

LabVIEW

PLATFORMA EDUKACYJNA

Lekcja 0

Wprowadzenie

Przygotowali: Paulina Grela, Sylwia Jabłońska,
Kamil Rychlewicz, Arkadiusz Szczech

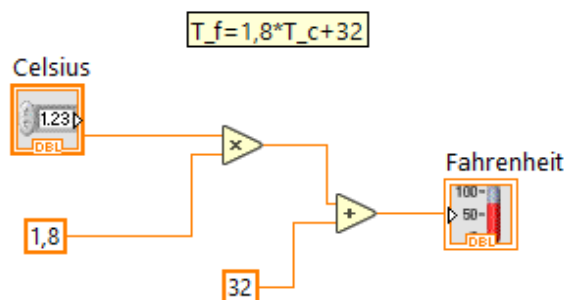
1. Wprowadzenie do kursu

Kurs przeznaczony jest dla osób, które nie miały wcześniej styczności ze środowiskiem LabVIEW. Autorzy mają nadzieję, że umożliwi on szybkie i przyjemne zapoznanie się z jego podstawowymi możliwościami. Jednocześnie zaznaczają, że nie jest to kompleksowe opracowanie i może zawierać błędy.

Programem wykorzystanym podczas szkolenia jest NI LabVIEW 2016 w wersji 32 bitowej. Istnieje możliwość pobrania 45-dniowej wersji trial ze strony internetowej producenta.

Do nauki programowania w tym środowisku wystarczy komputer z zainstalowanym programem. Rozwiązanie to nie wykorzystuje jednak pełnego potencjału oprogramowania. Z tego powodu do realizacji szkolenia wykorzystano zewnętrzne platformy sprzętowe. Pierwszą z nich jest popularna i niedroga płytko Arduino. W kursie został przedstawiony krok po kroku sposób konfiguracji oprogramowania oraz urządzenia, tak aby można było programować je w środowisku LabVIEW. W zamiarze autorów zabieg ten pozwoli wprowadzić czytelnika do obsługi fizycznego sprzętu przy niskim nakładzie finansowym. Zasadniczą część kursu stanowić będzie wykorzystanie platformy myRIO dostarczanej przez firmę National Instruments. Jest to platforma stosowana do prototypowania systemów sterujących przez studentów i inżynierów. Jest relatywnie droższa od Arduino, jednak jej możliwości są o wiele większe.

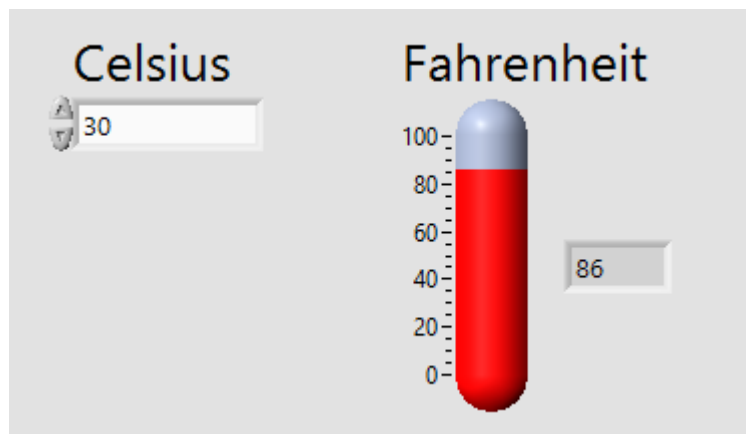
Środowisko LabVIEW wykorzystuje język G. Jest to graficzny język programowania opracowany i rozwijany przez firmę National Instruments od 1986 roku. Programy tworzone są za pomocą symboli graficznych łączonych ze sobą przewodami. Uzyskane połączenia definiują sposób wykonywania programu - określają przepływ danych w programie. W celu zobrazowania zagadnienia posłużono się prostym programem służącym do konwersji temperatury ze skali Celsjusza do skali Fahrenheita.



Wartość temperatury w skali Celsjusza mnożona jest przez liczbę 1.8. Następnie do uzyskanego wyniku dodana jest wartość 32. Dzięki tym zabiegom otrzymano temperaturę w skali Fahrenheita. Warto zaznaczyć, że dodawanie nie wykona się dopóki bloczek dodawania nie otrzyma wartości z bloczku mnożącego. Zauważenie tego jest podstawą do zrozumienia zasad wykonywania kodu w programie LabVIEW.

Zawartość programu umieszczona jest w menu Block Diagram. Po kliknięciu prawym klawiszem myszy na obszar okna, rozwija się menu Functions Palette. Umożliwia ono dodanie do programu bloczków graficznych, które tworzą program. Symbole te zostały opisane w dalszej części kursu.

W rozbudowanych programach obsługa wprowadzania zmiennych bezpośrednio w kodzie programu byłaby nieczytelna. Narzędziem służącym do tworzenia przyjaznego interfejsu komunikacyjnego między programem, a użytkownikiem jest okno Front Panel. Poniżej przedstawiony został interfejs wcześniej wykonanego programu.



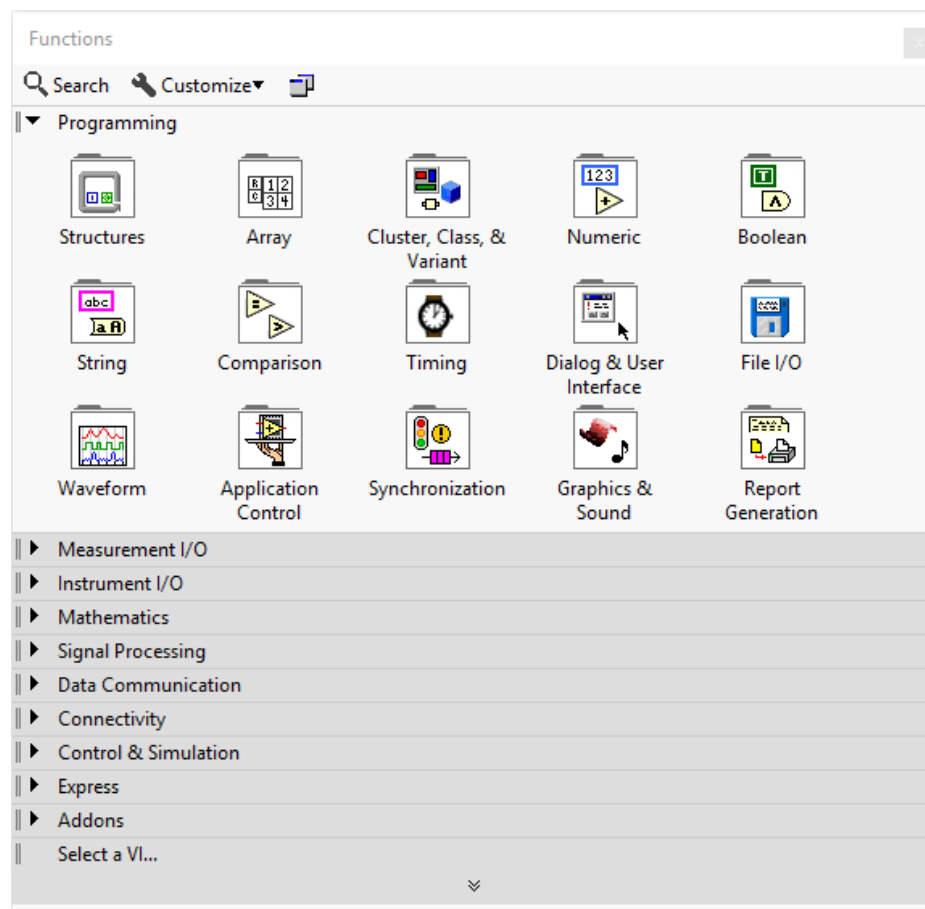
Funkcjonalność okna Front Panel została opisana w kolejnym rozdziale.

2. Funkcjonalność podstawowych bloków

2.1. Block Diagram

2.1.1. Functions Palette

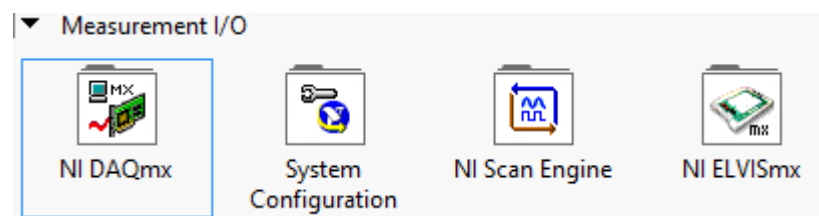
Podstawowym narzędziem dodawania elementów do programowego diagramu blokowego jest Paleta Funkcyjna. Składa się ona z funkcji, które pogrupowane zostały w dziesięć kategorii. Widok okna Palety Funkcyjnej wraz z rozwiniętym widokiem pierwszej kategorii Programming przedstawiono na rys. 1.



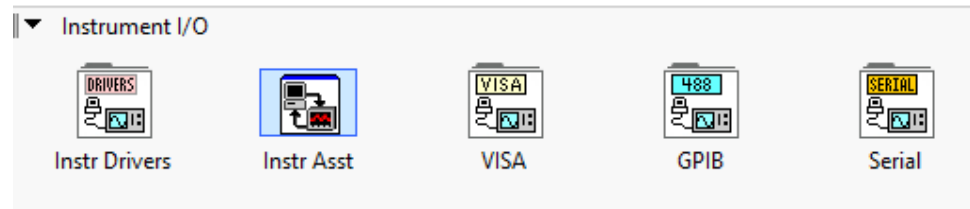
Rysunek 1 - Paleta Funkcyjna – Kategoria Programming

- Funkcja poszczególnych grup narzędzi w kategorii Programming:
 - a) Struktury – kategoria zawierająca m.in. realizację pętli programowych takich jak pętla for i while, a także tworzenie zmiennych różnych typów - globalnych, lokalnych, czy współdzielonych.
 - b) Array - zawiera różnego typu tablice.
 - c) Cluster, Class, & Variant – struktury, dzielenie ich oraz scalanie

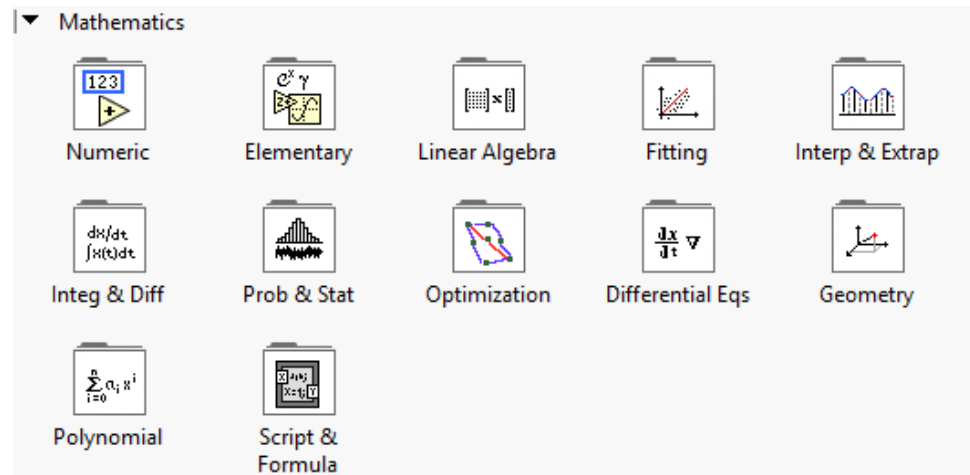
- d) Numeric – operacje matematyczne, m.in. sumowanie, potęgowanie, definicja nieskończoności
 - e) Boolean – operacje boolowskie, w których znajdziemy funkcje logiczne takie jak AND, OR, NOT, XOR i inne.
 - f) String - dostępne w programie operacje na ciągach znaków. Możemy tutaj m.in. definiować ciągi znaków, sprawdzać ich długość, zmieniać wielkość liter i wiele innych.
 - g) Comparison – operacje porównania służące do sprawdzania zakresu liczby, równości liczb, czy szukaniu wartości maksymalnych i minimalnych.
 - h) Timing – operacje czasowe, opóźnienia programowe, zegary systemowe
 - i) Dialog & User Interface - wyświetlanie okien dialogowych z wiadomościami, błędów.
 - j) File I/O- operacje na pliku, m.in. tworzenie, zapis/odczyt do/z pliku.
 - k) Waveform - wyświetlanie przebiegu sygnałów.
 - l) Application Control
 - m) Synchronization – narzędzia służące do synchronizacji programu, definiowanie semaforów i kolejek.
 - n) Graphics & Sound – tworzenie i odtwarzanie dźwięków o określonych parametrach, realizowanie wykresów i płaszczyzn 3D.
 - o) Report Generation - narzędzia służące do generacji raportów.
- kategoria Measurement I/O - zawiera ona wirtualne oprzyrządowanie wykorzystujące instrumenty programowalne takie jak: myDAQ oraz ELVIS II. Dzięki temu możemy wykorzystywać różnorodne narzędzia pomiarowe takie jak: oscyloskop, multimetr, analizator Bode i wiele innych.



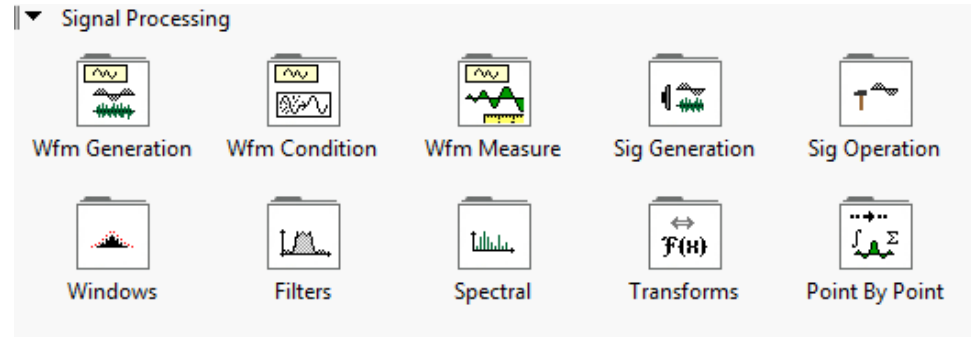
- kategoria Instrument I/O - kategoria ta zawiera komponenty, które w ogólnym przypadku umożliwiają komunikację środowiska LabVIEW z szeroko pojętymi przyrządami pomiarowymi oraz testującymi. Znajdują się tam m.in. funkcję obsługi interfejsu GPIB (General Purpose Interface Bus), który definiuje interfejs od strony fizycznej, elektrycznej oraz funkcjonalnej. VISA (Virtual Instrument Software Architecture) obejmuje większość znanych interfejsów komunikacyjnych.



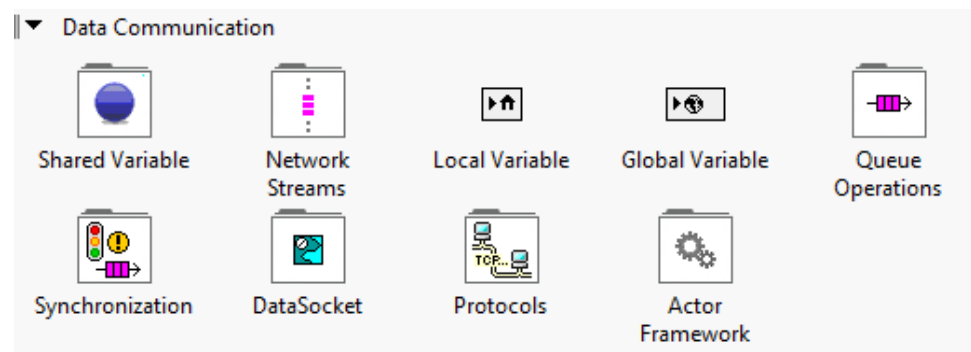
- kategoria Mathematics – zawiera wiele funkcji matematycznych, m.in. podstawowe operacje matematyczne, komponenty związane z algebrą, geometrią, całowaniem, różniczkami. Umożliwia tworzenie wielomianów, mechanizmu różnicowego, histogramów, optymalizację zbioru lub funkcji.



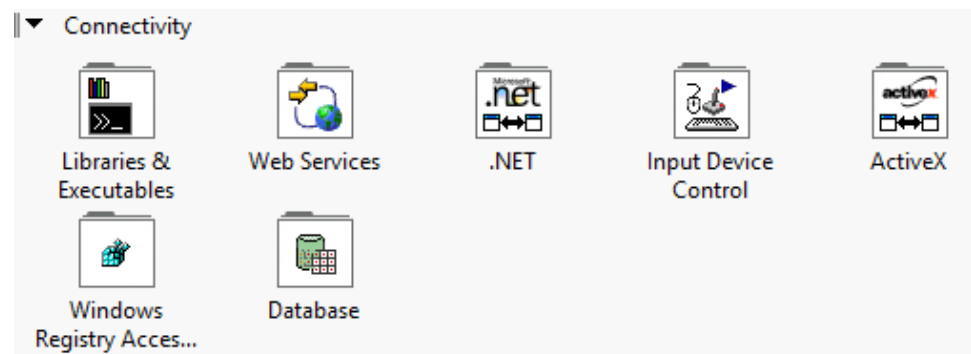
- kategoria Signal Processing - kategoria zawiera komponenty umożliwiające przetwarzanie sygnałów. Znajdziemy tutaj transformaty Laplaca i Fouriera, bloki, które umożliwią wyliczenie FFT lub DFT oraz wygenerowanie różnych sygnałów, wyznaczenie widma sygnału oraz przefiltrowania go.



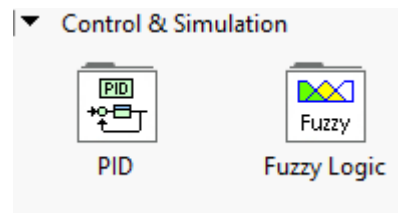
- kategoria Data Communication – odpowiada za komunikację pomiędzy danymi, synchronizację ich, tworzenie kolejki wykonania oraz współdzielenia zasobów oraz zmiennych, tworzenie zmiennych lokalnych oraz globalnych.



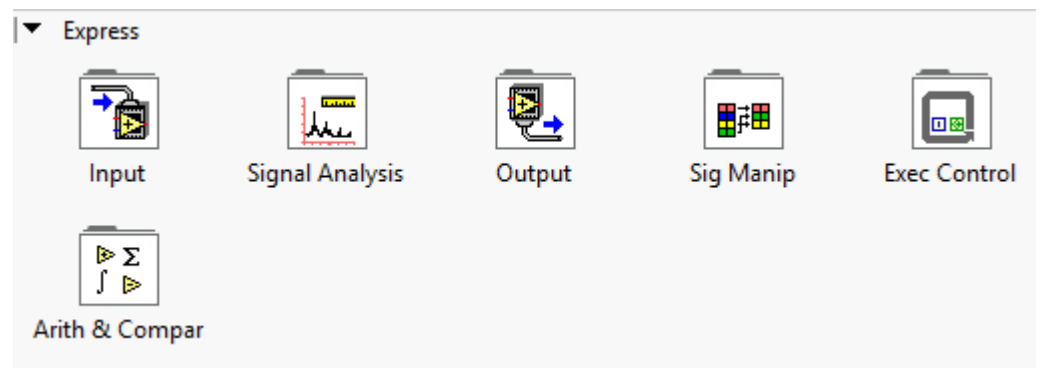
- kategoria Connectivity - obejmuje komponenty umożliwiające integracje programu LabView z interfejsami dostarczonymi przez inne środowiska programistyczne. Zawiera biblioteki pozwalające na interakcję z technologią .NET oraz ActiveX.



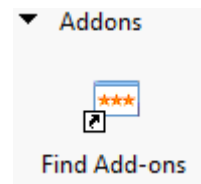
- kategoria Control & Simulation – odpowiada za symulację systemów dynamicznych, projektowanie zaawansowanych sterowników i pozwala na wdrażanie systemów sterowania do sprzętu w czasie rzeczywistym



- kategoria Express – zawiera najbardziej przydatne oraz najczęściej używane komponenty w celu ułatwienia i przyspieszenia pisania programów z kategorii opisanych powyżej

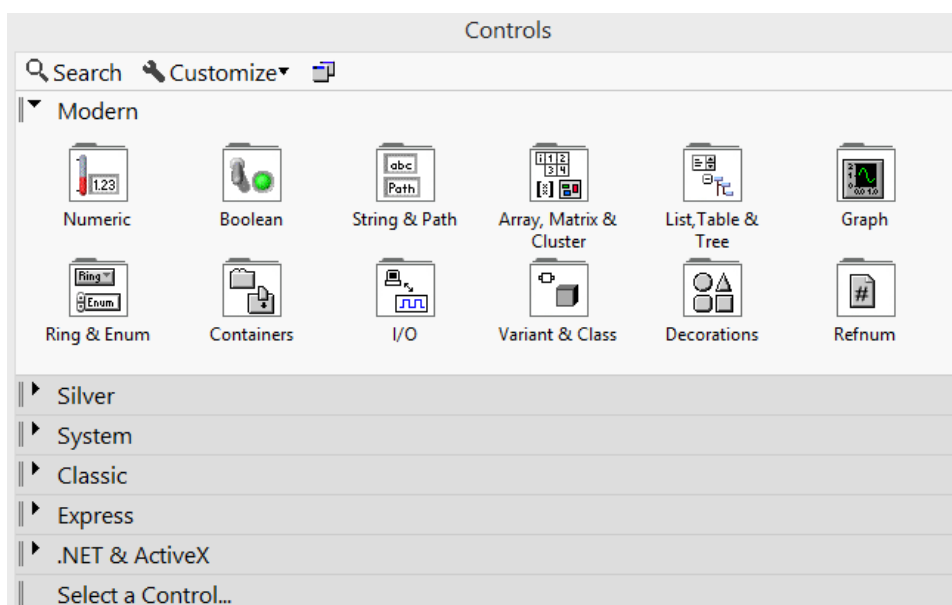


- kategoria Addons – przekierowuje nas na stronę National Instruments



Dokładniejszy opis działania poszczególnych narzędzi zostanie przedstawiony przy realizacji ćwiczeń, w których wykorzystane będą poszczególne narzędzia.

2.1.2. Controls Palette



Rysunek 2 - Widok poszczególnych kategorii ikon z palety kontrolnej LabVIEW

Funkcja poszczególnych grup narzędzi w kategorii Modern:

- a. Numeric - narzędzia służące do wpisywania i wizualizacji danych numerycznych poprzez pola liczbowe, suwaki, pokręta, mierniki itp.
- b. Boolean – reprezentacja danych binarnych poprzez przełączniki, przyciski, sygnalizatory.
- c. String & Path – wprowadzanie i wyświetlanie danych tekstowych.
- d. Array, Cluster & Matrix – tworzenie struktur, tablic i macierzy danych.
- e. List, Table & Tree - obsługa list, tabel i drzew.
- f. Graph - narzędzia służące do wizualizacji danych w postaci przebiegów sygnałów, oscyloskopów, grafów 3D.
- g. Ring & Enum - narzędzia służące do tworzenia list danych tekstowych spośród których możemy dokonywać wyboru aktualnie aktywnego.
- h. Containers - narzędzia pozwalające na wyświetlanie okna frontowego z innego VI lub do grupowania zadań i podziału ekranu w obrębie jednego VI.
- i. I/O - narzędzia służące do przekazywania nazw kanałów i danych z podłączonego urządzenia.
- j. Variant & Class – interakcje ze zmiennych oraz danymi w utworzonych klasach.

- k. Decorations – elementy bez istotnych funkcji, służące do oprawy graficznej programu.
- l. Refnum - narzędzia służące do pracy z plikami, ścieżkami, a także przekazywania danych do podprogramów.

Pozostałe kategorie takie jak Silver, System, czy Classic zawierają elementy o analogicznym działaniu, jednak ze zmienioną szatą graficzną.

2.2. Typy zmiennych

- Integer – typ całkowity ze znakiem



- a. Byte – bajt, 8-bit.
- b. Word - słowo, 16-bit.
- c. Long - podwójne słowo, 32-bit.

- Unsigned integer – typ całkowity bez znaku



- a. Unsigned byte – bajt, 8-bit.
- b. Unsigned word - słowo, 16-bit
- c. Unsigned long - podwójne słowo, 32-bit.

- Typy zmiennoprzecinkowe



- a. Rzeczywiste
 - Single precision - pojedyncza precyzja, 32-bit.
 - Double precision - podwójna precyzja, 64-bit.
 - Extended precision – rozszerzona precyzja, 128-bit.

b. Zespalone

- Complex single precision - pojedyncza precyzja
- Complex double precision - podwójna precyzja
- Complex extended precision – rozszerzona precyzja

- Boolean – typ boolowski (wartość logiczna prawda/fałsz)



- String – ciąg znaków tekstowych

