

Wstęp

Środowisko programistyczne LabVIEW znajduje zastosowanie w automatyce, robotyce, mechatronice, czyli w obszarach, w których mamy do czynienia z pomiarami, sterowaniem, akwizycją i przetwarzaniem danych oraz potrzebą integracji tych funkcji w jednym programie.

Cechy wyróżniające środowisko LabVIEW:

- Graficzny język programowania (G) – opracowanie programu sprowadza się do łączenia blozków na diagramie blokowym,
- Programowanie wysokiego poziomu – LabVIEW wyposażone jest w dużą liczbę bibliotek i blozków funkcyjnych, które pozwalają programiście skupić się na realizacji zadania stawianego przed programem, a nie na szczegółach implementacji (zarządzanie pamięcią, obsługa sterowników urządzeń),
- Szybkie prototypowanie – w LabVIEW jednocześnie powstaje zarówno diagram blokowy, określający logikę działania programu, jak również panel frontowy zawierający elementy interfejsu użytkownika. Takie podejście znacząco zwiększa efektywność tworzenia aplikacji i umożliwia powstanie pierwszego prototypu w bardzo krótkim czasie.

W środowisku **LabVIEW** można wyróżnić cztery kategorie stosowanych danych:

- **typ Boolean**, obejmujący dane logiczne
- **typ Numeri**, obejmujące dane numeryczne
- **typ String**, obejmujące dane łańcuchowe
- **typ dynamiczny**

Każdemu typowi danych odpowiada inny rodzaj połączenia („drutowania”), co zostało zaprezentowane w tabeli poniżej (Tabela 1)

TABELA 1 Typy połączeń do transferu danych między obiektami



























| Typy danych | Skalar | Macierz 1D | Macierz 2D | |
|------------------------|---|---|---|---|
| Liczbowe (Numeri) |  |  |  | Zmiennoprzecinkowe Całkowite (Integer) |
| Logiczne (Boolean) |  |  |  | Zielone |
| Łańcuchowe (String) |  |  |  | Różowe / Purpurowe |
| Dynamiczne |  |  |  | Niebieska „kratka” |

Tabela 1. Ikony i podstawowe typy zmiennych

| Ikona | Terminal | Typ |
|---|---|--|
|  |  | Typ numeryczny, liczby zmiennoprzecinkowe: <ul style="list-style-type: none"> - pojedynczej precyzji - podwójnej precyzji - rozszerzonej precyzji |
|  |  | |
|  |  | |
|  |  | Typ numeryczny, liczby całkowite ze znakiem <ul style="list-style-type: none"> - 8 bitowe - 16 bitowe - 32 bitowe - 64 bitowe |
|  |  | |
|  |  | |
|  |  | |



Typ numeryczny, liczby całkowite bez znaku

- 8 bitowe
- 16 bitowe
- 32 bitowe
- 64 bitowe



Typ logiczny, przyjmuje wartości FALSE(0) lub TRUE (1)



Typ znakowy



Tablica, która może zawierać elementy różnego typu



Klaster danych – jest odpowiednikiem struktury w języku C, grupuje dane różnych typów

- CTRL + E – przełączanie się pomiędzy oknami,
- CTRL + T – równomierne rozłożenie okien obok siebie na ekranie.

