Komputerowe techniki pomiarowe [LAB]

Wojciech Olszewski

Zajęcia 05

Hierarchiczna budowa programu

- Dobrą praktyką programowania w LabVIEW jest ograniczony rozmiar diagramu:
  - zbyt duży diagram powoduje, że program jest mało przejrzysty, trudno w nim odnaleźć błędy, dokonać zmian.

- Strukturę hierarchiczną programu wyróżnia fakt, że fragmenty kodu, stanowiące zamkniętą całość zastąpione są przez podprogramy (*SubVI*) symbolizowane na diagramie jako pojedyncze ikony.
- Podprogramy mogą wykorzystywać inne podprogramy itd. aż do procedur najniższego poziomu. Nie ma ograniczeń w ilości warstw w hierarchii.
- Poszczególne podprogramy mogą być sprawdzane/modyfikowane oddzielnie, bez konieczności ingerencji w schemat programu głównego.

Hierarchiczna budowa programu

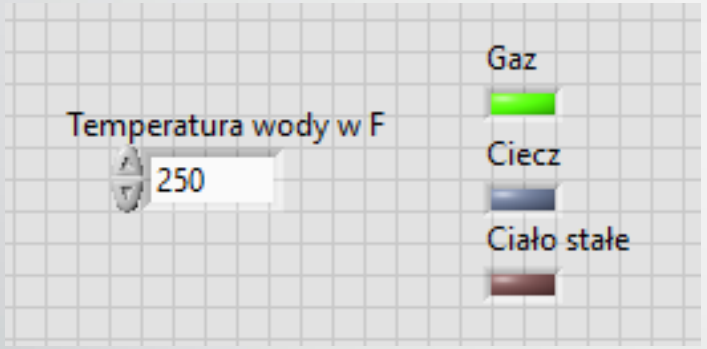
- *SubVI* stanowi osobny plik (.vi) i jest opisany w podobny sposób co *VI:*
  - zawiera wejścia, wyjścia
  - reprezentowany jest jako ikona
- **Główne zalety *SubVI*:**
  - łatwość reprezentacji skomplikowanego kawałka programu w postaci ikony, oraz lepsza czytelność programu
  - osobny plik z *SubVI* pozwala na zastosowanie w różnych projektach
- **Wady *SubVI*:**
  - zewnętrzny plik może zapodziać się i spowodować iż projekt VI nie będzie funkcjonować (jest to w prawdzie wina programisty, niemniej jednak nadmiar plików może być nieco kłopotliwy)
  - niedokładnie zdefiniowane wartości początkowe *SubVI* mogą być przyczyną nieprawidłowego funkcjonowania algorytmu

Hierarchiczna budowa programu

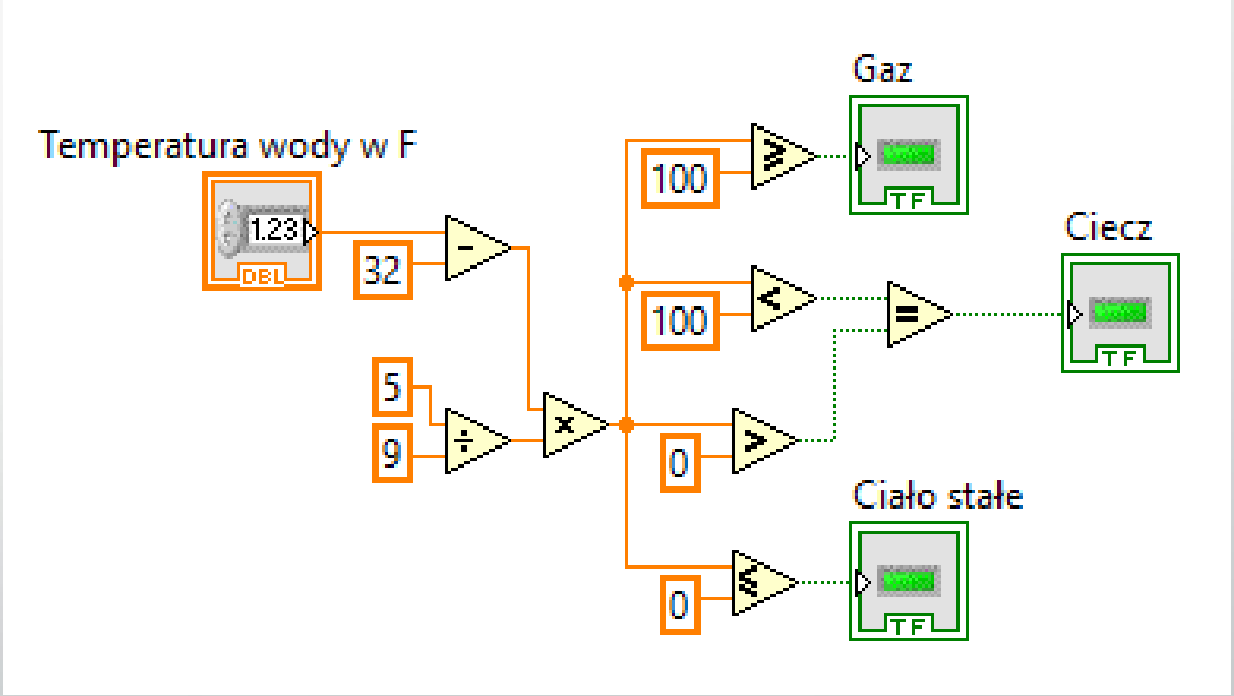
- Tworzenie *SubVI:*
  - tworzymy panel frontowy i diagram blokowy
  - sprawdzamy funkcjonowanie
  - projektujemy ikonę podprogramu
  - odpowiednim terminalom projektu przypisujemy kontrolki i wyjścia

Zadanie 16a

- Wykonaj prosty wirtualny instrument który na podstawie wprowadzonej temperatury w stopniach Fahrenheita zaświeci jedną z trzech diod (gaz, ciecz, ciało stałe). Zapisz tak przygotowany VI jako zadanie16a.vi

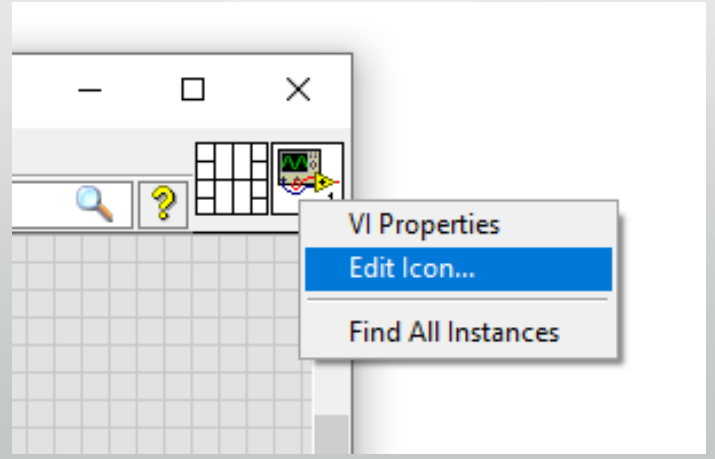


$$T_{Fahrenheit} = 32 + \frac{9}{5} T_{Celsius}$$



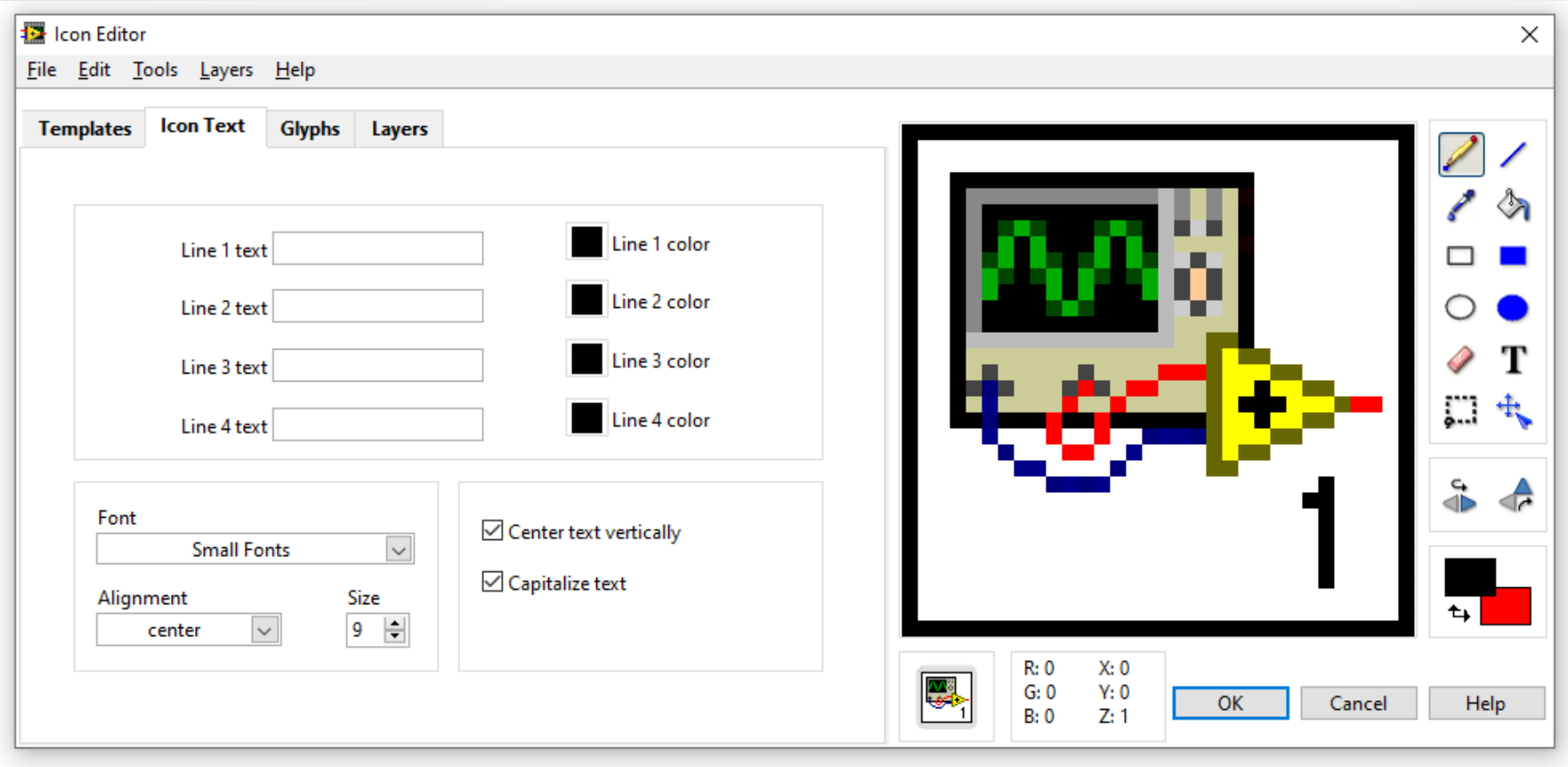
Hierarchiczna budowa programu

- Projektowanie ikony*:*
  - każdy podprogram użyty w innym programie/podprogramie powinien mieć własną ikonę pozwalającą na jego łatwe zidentyfikowanie
  - można skorzystać z ikony domyślnej, ale trudno będzie odróżnić poszczególne podprogramy
- W celu wykonania ikony klikamy prawym przyciskiem myszy na polu ikony w prawym górnym rogu panelu frontowego i wybieramy opcję *Edit Icon…*



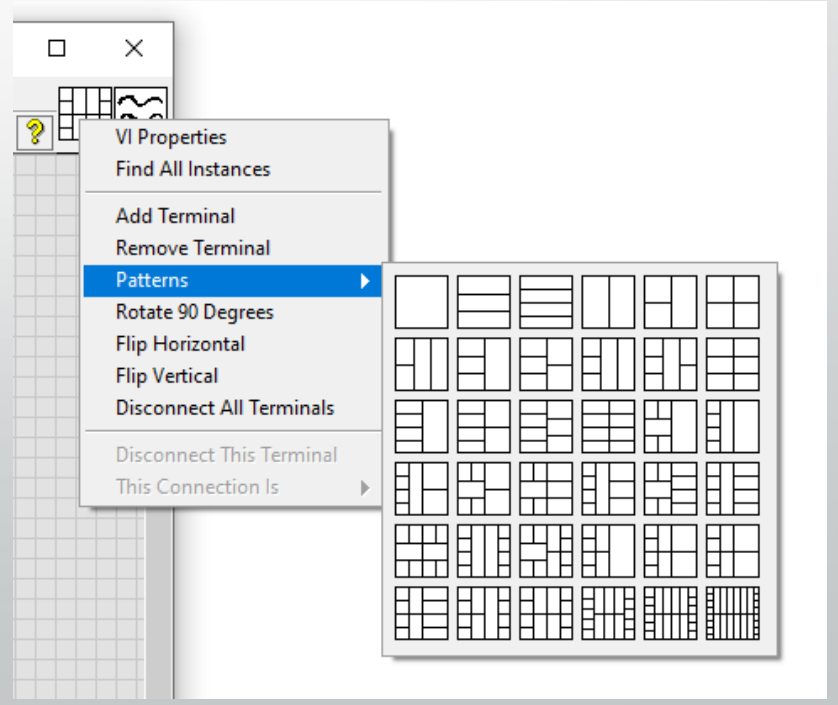
Hierarchiczna budowa programu

- Wykorzystując narzędzia dostępne w edytorze projektujemy własną ikonę.
- Ikonę można również wykonać w dowolnym innym edytorze i skopiować ją z wykorzystaniem funkcji Schowka.



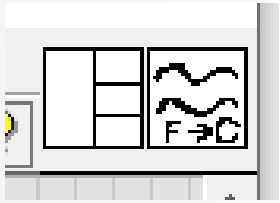
Hierarchiczna budowa programu

- Definicja zacisków podprogramu:
  - Należy zdefiniować ile wejść i wyjść ma mieć nasz podprogram
- W tym celu klikamy prawym przyciskiem myszy na polu zacisków w prawym górnym rogu panelu frontowego, wybieramy opcję *Patterns*, a następnie interesujący nas predefiniowany przypadek.
- Parametry wejściowe podłączone są do zacisków z lewej strony, wyjściowe do zacisków z prawej strony.
- Pojedynczy zacisk można dodać/usunąć przez polecenie *Add/Remove Terminal.*
- Zaciski można również obrócić lub wykonać ich lustrzane odbicie.



Hierarchiczna budowa programu

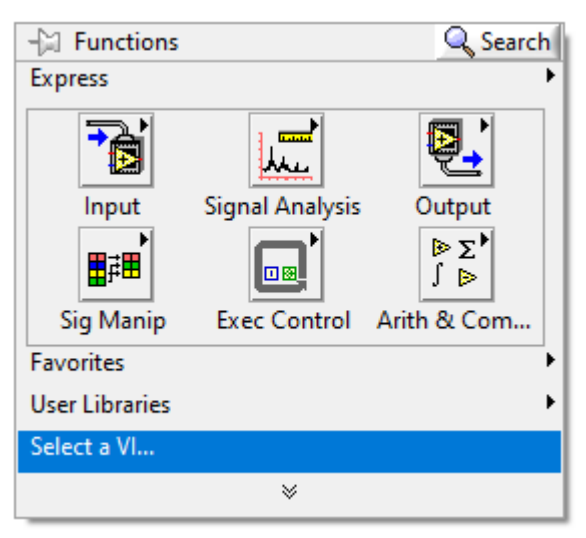
- W przypadku przykładu 16a mamy 1 kontrolkę wejścia (temp. w F) oraz 3 wskaźniki wyjścia (diody). Potrzebujemy zatem następującego układu zacisków:



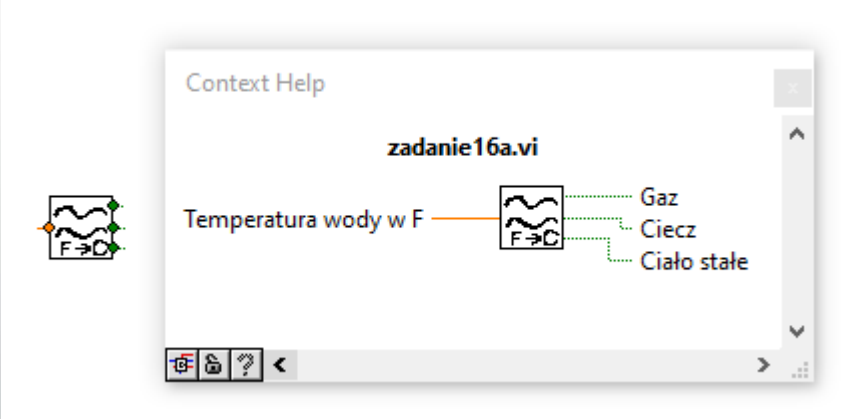
- Aby przypisać obiekty na panelu poszczególnym zaciskom, należy:
  - kursorem w postaci szpulki kliknąć w wybrane pole zacisku (pole zmieni kolor na czarny),
  - kliknąć obiekt na panelu frontowym odpowiadający polu zacisku (pole zmieni kolor na odpowiadający typowi zmiennej),
  - kliknąć na wolnym miejscu na panelu frontowym,
  - czynności powtórzyć dla wszystkich zacisków.

Hierarchiczna budowa programu

- Gotowy podprogram, po utworzeniu ikony i zacisków należy zapisać do pliku.
- Następnie aby go wykorzystać, na diagramie blokowym programu głównego należy wybrać z palety *Functions* pozycję *Select a VI* i wskazać wybrany podprogram.

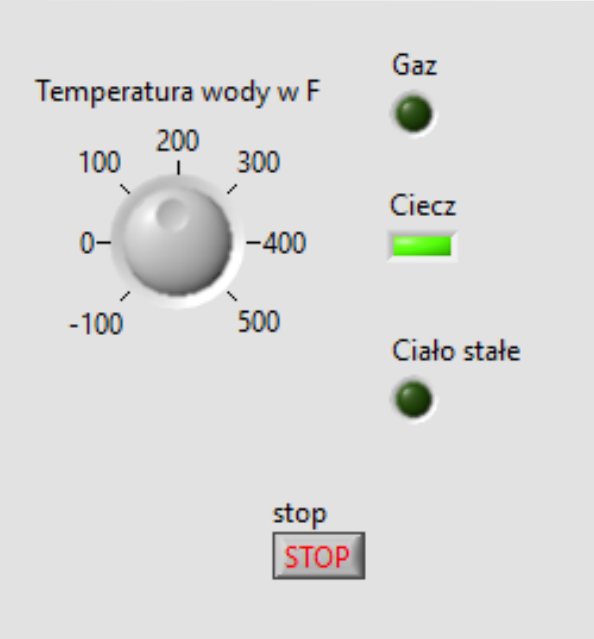


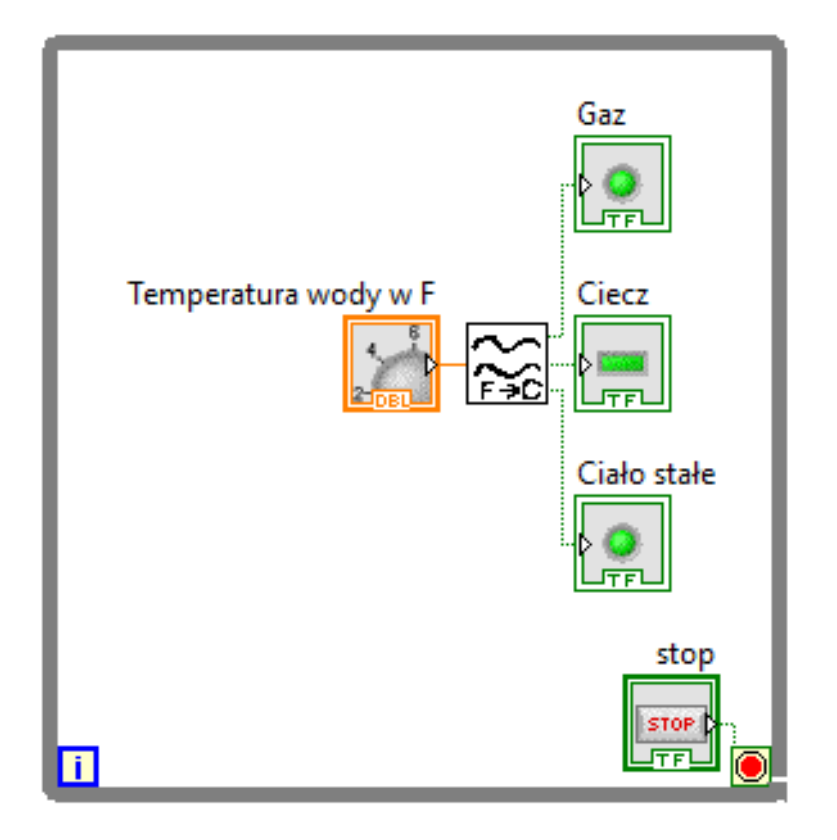
ikona reprezentująca podprogram
wraz z pomocą opisującą poszczególne zaciski
(Ctrl + H)



Zadanie 16b

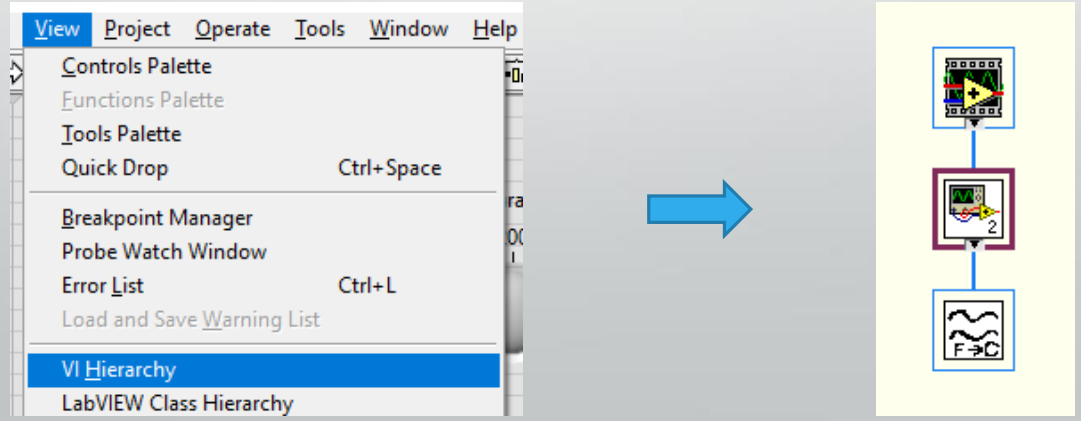
- Wykonaj to samo zadanie z wykorzystaniem utworzonego wcześniej podprogramu.





Hierarchiczna budowa programu

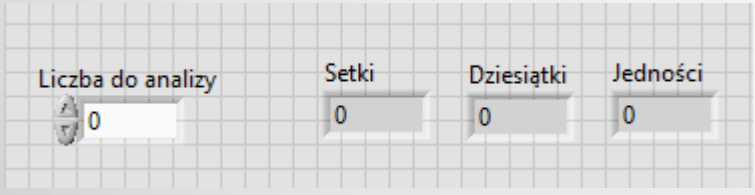
- Istnieją trzy metody programowania:
  - z góry do dołu (*top-down*) – planuje się całość programu, a następnie stopniowo przechodzi się do szczegółów,
  - z dołu do góry (*bottom-up*) – polega na składaniu całości programu głównego z wcześniej przygotowanych podprogramów,
  - lub z wykorzystaniem obu powyższych jednocześnie.
- Aby zobaczyć hierarchię programu wraz z wzajemnymi powiązaniami między poszczególnymi podprogramami należy wybrać opcję *View – Vi Hierarchy*

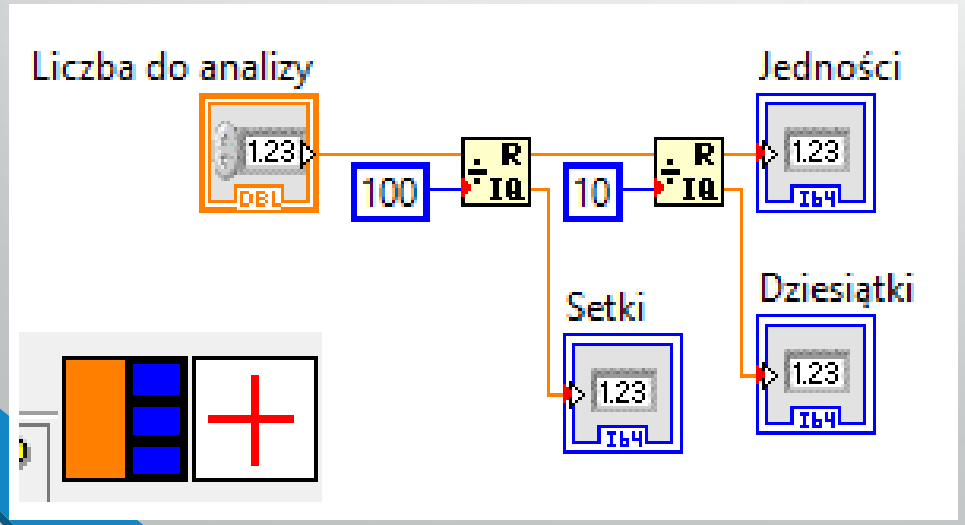


Zadanie 17

- Stworzyć narzędzie wirtualne, które po otrzymaniu liczby rzeczywistej zwróci 3 cyfry odpowiadające odpowiednio ilości jednostek, dziesiątek i setek danej liczby rzeczywistej.
  Następnie użyć tego narzędzia w osobnym narzędziu wirtualnym, które wyświetli sumę cyfr, z których składa się wprowadzona liczba rzeczywista.

Zadanie17a.vi





Zadanie17b.vi