

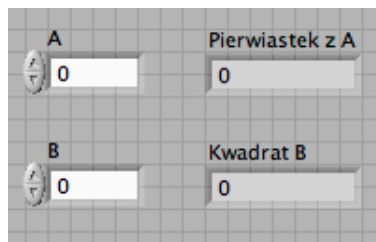
Zadania - LabView

LabVIEW

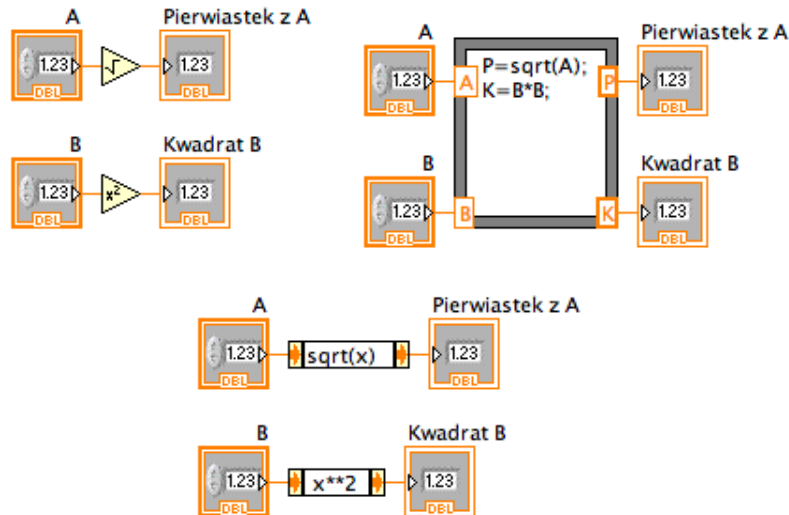
Zad. 1. Napisz program `pierwiastek_i_kwadrat.vi` (rys. 1 i 2) umożliwiający wprowadzenie dwóch liczb i obliczający pierwiastek kwadratowy pierwszej oraz kwadrat drugiej liczby.

W schemacie programu skorzystaj z obiektów:

- kwadratu liczby (Programming → Numeric → Square) i pierwiastka kwadratowego (Programming → Numeric → Square Root),
- obiektu umożliwiającego zapisanie wyrażeń matematycznych (Programming → Structures → Formula Node lub — dla wyrażeń jednej zmiennej — Programming → Numeric → Expression Node). W LabView 2019 lokalizacja obiektów jest inna: Mathematics → Scripts & Formulas
- pola tekstowego umożliwiającego wprowadzenie liczby (menu kontekstowe złącza → Create → Control),
- pola tekstowego umożliwiającego prezentację liczby (menu kontekstowe złącza → Create → Indicator).



Rysunek 1: Wygląd panelu sterującego programu `pierwiastek_i_kwadrat.vi`.



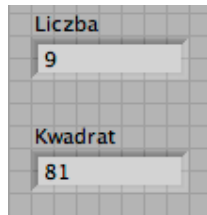
Rysunek 2: Schematy trzech wersji programu pierwiastek_i_kwadrat.vi.

Zad. 2. Napisz program `suma_kwadratow.vi` umożliwiający wprowadzenie dwóch liczb i obliczający sumę ich kwadratów. Skorzystaj z obiektu `Formuła` z palety `Express`. Następnie znajdź w paletce `Express` obiekt `Simulate Signal`. Użyj dwóch egzemplarzy takiego obiektu do generacji dwóch sygnałów sinusoidalnych o różnych parametrach, a następnie dodaj do siebie te sygnały korzystając z obiektu `Formuła`.

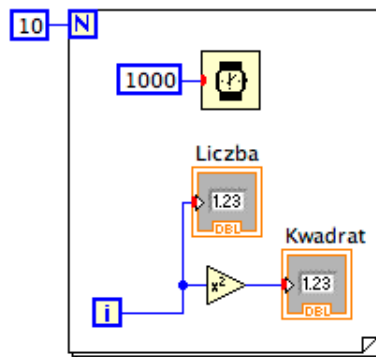
Zad. 3. Napisz program `kwadraty.vi` (rys. 3 i 4) obliczający kwadraty pierwszych 10 liczb całkowitych $0, 1, \dots, 9$. Pomiedzy kolejnymi obliczeniami wprowadź opóźnienie o długości 1 sekundy.

W schemacie programu skorzystaj z obiektów

- pętli `for` (`Programming` → `Structures` → `For Loop`),
- stałej liczbowej (menu kontekstowe złącza → `Create` → `Constant`),
- obiektu wstrzymującego pracę programu na n milisekund (`Programming` → `Timing` → `Wait`).



Rysunek 3: Wygląd panelu sterującego programu kwadraty.vi.



Rysunek 4: Schemat programu kwadraty.vi.

Zad. 4. Napisz program `pierwiastek_z_warunkiem.vi` (rys. 5 i 6) obliczający pierwiastek kwadratowy liczby. W przypadku podania liczby ujemnej program powinien wypisać komunikat `Błędna dana wejściowa!` i zakończyć działanie. Następnie stwórz ikonę odpowiadającą programowi

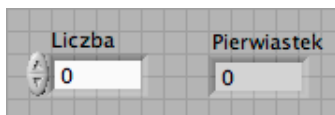


tak, aby możliwe było użycie programu w innym programie VI.

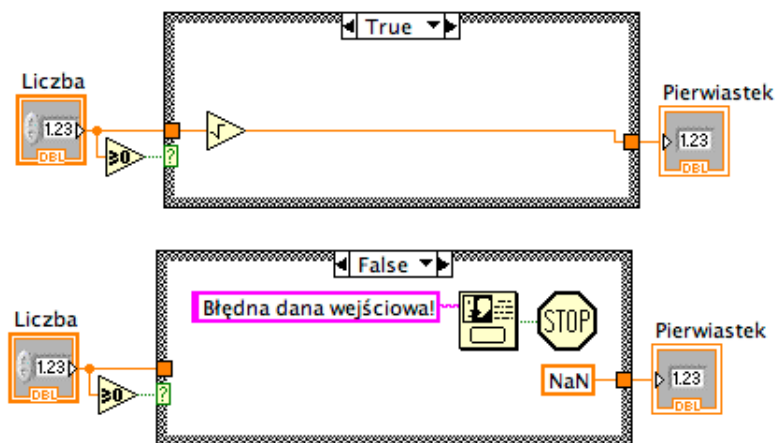
W schemacie programu skorzystaj z obiektów

- instrukcji warunkowej (Programming → Structures → Case Structure),
- warunku $x \geq 0$ (Programming → Comparison → Greater or Equal To 0),
- okienka dialogowego do wyświetlenia komunikatu (Programming → Dialog & User Interface → One Button Dialog),

- obiektu przerywającego pracę programu (Programming → Application Control → Stop).



Rysunek 5: Wygląd panelu sterującego programem pierwiastek z warunkiem.vi.



Rysunek 6: Schemat programu pierwiastek z warunkiem.vi.

Zad. 5. Korzystając z programu pierwiastek_z_warunkiem.vi napisz program pierwiastek_w_petli (rys. 7 i 8), w którym na bieżąco obliczaj wartość wyrażenia $1 + \sqrt{x}$.

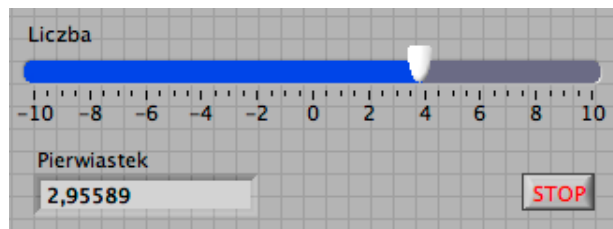
W panelu sterującym skorzystaj z obiektów

- suwaka zadającego wartość x w zakresie od -10 do 10 (Express → Numeric Controls → Horizontal Pointer Slide),
- przycisku do zatrzymania pracy programu (Express → Buttons & Switches → Stop Button).

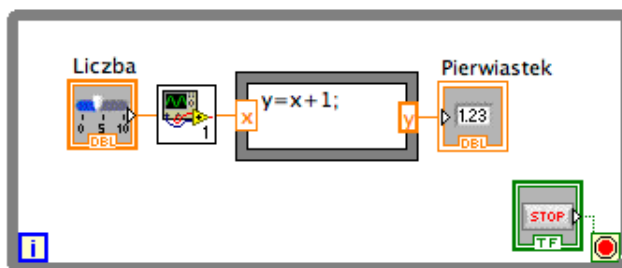
W schemacie programu skorzystaj z obiektów

- pętli while (Programming → Structures → While Loop),

- ikony reprezentującej program `pierwiastek_z_warunkiem.vi`.



Rysunek 7: Wygląd panelu sterującego programu `pierwiastek_w_petli.vi`.



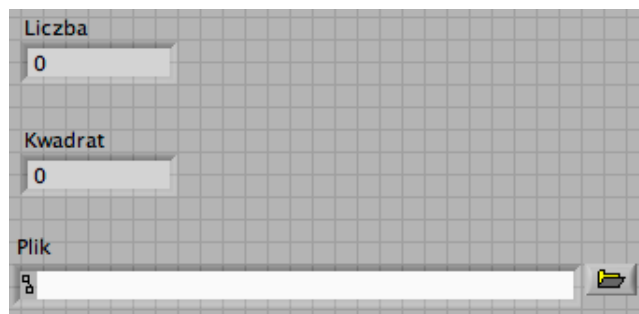
Rysunek 8: Schemat programu `pierwiastek_w_petli.vi`.

Zad. 6. Na podstawie programu `kwadraty.vi` napisz program `kwadraty_z_zapisem_do_pliku.vi` (rys. 9 i 10) zapisujący wyniki obliczeń do pliku tekstowego. W kolejnych wierszach zapisuj liczbę i jej kwadrat oddzielone spacją.

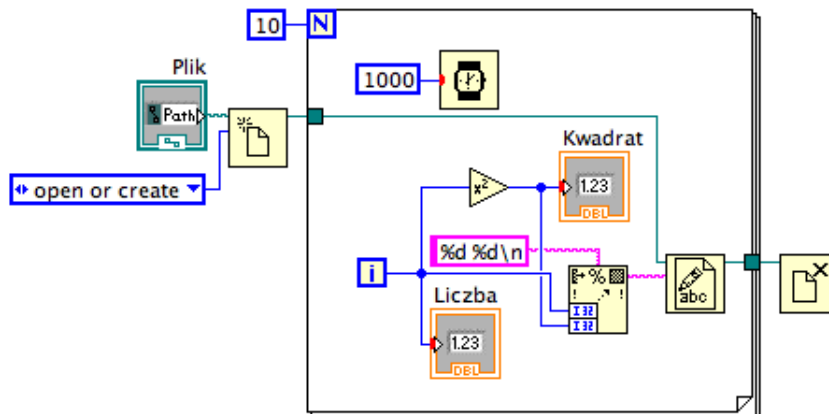
W schemacie programu skorzystaj z

- obiektu przygotowującego plik do użycia (Programming → File I/O → Open/Create/Replace File).
- Obiektu realizującego czynność zapisu do pliku (Programming → File I/O → Write To Text File).
- Obiektu kończącego użycie pliku (Programming → File I/O → Close File).
- Obiektu przygotowującego napis (Programming → String → Format Into String). Do złącza format string należy podać napis, w którym mogą występować symbole specjalne, m.in.:

- %d — liczba całkowita,
- %f — liczba zmiennopozycyjna (rzeczywista),
- %s — napis,
- %. ; — stosowanie kropki jako znaku rozdzielającego część całkowitą i ułamkową liczby,
- \n — koniec wiersza.



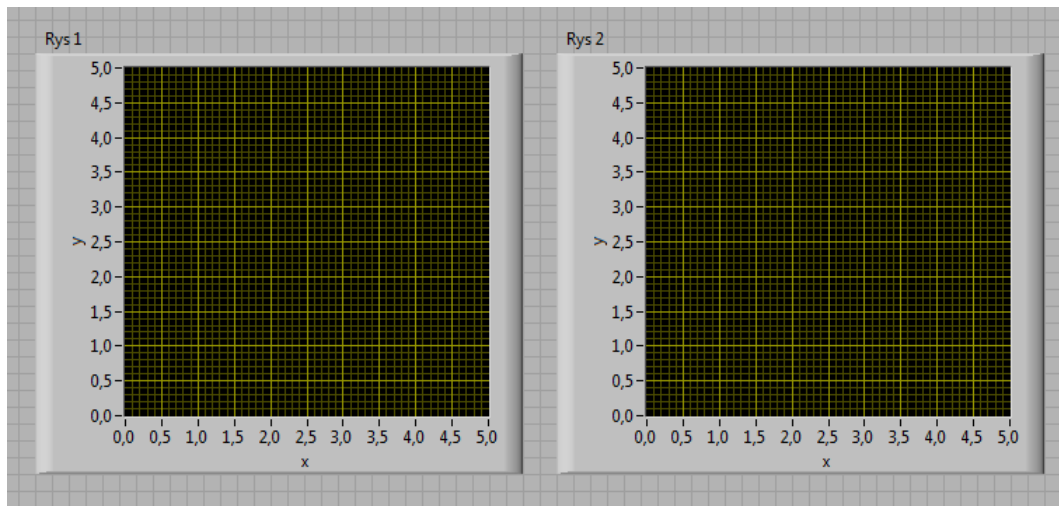
Rysunek 9: Wygląd panelu sterującego programu kwadraty_z_zapisem_do_pliku.vi.



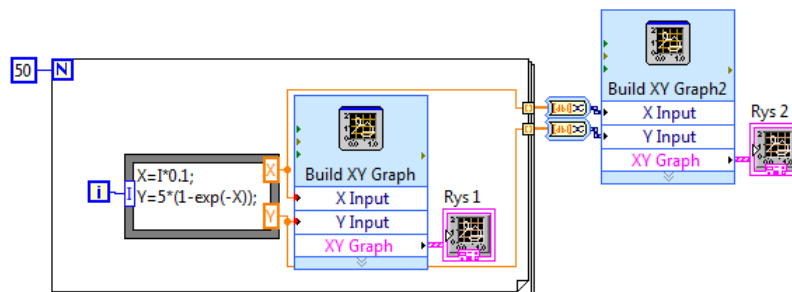
Rysunek 10: Schemat programu kwadraty_z_zapisem_do_pliku.vi.

Zad. 7. Napisz program `suma.vi` sumujący liczby całkowite od 1 do 100. Skorzystaj z pętli `for`. Użyj elementu *shift register* tej pętli.

Zad. 8. Napisz program wykres1 (rys. 11 i 12) rysujący wykres funkcji $y(x) = 5(1 - e^{-x})$ dla $x \in \{0, 0.1, 0.2, \dots, 4.8, 4.9\}$.



Rysunek 11: Wygląd panelu sterującego programu wykres1_express.



Rysunek 12: Schemat programu wykres1_express.

Zbuduj program przy użyciu obiektów z palety Express. W panelu sterującym użyj obiektów: Express → GraphIndicators → XY Graph. W schemacie programu, we własnościach (Properties) obiektu Build XV Graph eksperymentuj ze zmianą ustawienia: Clear Data on Each Call.

Zad. 9. Znając teoretyczne wartości tłumienia

$$t(f) = \frac{1}{\sqrt{(2\pi f RC)^2 + 1}}$$

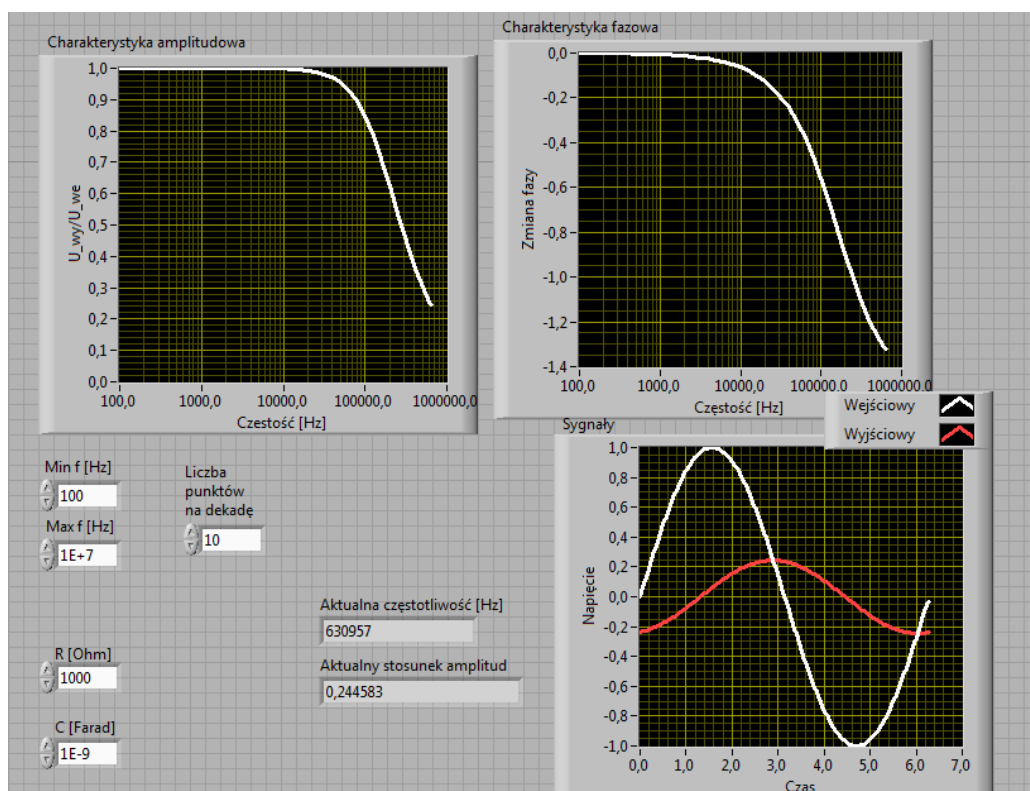
i przesunięcia fazy

$$\varphi(f) = \arctan(-2\pi fRC)$$

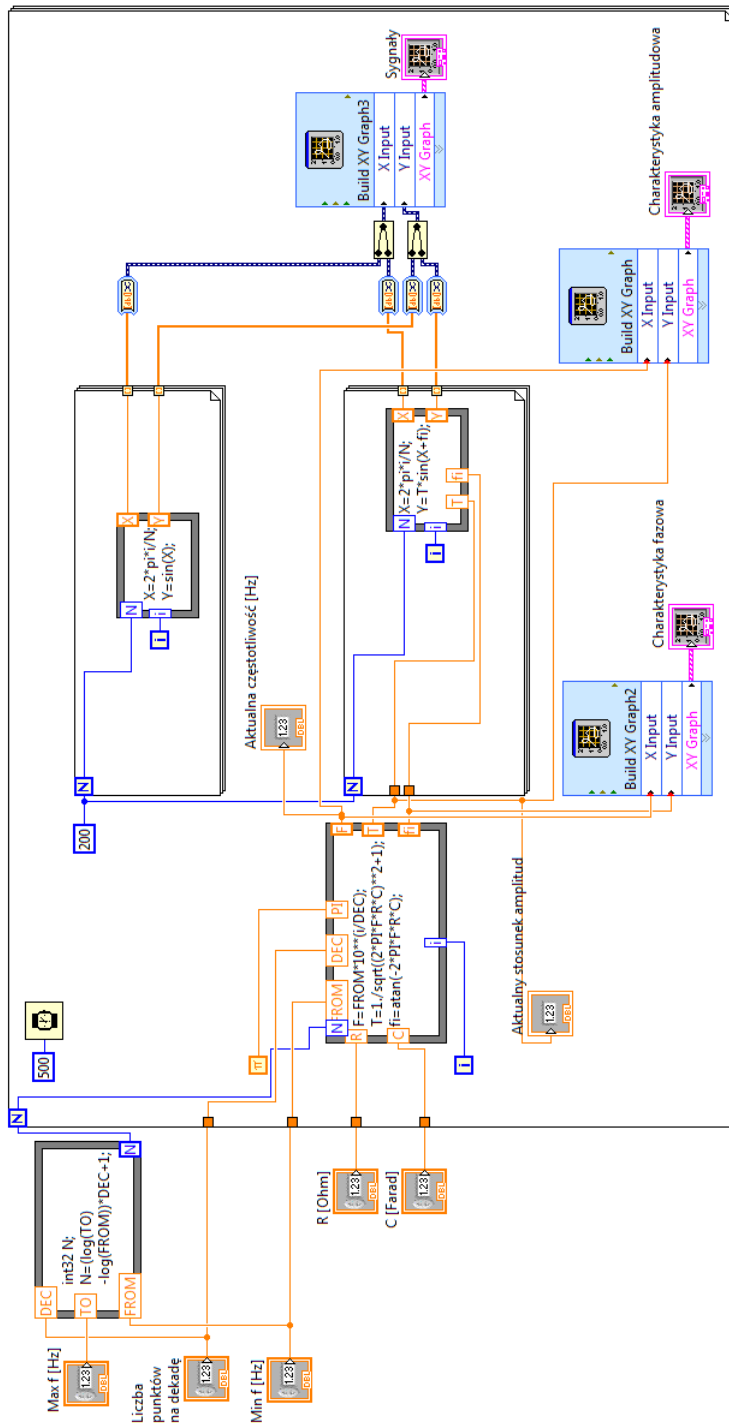
sygnału wyjściowego w filtrze dolnoprzepustowym RC napisz program teoria_RC (rys. 13 i 14) będący komputerową symulacją doświadczenia pomiaru charakterystyk układu. Uprość program w porównaniu do tego pokazanego na rysunkach, tworząc subvi sinus.vi przygotowujący dane do wykresu funkcji

$$y = T \sin(x + \phi) \text{ dla } x \in 0 \dots 2\pi$$

i używając go dwukrotnie.



Rysunek 13: Wygląd panelu sterującego programu teoria_RC_express.



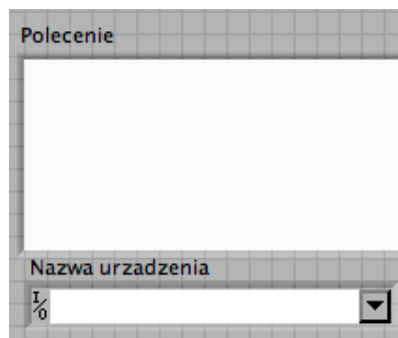
Rysunek 14: Schemat programu teoria_RC_express.

Wprowadzenie:

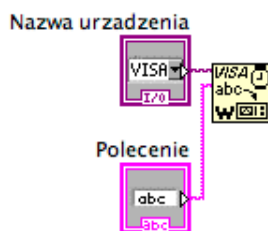
Poniżej pokazane są schematy dwóch najprostszych programów pozwalających na komunikację z urządzeniami pomiarowymi (rys. 15 i 16) i (rys. 17 i 18). Programy pozwalają na przesłanie polecenia (lub zapytania) do urządzenia pomiarowego i (w przypadku zapytania) na odebranie odpowiedzi.

Programy wykorzystują obiekty:

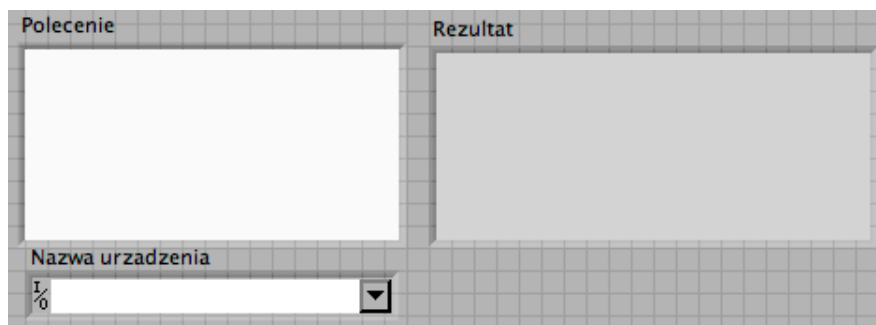
- przesłania polecenia do urządzenia (Instrument I/O → VISA → VISA Write),
- odczytania odpowiedzi z urządzenia (Instrument I/O → VISA → VISA Read).



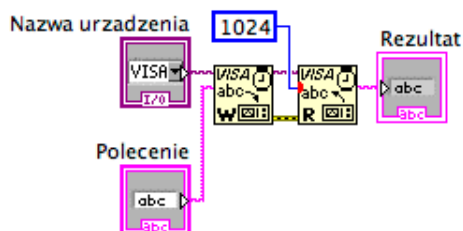
Rysunek 15: Wygląd panelu sterującego programu polecenie.



Rysunek 16: Schemat programu polecenie.



Rysunek 17: Wygląd panelu sterującego programu zapytanie.



Rysunek 18: Schemat programu zapytanie.

Zad. 10. Napisz program **generator** sterujący pracą generatora dostępnego na pracowni, ustawiający amplitudę sygnału 10 V oraz pozwalający na zmianę częstotliwości w zakresie 100...1000 Hz. W schemacie programu skorzystaj z obiektów sterownika generatora:

- przygotowującego generator do pracy (**Initialize**),
- załączającego sygnał wyjściowy (**Action-Status** → **Enable Output**),
- ustalającego parametry generowanego sygnału (**Configure** → **Configure Standard Waveform**).
- Do pierwszego kanału oscyloskopu podłącz sygnał z generatora.

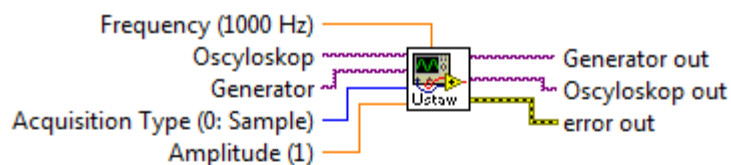
Zad. 11. Do programu **generator** dodaj z palety **Instrument I/O** → **Instrument Drivers** → ... obiekt przygotowujący oscyloskop do pracy (**Initialize**) i korzystając z poleceń opisanych w instrukcji <http://www.fuw.edu.pl/~kaste/ipw/scpi.pdf> lub w innych instrukcjach typu **Programming Guide** włącz kanał pierwszy.

Zad. 12. Do programu z poprzedniego zadania dodaj ustawienie wzmocnienia na pierwszym kanale oscyloskopu i ustawienie podstawy czasu. Ustawienia muszą zależeć odpowiednio od amplitudy i częstotliwości sygnału z generatora, tak żeby w oknie oscyloskopu zawsze były widoczne dwa okresy i amplituda peak-to-peak była większa niż połowa wysokości okna oscyloskopu. Użyj obiektów VISA i obiektu zamieniającego liczby na napis, analogicznie jak w przypadku zapisu do pliku.

Zad. 13. Napisz podprogram (subvi) *ustaw*, w którym wykonywane są wszystkie takie ustawienia początkowe dla generatora i obu kanałów oscyloskopu, jakie były robione przy pomiarach bez użycia programu Labview:

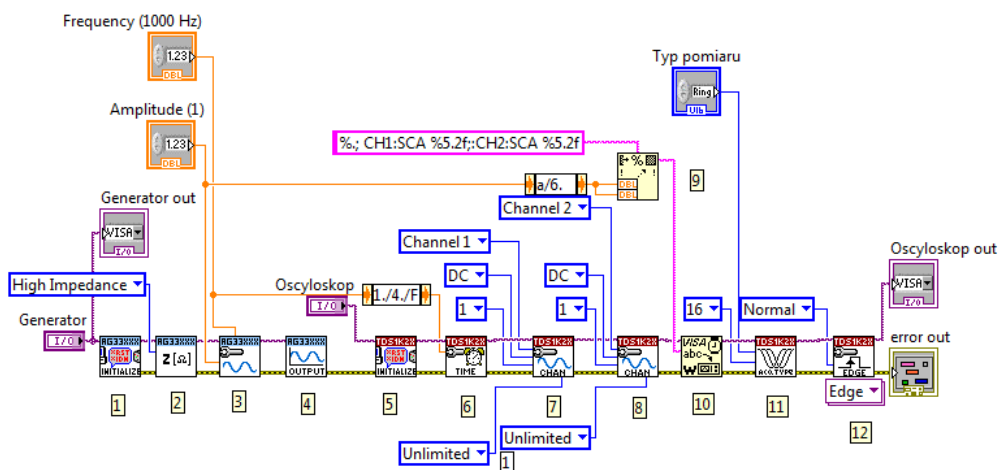
1. inicjalizacja generatora (czyszczenie buforów)
2. ustawienie wyjściowej impedancji generatora
3. ustawienia sygnału z generatora (częstotliwość, amplituda, offset)
4. włączenie wysyłania sygnału z generatora
5. inicjalizacja oscyloskopu (czyszczenie buforów)
6. konfiguracja podstawy czasu
7. konfiguracja ustawień pierwszego kanału oscyloskopu
8. konfiguracja ustawień drugiego kanału oscyloskopu
9. ustawienie takiego samego wzmocnienia dla obu kanałów (w zależności od amplitudy sygnału): zamiana liczby na ciąg znaków wysyłanych następnie do urządzenia i przesłanie polecenia do urządzenia
10. ustawienia typu pomiaru (uśrednianie)
11. ustawienia wyzwalania sygnału

Sprawdź działanie programu włączając ikonę z żarówką. Stwórz podprogram (subvi) (rysunek 19).



Rysunek 19: Podprogram *ustaw*.

Przykład podprogramu ustaw z wykorzystaniem sterowników generatora Agilent i oscyloskopu Tektronix (rysunek 20):



Rysunek 20: Schemat programu ustaw.

Opis schematu programu (rys. 20):

1. inicjalizacja generatora
2. ustawienie wyjściowej impedancji generatora
3. ustawienia sygnału z generatora (częstotliwość, amplituda)
4. ustawienie otwartego/zamkniętego wyjścia z generatora
5. inicjalizacja oscyloskopu
6. konfiguracja podstawy czasu
7. konfiguracja ustawień pierwszego kanału oscyloskopu
8. konfiguracja ustawień drugiego kanału oscyloskopu
9. ustawienie takiego samego wzmocnienia dla obu kanałów (w zależności od amplitudy sygnału): zamiana liczby na ciąg znaków wysyłanych następnie do urządzenia
10. przesłanie polecenia do urządzenia
11. ustawienia typu pomiaru

12. ustawienia wyzwalania sygnału

Zad. 14. Napisz program charakterystyki, który w pętli wykonuje pomiary potrzebne do wykonania charakterystyk: amplitudowej i fazowej. Elementy programu:

1. wyznaczanie liczby obiegu pętli na podstawie zadeklarowanego zakresu częstości i liczby punktów na dekadę (jak w programie `teoria_RC`)
2. początkowe ustawienia oscyloskopu i generatora (podprogram `ustaw`)
3. w środku pętli:
 - ustawienie aktualnej częstości na generatorze (wyliczanie częstości dla których będą mierzone amplitudy sygnałów i przesunięcie fazowe jak w programie `teoria_RC`)
 - ustawienie podstawy czasu (dobór podstawy czasu do aktualnej częstości jak w podprogramie `ustaw`)
 - ustawienia zbierania danych i włączenie zbierania danych
 - pomiary amplitudy peak-to-peak dla obu kanałów (informacja bezpośrednio z oscyloskopu)
 - pomiary amplitudy, fazy i częstości za pomocą precyzyjnej analizy przebiegów robionej przez LabView (`Fetch Waveform` i `Signal Processing` → `Waveform Measurements` → `Extract Single Tone Information`)
 - ustawianie wzmocnień obu kanałów na podstawie zmierzonej amplitudy