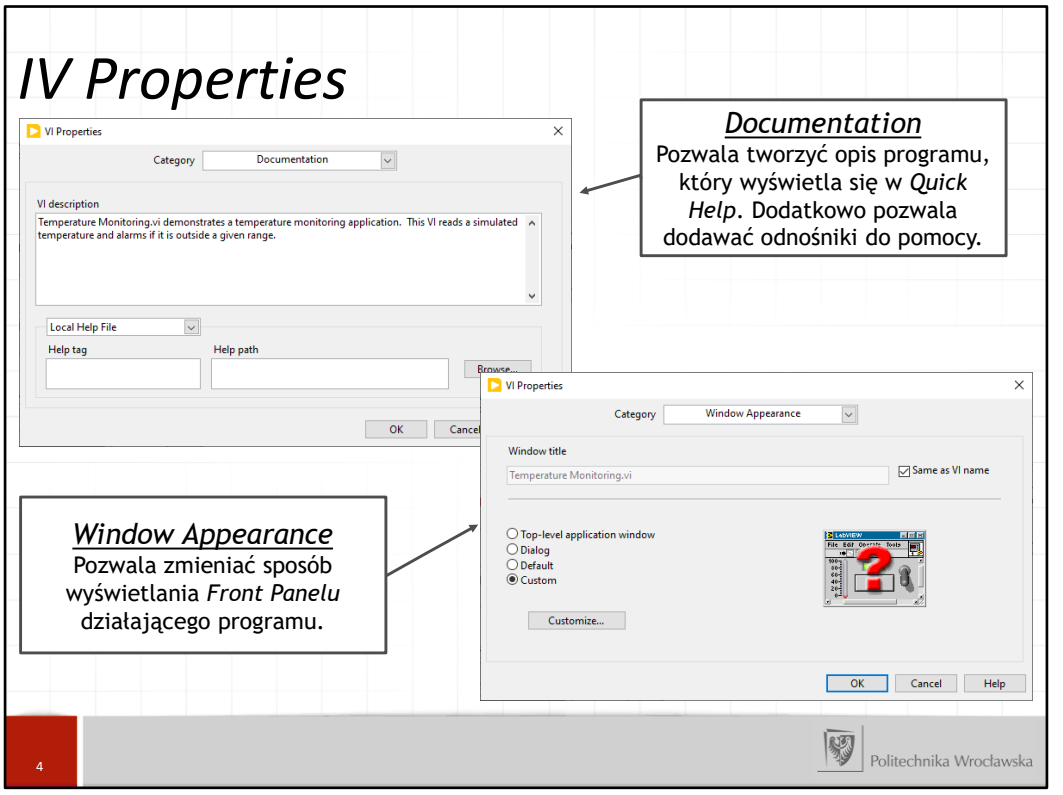
Programy w LabVIEW mogą być organizowane w projekty, zawierające pliki konieczne do ich działania (pliki kodu, biblioteki, niestandardowe kontrolki, pliki tekstowe, pliki graficzne, etc.). W ramach projektu pliki te mogą zostać posegregowane w folderach. Podstawowym narzędziem służącym do obsługi zawartości projektu jest *Project Explorer*, którego okno pojawia się w momencie otwarcia/stworzenia projektu. Zawartość projektu można przeglądać jako listę jego elementów lub jako listę plików na dysku komputera.



Istnieją dwa główne typy folderów, jakie dostępne są w projektach w LabVIEW: wirtualne (*Virtual Folders*) oraz uzupełniające się automatycznie (*Autopopulating Folders*).

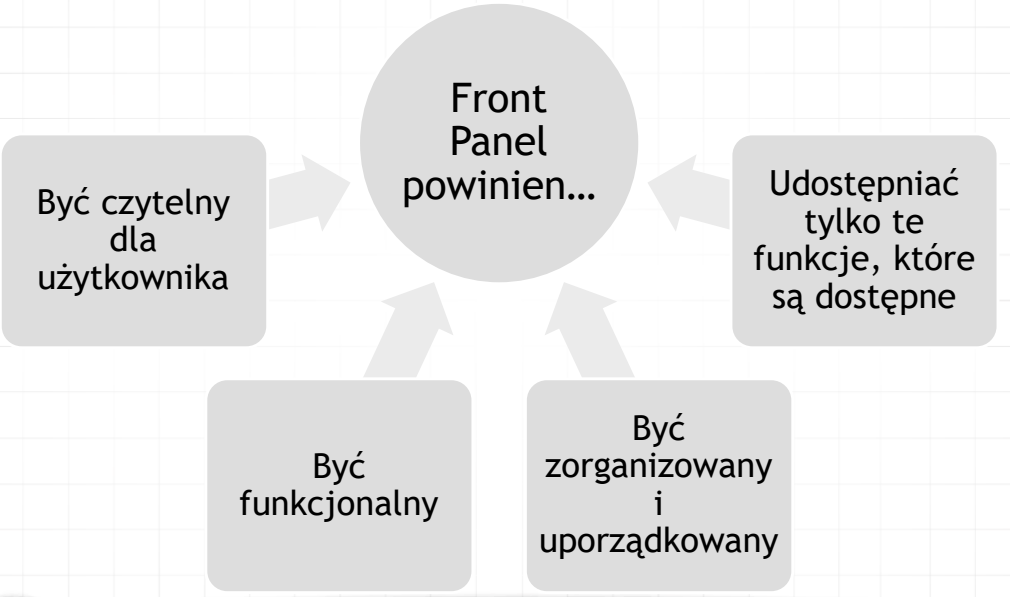
Aby dodać folder wirtualny należy kliknąć prawym przyciskiem myszy na poziom w którym chcemy go stworzyć i wybrać *New>Virtual Folder*. Aby dodać pliki do takiego folderu należy kliknąć na niego prawym przyciskiem myszy i wybrać *Add>File*. Można także stworzyć folder i jednocześnie (jednorazowo) automatycznie dodać jego zawartość - kliknąć prawym przyciskiem myszy na poziom w którym chcemy go stworzyć i wybrać *Add>Folder (Snapshot)…*.

Aby stworzyć folder typu *Autopopulating*, należy kliknąć prawym przyciskiem myszy na poziom w którym chcemy go stworzyć i wybrać *Add>Folder (Autopopulating)…*.

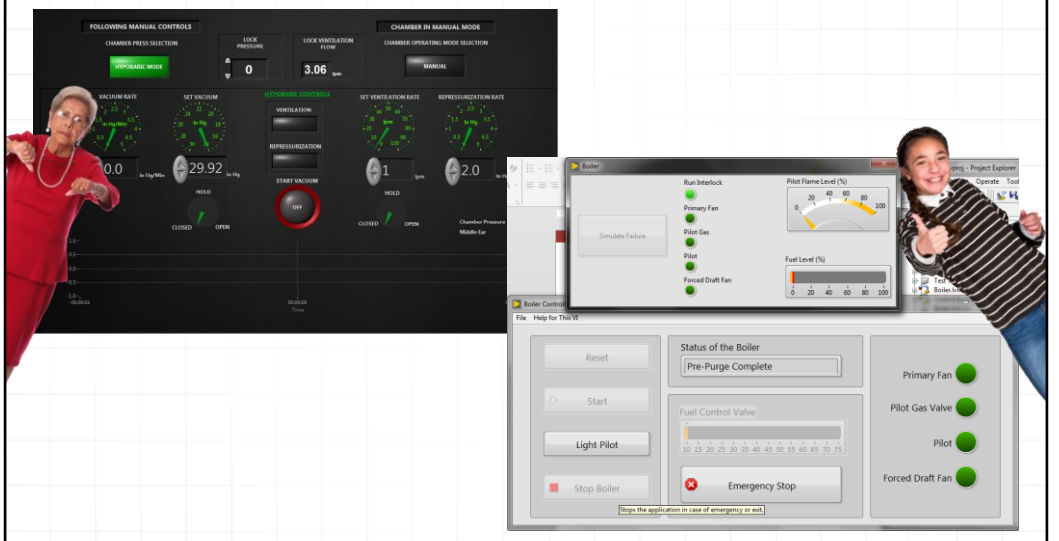


Aby wyświetlić okno własności programu (*VI Properties*), należy wybrać *File>VI Properties*.

Zasady projektowania *Front Panelu*

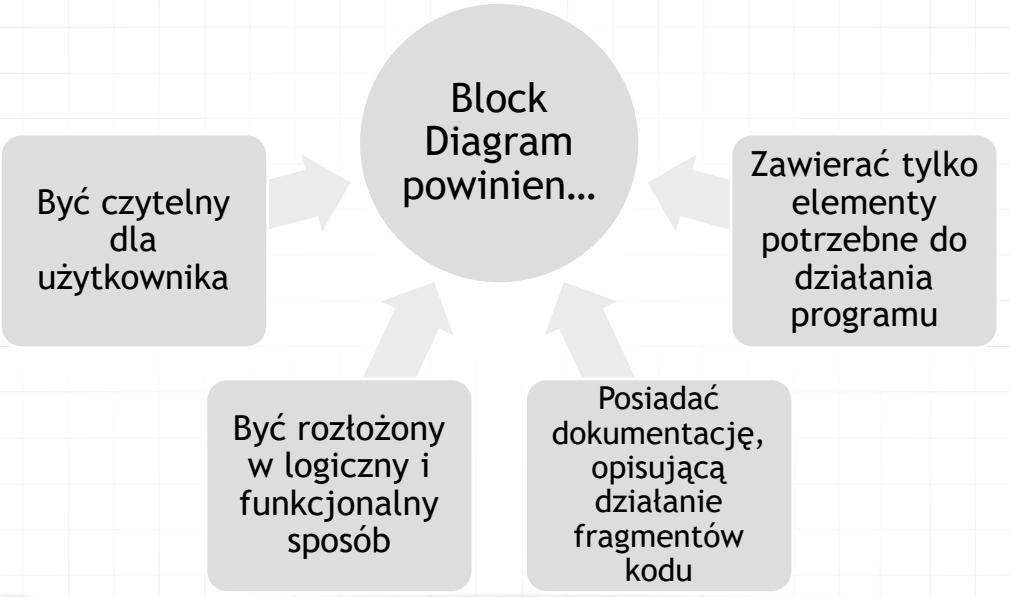


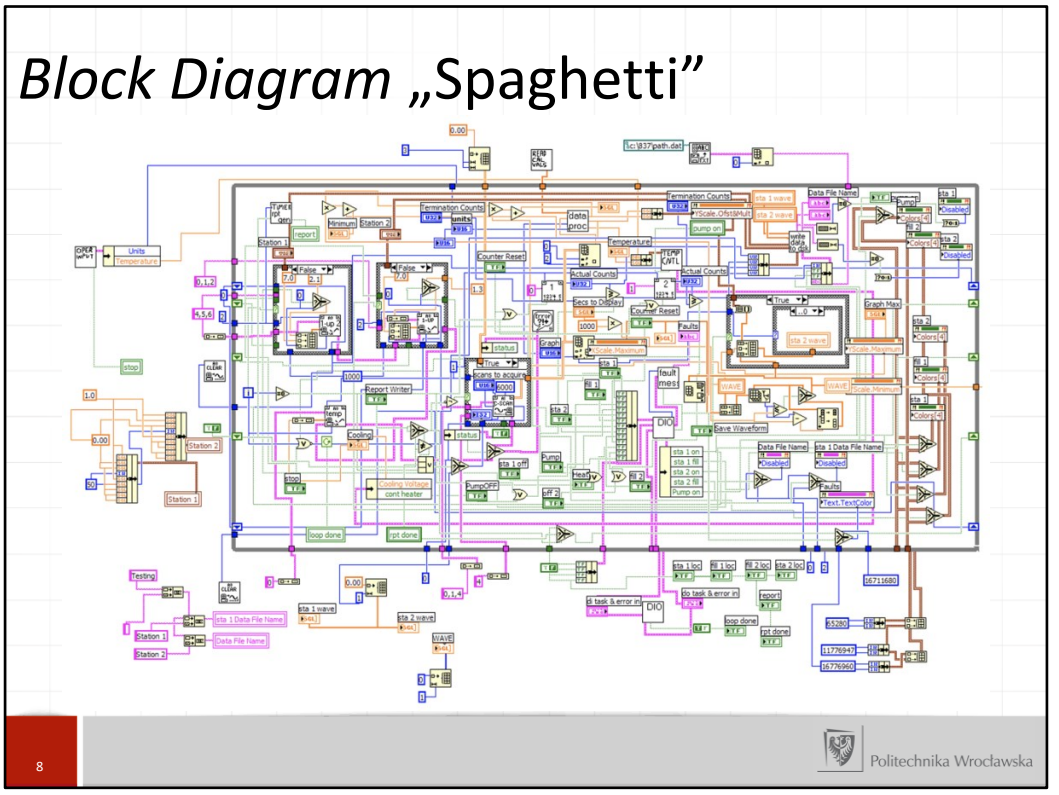
Zasady projektowania *Front Panelu*



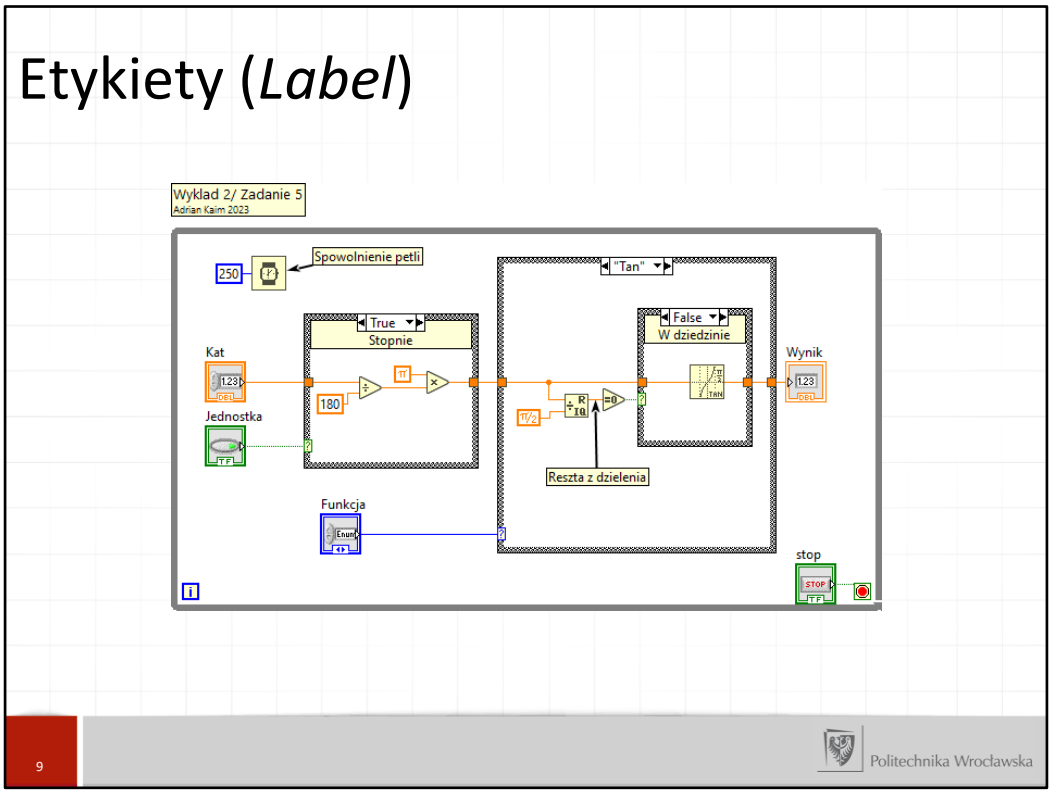
Przykłady dobrze i źle zaprojektowanych *Front Paneli*.

Zasady projektowania *Block Diagramu*

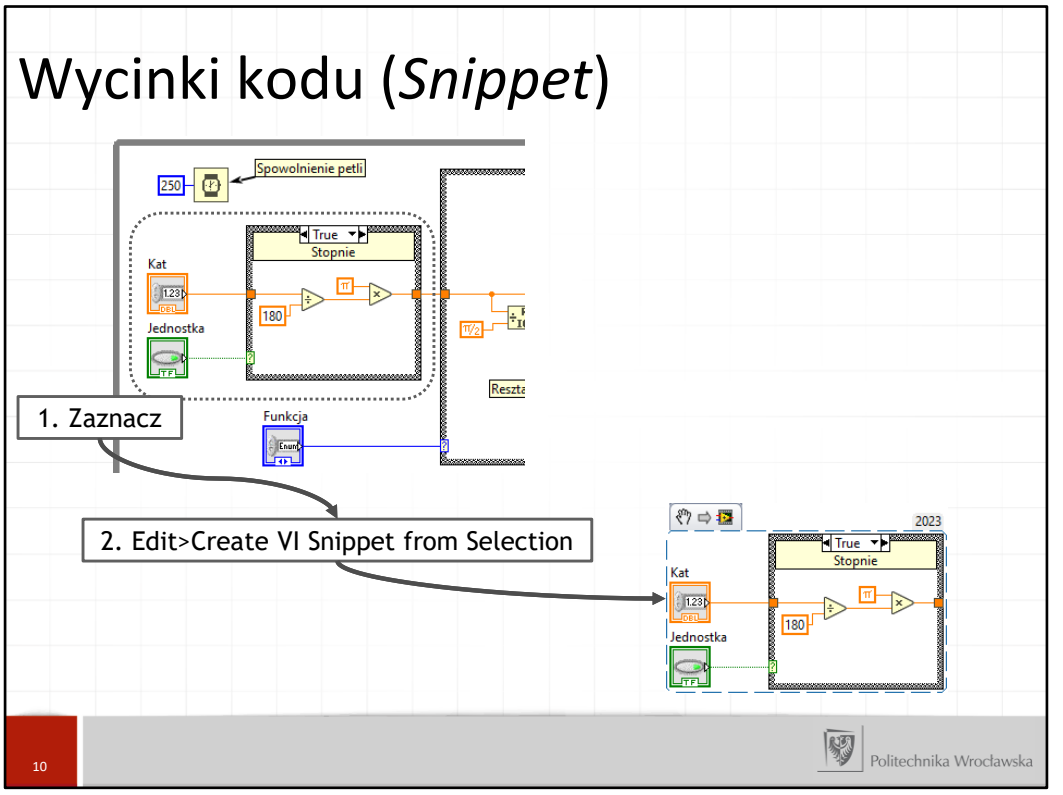




Przykład źle zaprojektowanego *Block Diagramu*.



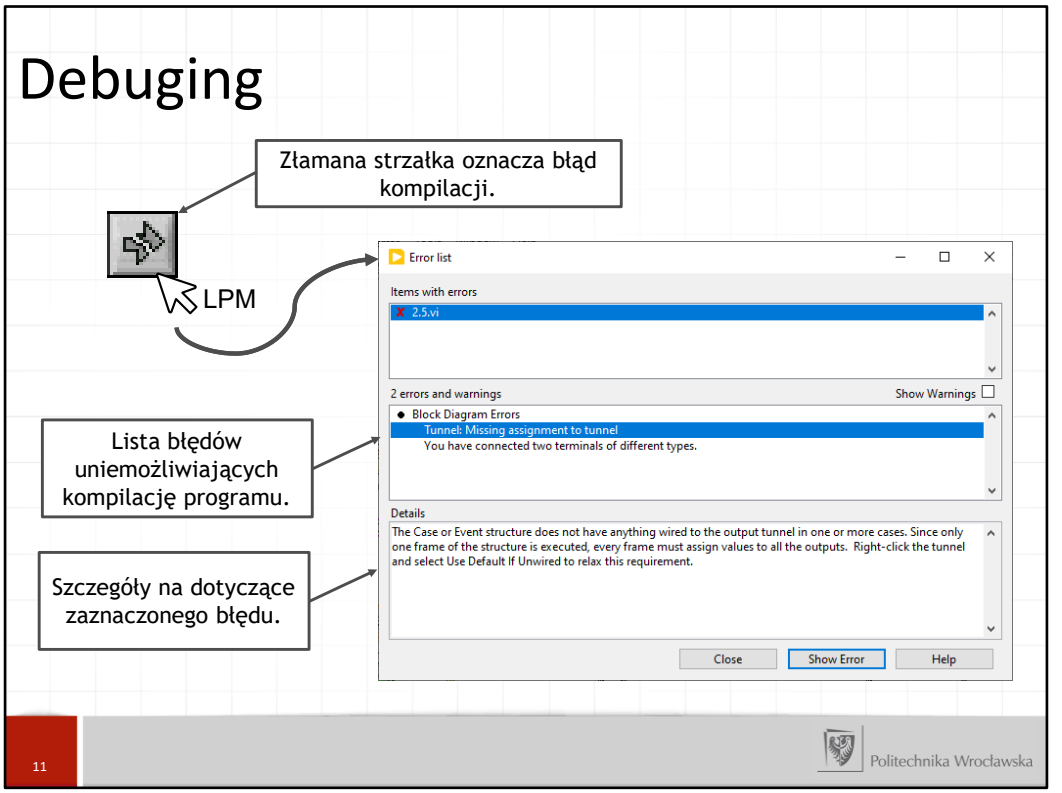
Do opisywania *Block Diagramu* oraz *Front Panelu* można wykorzystać etykiety. Aby dodać swobodną etykietę należy dwukrotnie kliknąć lewym przyciskiem myszy w wolnym miejscu. Swobodne etykiety mogą zostać zaczepione do elementów *Block Diagramu* przez połączenie ich strzałką, pojawiającą się po najechaniu kursorem na etykietę. Możliwe jest także stworzenie etykiet poddiagramów w strukturach – aby to zrobić, należy kliknąć prawym przyciskiem myszy na daną strukturę i wybrać *Visible Items>Subdiagram Label*.



Pomocne w dokumentacji kodu (np. na forach internetowych) bywają tzw. wycinki kodu (*Snippet*). Wycinek taki jest graficzną reprezentacją fragmentu Block Diagramu, zapisaną w formacie PNG. W metadanych tego pliku graficznego zapisywany jest cały kod reprezentowany przez wycinek, co pozwala na proste uruchomienie go w środowisku LabVIEW.

Aby stworzyć wycinek należy zaznaczyć wybrany fragment kodu i wybrać *Edit>Create VI Snippet from Selection*, po czym należy go zapisać. Zapisanego wycinka nie można edytować w innych programach graficznych, gdyż usuwają one zawarty w metadanych kod.

Aby umieścić wycinek na Block Diagramie, wystarczy przeciągnąć na niego plik z wycinkiem (w przypadku gdy wycinek pochodzi z Internetu, należy go najpierw zapisać na komputerze).

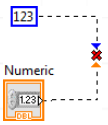


Błędy kompilacji sygnalizowane są złamaną strzałką uruchamiania programu. Po jej kliknięciu uzyskuje się dostęp do listy błędów, które uniemożliwiają kompilację i ich szczegółowych opisów.

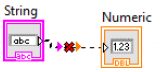
Debuging

Częste przyczyny błędu kompilacji:

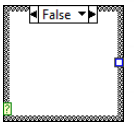
a)



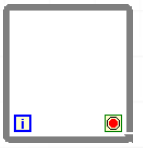
b)



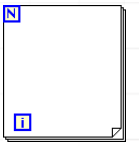
c)

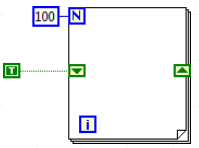


d)

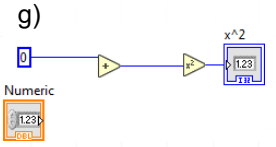


e)





g)



Pytanie 1
Dlaczego program nie kompiluje w każdym z przypadków?

Politechnika Wrocławska

Narzędzia do debugowania

Wykonywanie programu jest spowolnione, a przepływające dane są podświetlane - pozwala to na kontrolę przepływu danych

Wykonywanie Vi krok po kroku, pozwala na kontrolę działania poszczególnych węzłów

Próbkowanie pozwala na podglądanie wartości przepływających w drutach

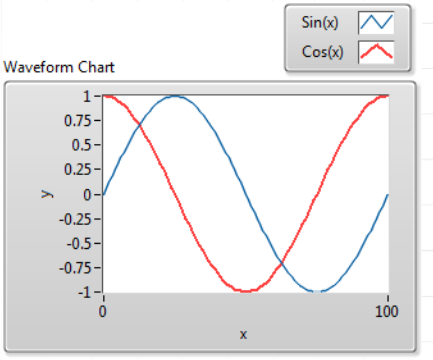
Wykres typu *Waveform Chart*

Indykator numeryczny

Może wyświetlać wiele wykresów

Na osi x kolejne próbki

Wykorzystywany do prezentacji danych „w czasie rzeczywistym”



Politechnika Wrocławska

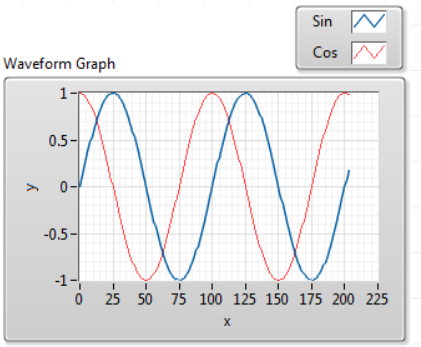
Wykres typu *Waveform Graph*

Indykator numeryczny

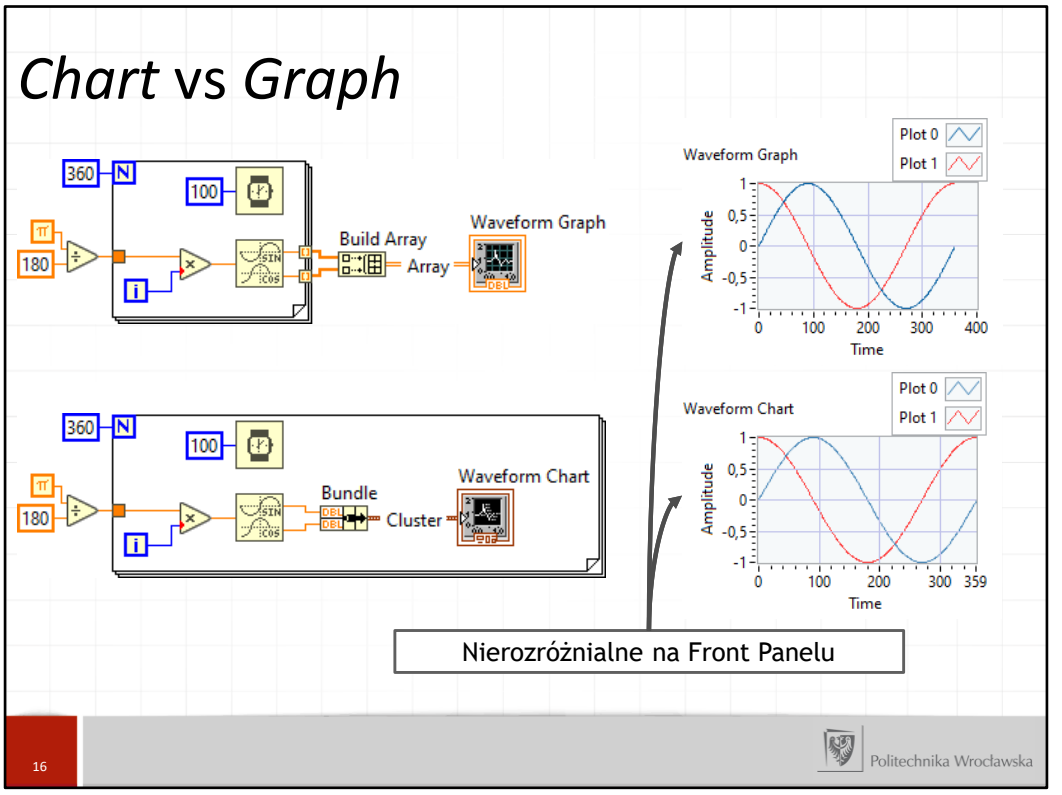
Może wyświetlać wiele wykresów o stałym próbkowaniu

Na osi x kolejne próbki

Wykorzystywany do prezentacji wcześniej wykreowanych danych



Sprawdź też: Przykład 1

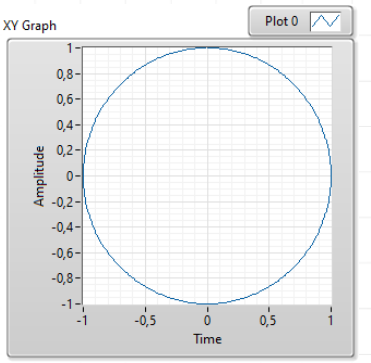


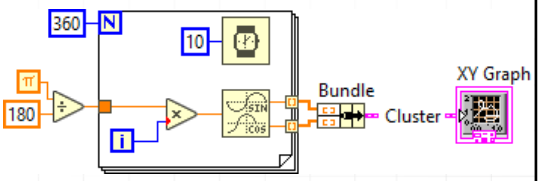
Waveform Graph oczekuje gotowego zestawu danych w postaci tablicy 1D. Aby wyświetlić wiele sygnałów na jednym wykresie, należy złączyć je w tablicę 2D.

Waveform Chart oczekuje w każdym okresie pętli jednego punktu (jednej wartości). Aby wyświetlić kilka sygnałów na jednym wykresie, należy zebrać je w klaster.

Wykres typu *Graph XY*

- Graficzne przedstawienie danych
- Wyświetla zależność x(y)
- Jako danych wejściowych używa dwóch tablic połączonych w klaster
- Można przyłączyć wiele klastrów
- Wykorzystywany do prezentacji wcześniej wykreowanych danych

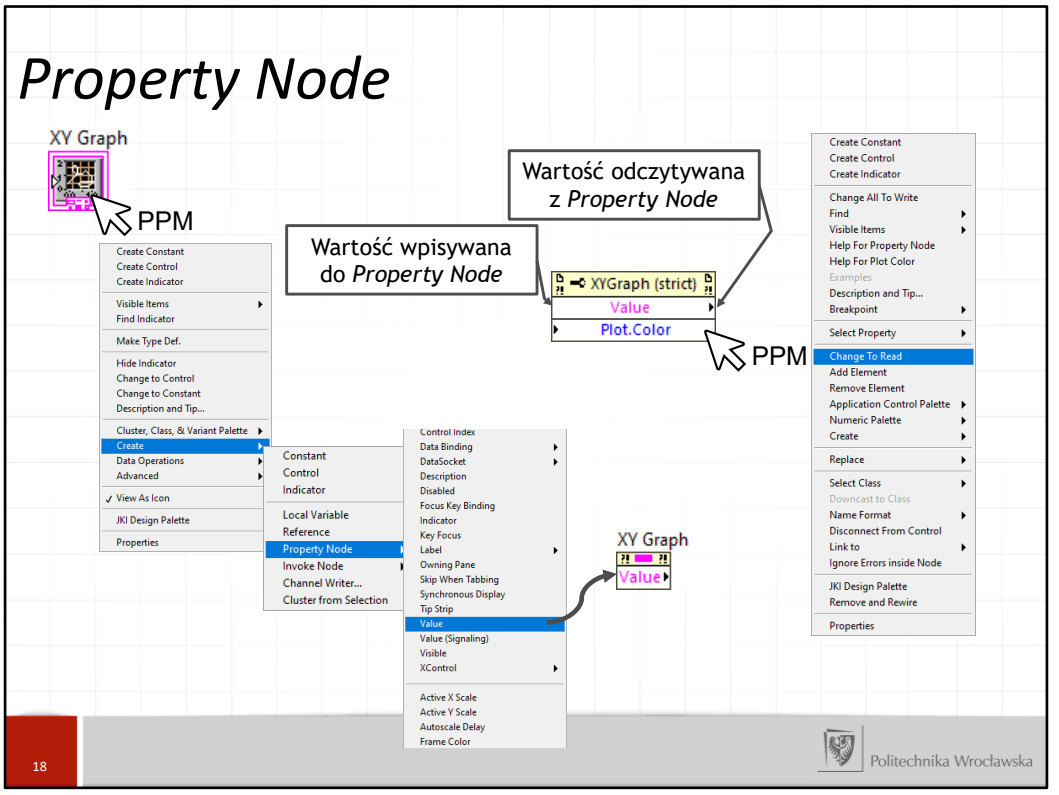




17

Politechnika Wrocławska

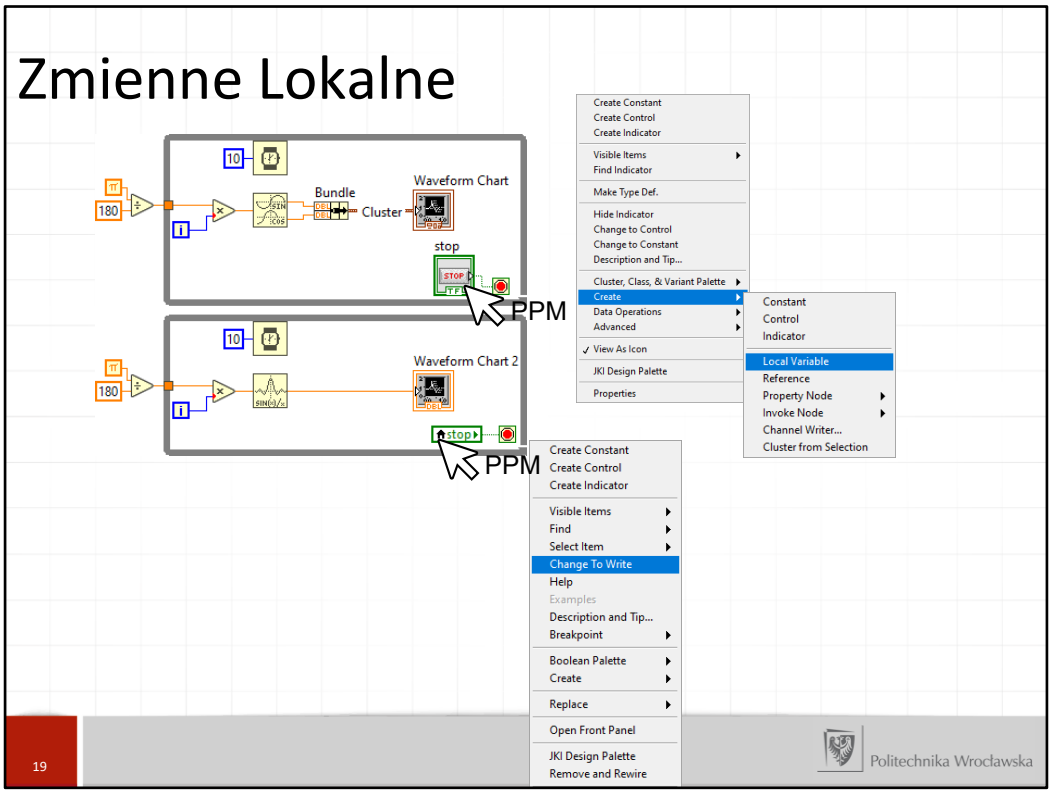
Wykres typu XY Graph oczekuje klastra złożonego z tablicy 1D współrzędnych X oraz tablicy 1D współrzędnych Y. Aby wyświetlić wiele sygnałów na takim wykresie, należy zebrać klastry w tablicę.



Property Node pozwala na programowe zmienianie i odczytywanie własności indykatorów i kontrolek (np.: położenie, kolor, zakresy wprowadzanych danych, wartość elementu). Uwaga: nie wszystkie własności mogą być zmieniane po skompilowaniu programu, nie można ich także zmieniać programowo (przykład: nazwy elementów w kontrolce wyliczeniowej *Enum*).

Aby stworzyć *Property Node* należy kliknąć prawym przyciskiem myszy na kontrolkę/indykator i wybrać *Create>Property Node*, a następnie wskazać własność, której ma dotyczyć. Jeden *Property Node* może dawać dostęp do więcej niż jednej własności (graficzne rozciągnięcie lub PPM>*Add Element*). Jeśli korzysta się z dostępu do wielu własności w ramach jednego *Property Node*, wpisywanie/ odczytywanie odbywa się kolejno z góry na dół.

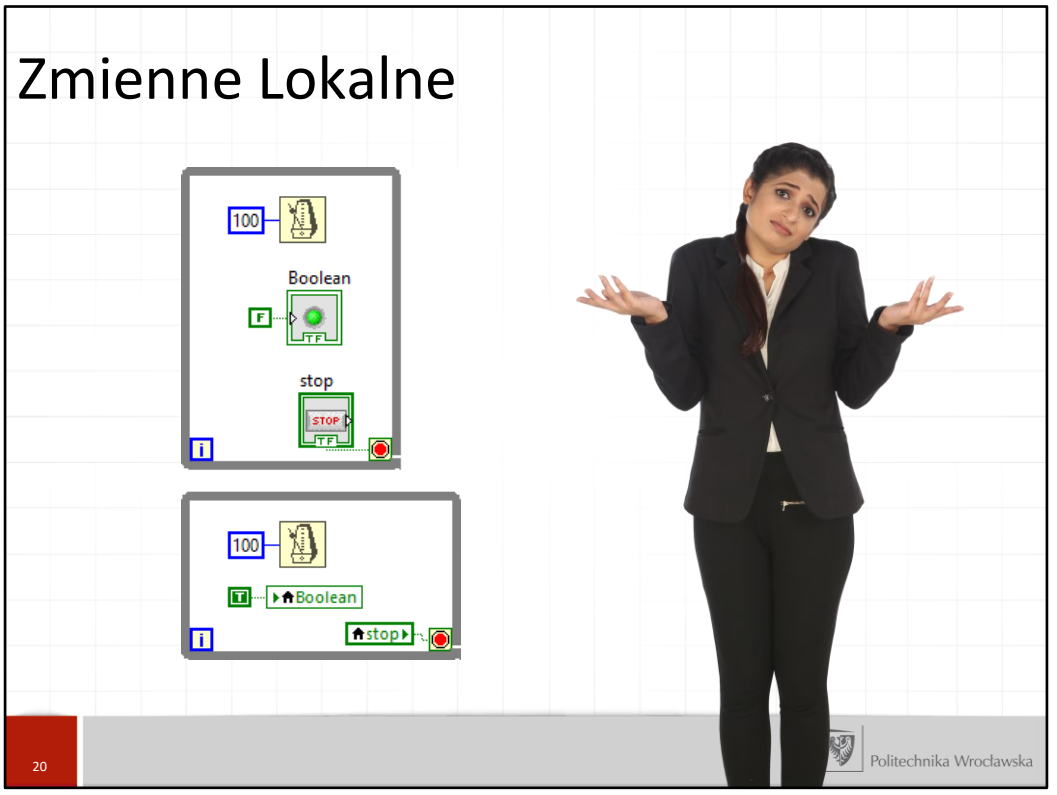
Aby przełączyć własność pomiędzy trybem odczytywania i wpisywania należy kliknąć na nią prawym przyciskiem myszy i wybrać *Change To Read/Write*.



Zmienne lokalne pozwalają uzyskać dostęp do kontrolki lub indykatora w wielu miejscach na Block Diagramie.

Aby stworzyć zmienną lokalną należy kliknąć prawym przyciskiem myszy na kontrolkę/indykator i wybrać *Create>Local Variable*.

Możliwe jest przełączanie trybu działania zmiennej z wpisywania na odczytywanie wartości, w tym celu należy kliknąć na nią prawym przyciskiem myszy i wybrać *Change To Read/Write*.



Należy uważać, aby nie nadpisywać zmiennych lokalnych w dwóch miejscach programu, a w szczególności by nie robić tego jednocześnie. W przypadku takim nie ma pewności które nadpisanie wykona się pierwsze, przez co program może działać w nieoczekiwany i niestabilny sposób.

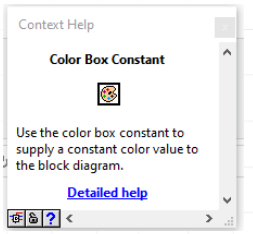
Dioda w pokazanym powyżej przykładzie migać będzie w losowy sposób.

Zadanie 1

Pobierz program ze strony www.

Zlokalizuje błędy w programie. Napraw je. Oznacz naprawione błędy etykietami na Block Diagramie albo Front Panelu.

Wskazówka: typ danych *Color Box Constant*



Zadanie 2

Zaprojektuj etykietę nagłówkową, zawierającą miejsce na:

- nazwę programu lub funkcji,
- dane autora,
- numer wersji kodu oraz datę stworzenia,
- opis funkcjonalności.

Zapisz etykietę jako wycinek (*snippet*).

Zadanie 3

Napisz program do rysowania prostej zadanej przez współczynnik kierunkowy oraz wyraz wolny.
Podpowiedź: do utworzenia tablic wykorzystaj pętle *For* oraz *Auto-indexing.*

Zadanie 4

Napisz program do symulacji sygnału sinusoidalnego, który będzie wyświetlany na wykresie typu *Waveform Chart*. Wykorzystaj *Property Nodes* kontrolki *Waveform Chart*, aby umożliwić użytkownikowi zmianę koloru tła, wykresu, siatki. (Podpowiedź: wykorzystaj kontrolkę Color box).

Zadanie 5

Napisz program symulujący pracę sygnalizatora drogowego.

Zadanie 6

Na wykresie typu *Graph XY* narysuj okrąg o zadanym przez użytkownika promieniu.

Zadanie 7

Napisz program do tworzenia krzywych Lissajous.