



Politechnika Wroclawska

Laboratorium Wirtualne Przyrządy Pomiarowe

Ćwiczenie 1

Zajęcia wstępne

Zagadnienia do przygotowania

1. Interfejs LabVIEW: VI, Panel przedni (Front Panel), Diagram blokowy (Block Diagram), SubVI
2. Elementy panelu przedniego: Kontrolka (Control), Indykator (Indicator)
3. Podstawowe struktury danych: Tablica (Array), Klaster (Cluster)
4. DataFlow.

Spis treści

1 BHP i zasady zaliczenia	1
2 LabVIEW	1
2.1 Modułowość kodu	1
2.2 Wejścia/wyjścia SubVI i DataFlow	2
3 Zadania:	2
3.1 Przemiatanie	2
3.2 Rysowanie wykresów	2

mgr inż. Krzysztof Kwoka (krzysztof.kwoka@pwr.edu.pl)

Wydział Elektroniki Mikrosystemów i Fotoniki

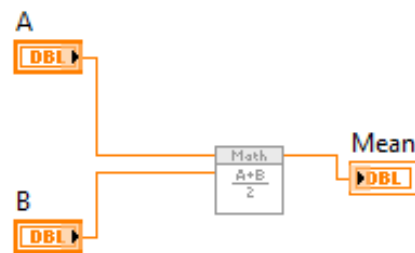
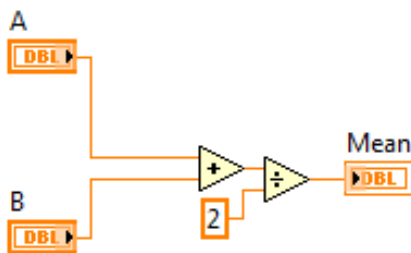
1 BHP i zasady zaliczenia

Na pierwszych zajęciach w semestrze podane zostaną zasady zaliczenia i przeprowadzone będzie szkolenie BHP.

2 LabVIEW

2.1 Modułowość kodu

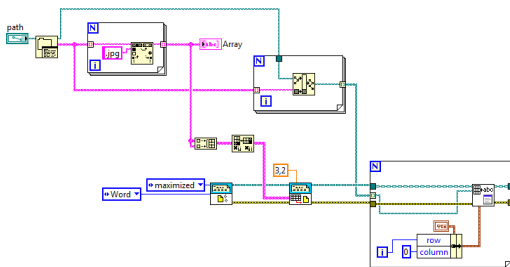
Każdy program napisany w LabVIEW (VI — *Virtual Instrument*) może stanowić podprogram (SubVI — *Virtual Instrument*). Dzięki temu działające porcje kodu można wykorzystywać w wielu projektach bez potrzeby przepisywania tych samych funkcji. Przykład:



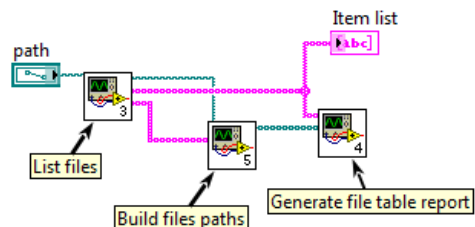
Rysunek 1: Średnia arytmetyczna jako kod

Rysunek 2: Średnia arytmetyczna jako SubVI

Modułowość służy także do poprawy czytelności kodu, przez zbieranie kompletnych fragmentów programu w mniejsze, gotowe funkcje. Przykład:

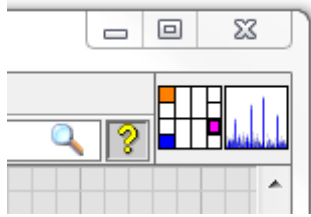


Rysunek 3: Cały kod programu



Rysunek 4: Kod zebrany do podprogramów

Aby stworzyć SubVI można zaznaczyć potrzebny fragment kodu i wybierając z paska menu *Edit* użyć opcji *Create Sub VI*. Drugą metodą jest samodzielne napisanie programu i przypisanie kontrolki do wejść i wyjść na terminalu połączeń w prawym górnym rogu panelu frontowego. Niezależnie od tego w jaki sposób zbudowany zostanie SubVI należy zmienić mu ikonę i ustawić opis.



Rysunek 5: Zmiana ikony i połączenie terminala

2.2 Wejścia/wyjścia SubVI i DataFlow

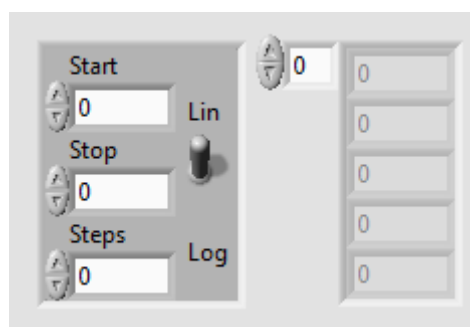
Podczas tworzenia kodu używając środowiska LabVIEW należy szczególną uwagę zwracać na specyficzne podejście do czytelności kodu. Wynika ono zarówno z zaprezentowanej wcześniej modułowości kodu, ale także idei przepływu danych (ang. *Data Flow*). Aby uprościć kod do postaci bardziej czytelnej dla użytkownika należy stosować kilka zasad:

- Dane "płyną" z lewej do prawej i nie zwracają (w konsekwencji wejścia SubVI powinny znajdować się po lewej, a wyjścia po prawej stronie),
- Nie należy krzyżować przewodów tam gdzie można tego uniknąć,
- Przewody powinny mieć jak najmniej zgięć,
- Najlepiej stosować domyślny terminal połączeń,
- Linie błędów należy prowadzić przez dolne narożniki SubVI,
- Ikony SubVI powinny mieć ramki i należy być konsekwentnym w ich projektowaniu,
- Kolorowanie kodu nie jest najlepszym pomysłem.

3 Zadania:

3.1 Przemiatanie

Wykonać SubVI z jednym wejściem i jednym wyjściem. Na wejście przyjmowany ma być klaster zawierający kontrolki numeryczne określające wartość początkową i końcową, liczbę kroków oraz kontrolkę typu boolean ustawiającą tryb przemiatania: liniowy bądź logarytmiczny. Na wyjściu SubVI należy otrzymać tablicę z kolejnymi krokami przemiatania. Przykładowy panel frontowy:



Rysunek 6: Przykładowy panel frontowy do zadania 1

3.2 Rysowanie wykresów

Używając SubVI z zadania 1 wykreślić wynik używając indykatora Waveform Graph oraz XY Graph.