

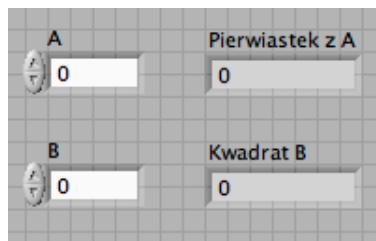
Technologie informacyjne i komunikacyjne R

LabVIEW

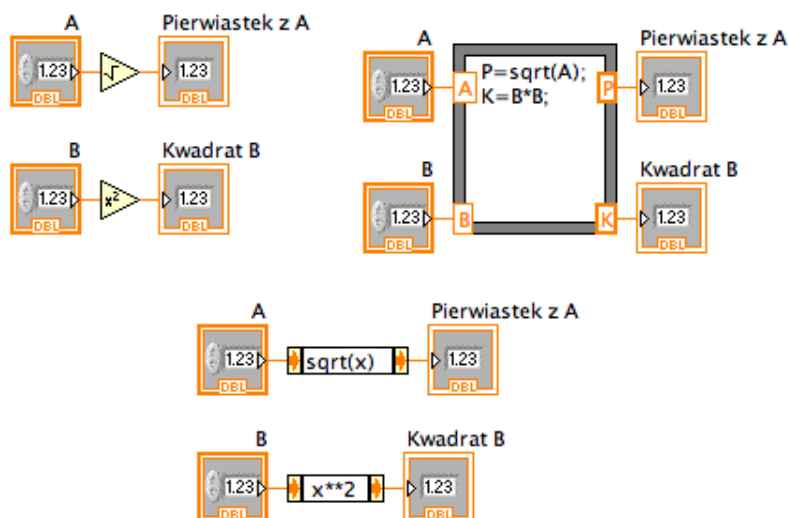
Zad. 1. Napisz program pierwiastek i kwadrat (rys. 1 i 2) umożliwiający wprowadzenie dwóch liczb i obliczający pierwiastek kwadratowy pierwszej oraz kwadrat drugiej liczby.

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),
- 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.

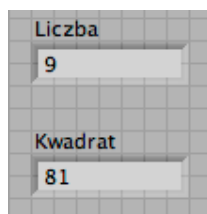


Rysunek 2: Schematy trzech wersji programu pierwiastek i kwadrat.

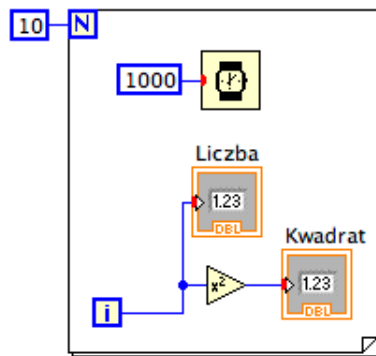
Zad. 2. Napisz program **kwadraty** (rys. 3 i 4) obliczający kwadraty pierwszych 10 liczb całkowitych $0, 1, \dots, 9$. Pomiedzy kolejnymi obliczeniami wprowadź opóźnienie o długości 1 sekundy.

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 programem kwadraty.



Rysunek 4: Schemat programu kwadraty.

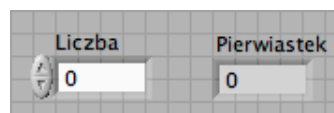
Zad. 3. Napisz program **pierwiastek z warunkiem** (rys. 5 i 6) obliczający pierwiastek kwadratowy liczby. W przypadku podania liczby ujemnej program powinien wypisać komunikat **Błędna dana wejściowa!** i zakończyć działanie. Następnie stwórz ikonę odpowiadającą programowi



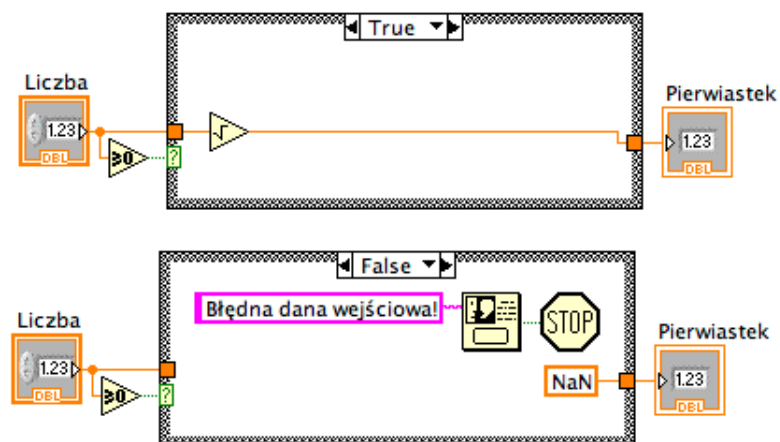
tak, aby możliwe było użycie programu w innym programie VI.

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 programu **pierwiastek z warunkiem**.

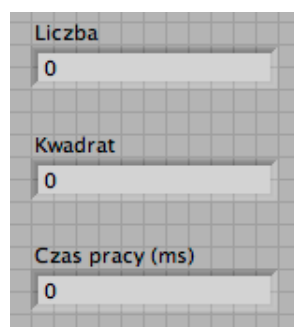


Rysunek 6: Schemat programu pierwiastek z warunkiem.

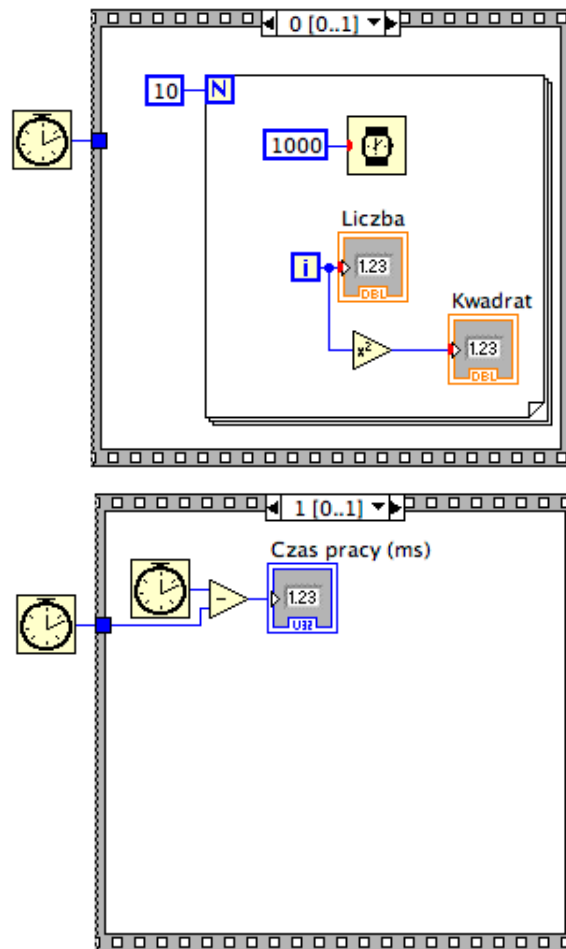
Zad. 4. Na podstawie programu kwadraty napisz program kwadraty z pomiarem czasu (rys. 7 i 8), w którym wprowadź pomiar czasu działania programu.

W schemacie programu skorzystaj z obiektów

- instrukcji sekwencyjnej (Programming → Structures → Stacked Sequence Structure),
- obiektu przekazującego aktualne wskazanie zegara w ms (Programming → Timing → Tick Count).



Rysunek 7: Wygląd panelu sterującego programem kwadraty z pomiarem czasu.

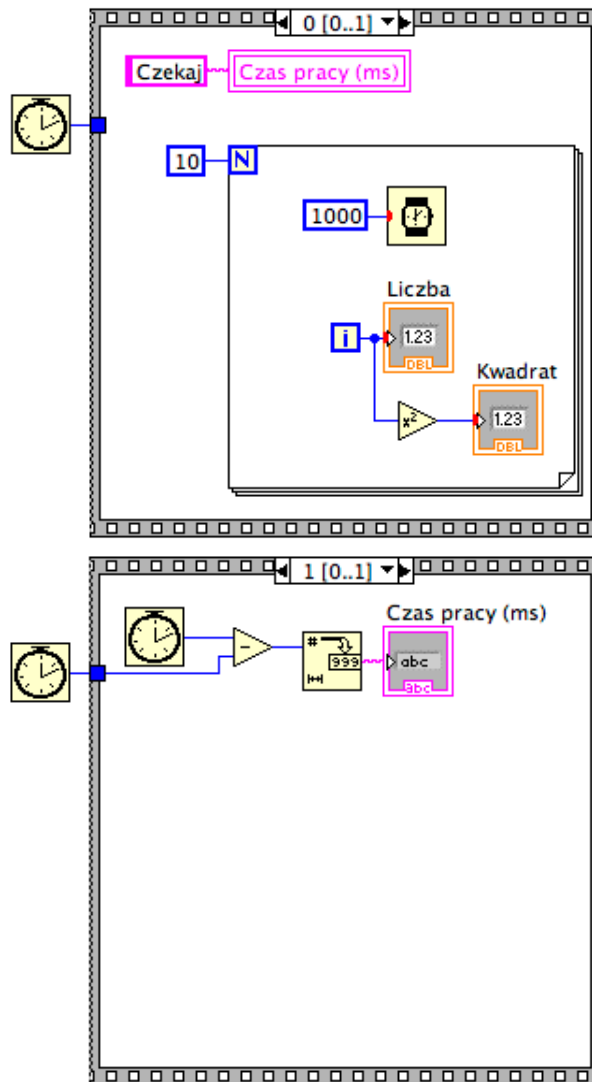


Rysunek 8: Schemat programu kwadraty z pomiarem czasu.

Następnie zmodyfikuj program tak (rys. 9), aby po kolejnym uruchomieniu w polu **Czas pracy** pojawiał się początkowo napis **Czekaj**. W tym celu zmień typ tego pola na tekstowy.

W schemacie programu skorzystaj z obiektów

- zmiennej lokalnej (Programming → Structures → LocalVariable),
- obiektu umożliwiającego konwersję liczby do napisu (Programming → String → String/Number Conversion → Number to Decimal String).



Rysunek 9: Schemat programu kwadraty z pomiarem czasu ze zmienną lokalną.

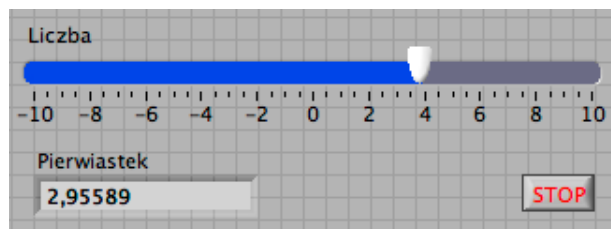
Zad. 5. Korzystając z programu pierwiastek z warunkiem napisz program pierwiastek w pętli (rys. 10 i 11), w którym na bieżąco obliczaj wartość wyrażenia $1 + \sqrt{x}$.

W panelu sterującym skorzystaj z obiektów

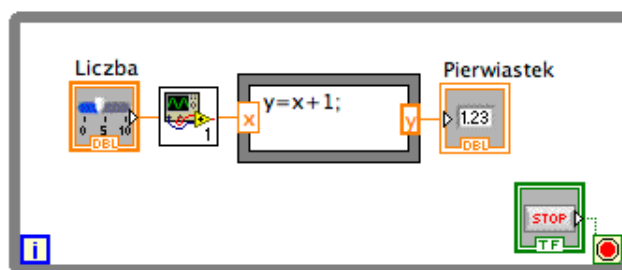
- suwaka zadającego wartość x w zakresie od -10 do 10 (Express → Numeric Controls → Horizontal Pointer Slide),
- przycisku do zatrzymania pracy programu (Express → Buttons & Switches → Stop Button).

W schemacie programu skorzystaj z obiektów

- pętli while (Programming → Structures → While Loop),
- ikony reprezentującej program pierwiastek z warunkiem.



Rysunek 10: Wygląd panelu sterującego programem pierwiastek w pętli.



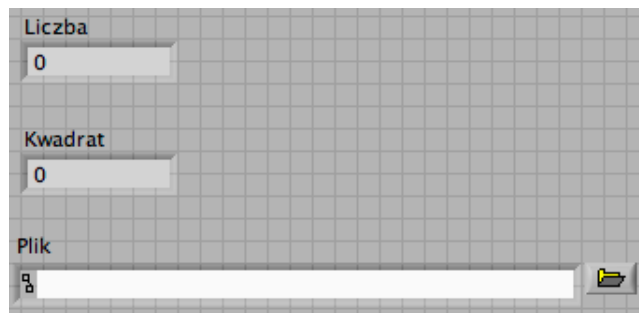
Rysunek 11: Schemat programu pierwiastek w pętli.

Zad. 6. Na podstawie programu kwadraty napisz program kwadraty z zapisem do pliku (rys. 12 i 13) zapisujący wyniki obliczeń do pliku tekstowego. W kolejnych wierszach zapisuj liczbę i jej kwadrat oddzielone spacją.

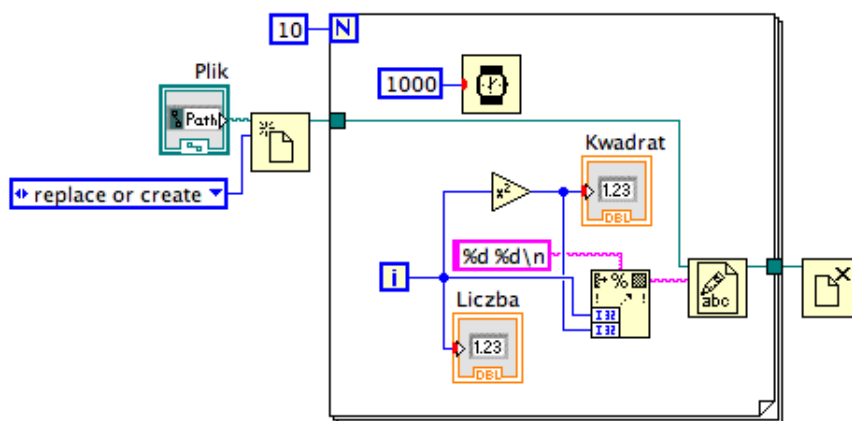
W schemacie programu skorzystaj z

- 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.

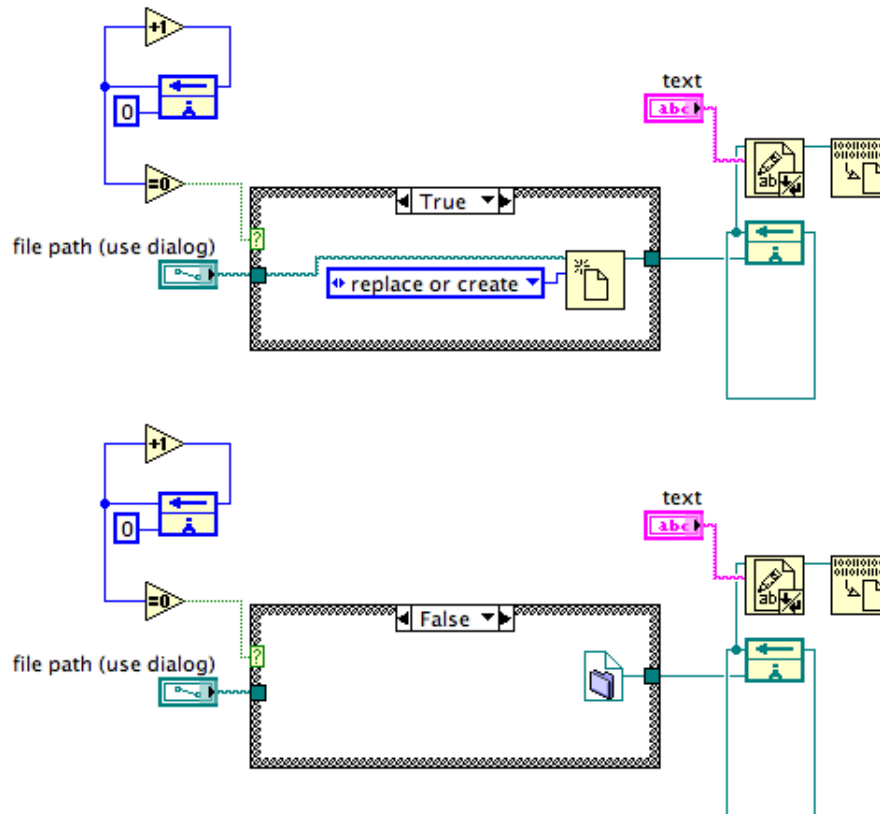


Rysunek 12: Wygląd panelu sterującego programu kwadraty z zapisem do pliku.

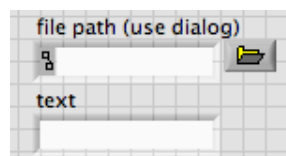


Rysunek 13: Schemat programu kwadraty z zapisem do pliku.

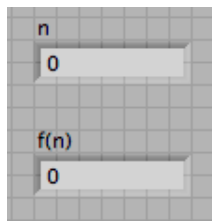
Operację zapisu do pliku wygodnie jest zrealizować w postaci podprogramu `pisz`.



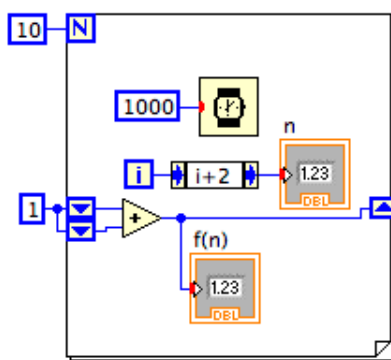
Rysunek 14: Schemat podprogramu `pisz`.



Rysunek 15: Wygląd panelu sterującego podprogramu `pisz`.



Rysunek 18: Wygląd panelu sterującego programu fibonacci.



Rysunek 19: Schemat programu fibonacci.

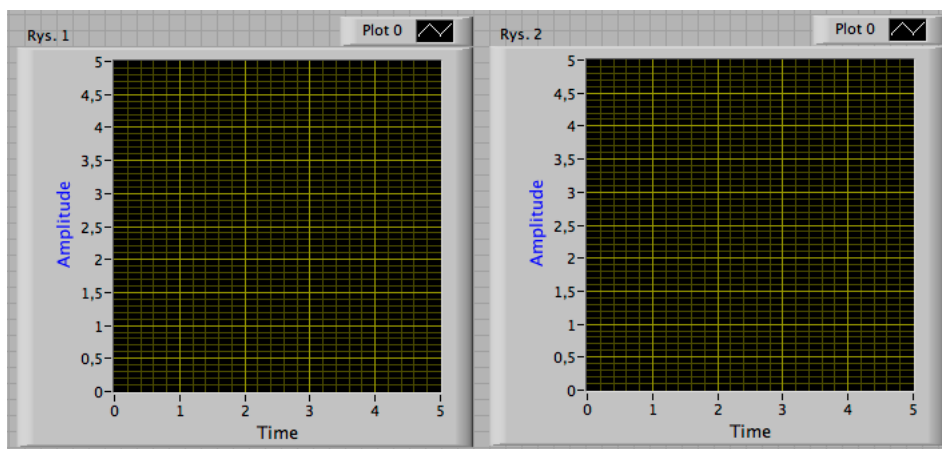
Zad. 8. Napisz program **wykres1** (rys. 20 i 21) rysujący wykres funkcji $y(x) = 5(1 - e^{-x})$ dla $x \in \{0, 0.1, 0.2, \dots, 4.8, 4.9\}$.

W panelu sterującym skorzystaj z obiektu

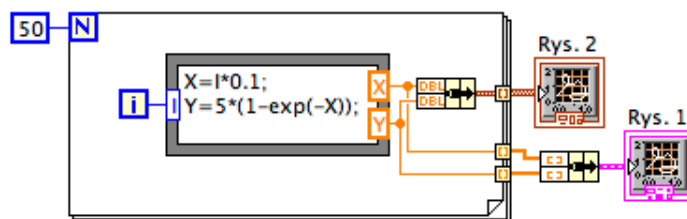
- wykresu funkcji (Modern $\rightarrow$ Graph $\rightarrow$ XY Graph).

W schemacie programu skorzystaj z obiektu

- tworzącego strukturę reprezentującą punkt (Programming $\rightarrow$ Cluster, Class & Variant $\rightarrow$ Bundle).

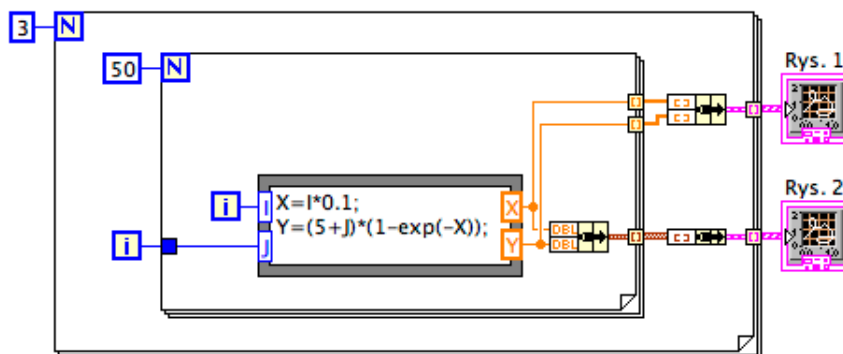


Rysunek 20: Wygląd panelu sterującego programu wykres1.



Rysunek 21: Schemat programu wykres1.

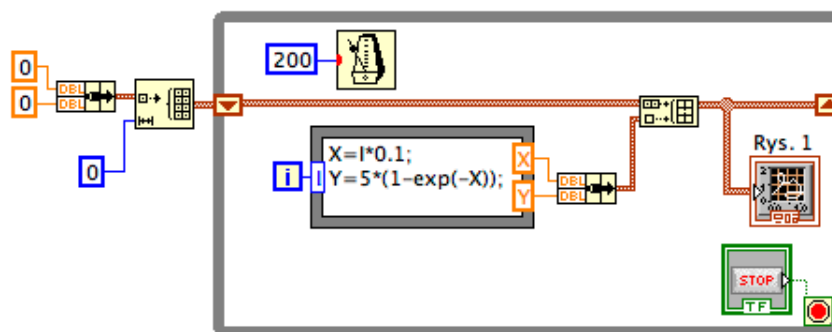
Następnie rozbuduj program (rys. 22) w taki sposób aby jednocześnie rysowane były 3 funkcje $y(x) = (5 + \alpha)(1 - e^{-x})$ dla $\alpha \in \{0, 1, 2\}$.



Rysunek 22: Schemat programu wykres3.

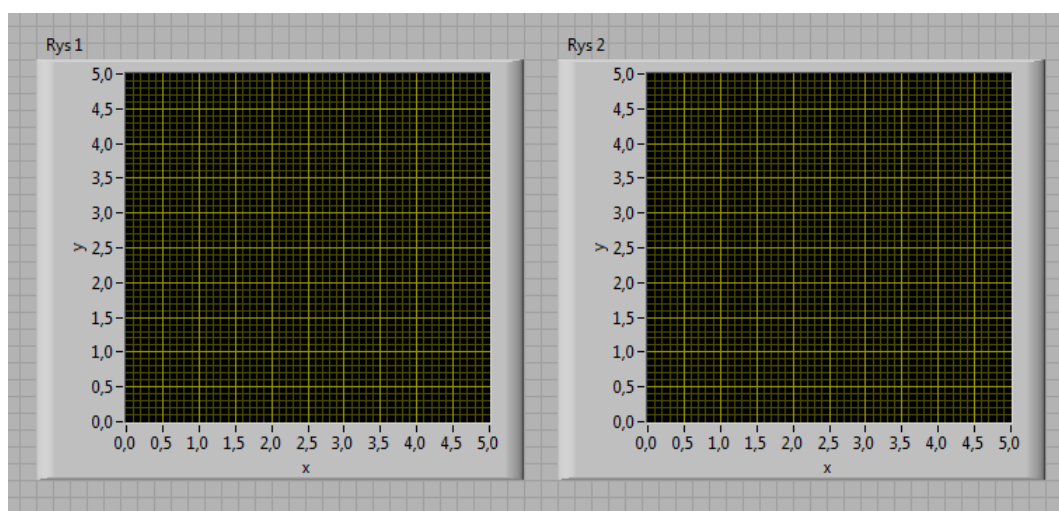
Zmodyfikuj program **wykres1** w taki sposób (rys. 23) aby rysowanie funkcji odbywało się na równoległe ze „zbieraniem” danych. W tym celu pomiędzy obliczeniami wartości $y(x)$ dla kolejnych x_i wprowadź opóźnienie o długości 0,2 sekundy. W schemacie programu skorzystaj z obiektów

- wstrzymującego pracę programu (Programming → Timing → Wait Until Next ms Multiple),
- tworzącego tablicę (Programming → Array → Initialize Array),
- dopisującego element na końcu tablicy (Programming → Array → Build Array).

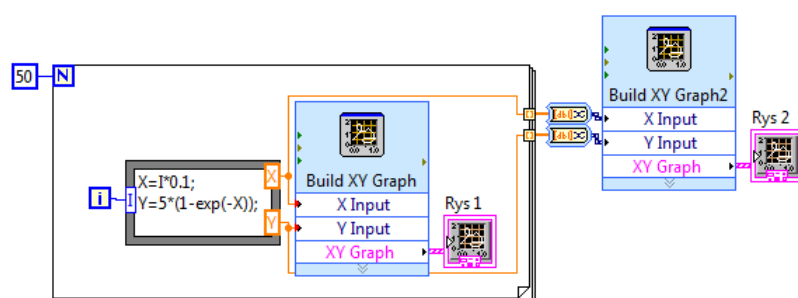


Rysunek 23: Schemat programu **wykres1** na bieżąco.

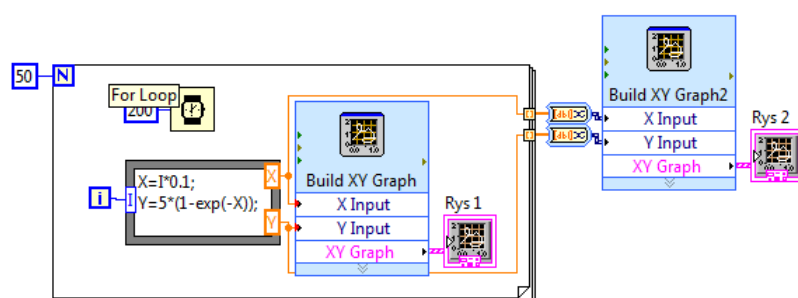
Zad. 9. Zbuduj programy z poprzedniego zadania przy użyciu obiektów z palety Express (rys. 24,25,26, 27). W panelu sterującym użyj obiektów: Express → GraphIndicators → XY Graph. W schemacie programu, we własnościach (Properties) obiektu Build XV Graph eksperymentuj ze zmianą ustawienia: Clear Data on Each Call.



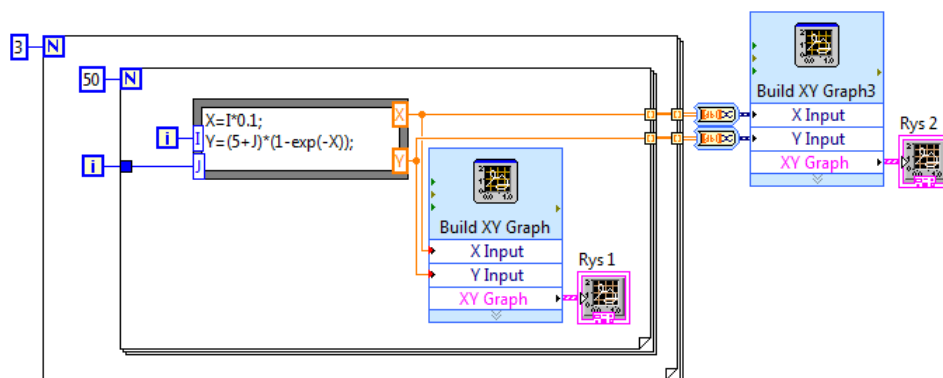
Rysunek 24: Wygląd panelu sterującego programu wykres1_express.



Rysunek 25: Schemat programu wykres1_express.



Rysunek 26: Schemat programu wykres1_na_bieząco_express.



Rysunek 27: Schemat programu wykres3.express.

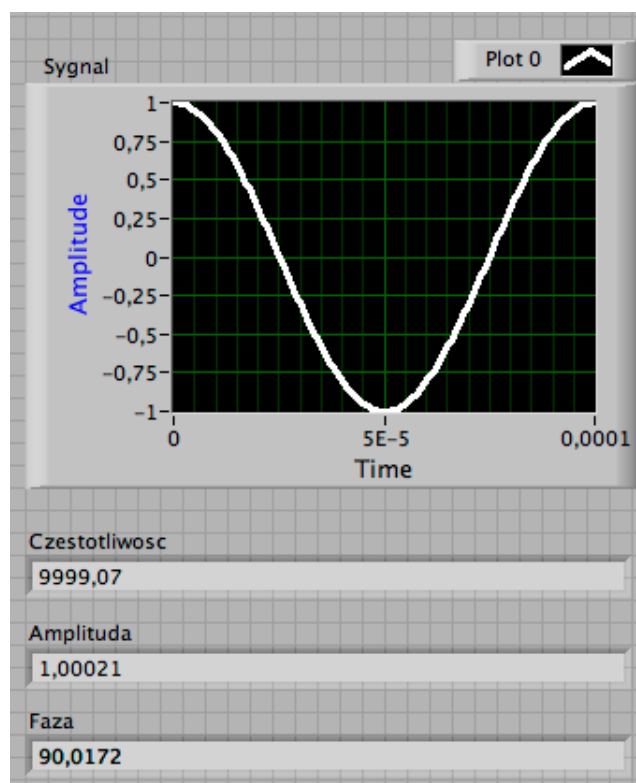
Zad. 10. Napisz program analiza sygnału (rys. 28 i 29), w którym przygotuj sygnał (ang. waveform) zawierający jeden okres funkcji $\cos x$ o częstotliwości 10 kHz. Wartości sygnału oblicz w odstępie $\Delta t = 1\mu s$. Następnie narysuj sygnał na wykresie i oblicz jego częstotliwość, amplitudę i fazę.

W panelu sterującym skorzystaj z obiektu

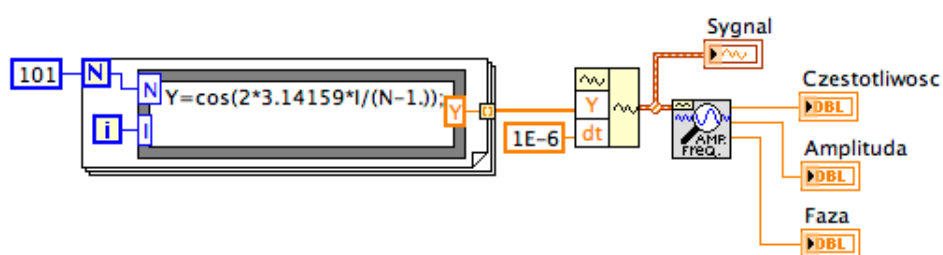
- wykresu sygnału (Modern $\rightarrow$ Graph $\rightarrow$ Waveform Graph).

W schemacie programu skorzystaj z obiektów

- utworzenia sygnału (Programming $\rightarrow$ Waveform $\rightarrow$ Build Waveform),
- analizy sygnału (Signal Processing $\rightarrow$ Waveform Measurements $\rightarrow$ Extract Single Tone Information i dalej w menu kontekstowym obiektu Select Type $\rightarrow$ Extract Single Tone Information 1 Char).



Rysunek 28: Wygląd panelu sterującego programu analiza sygnału.



Rysunek 29: Schemat programu analiza sygnału.

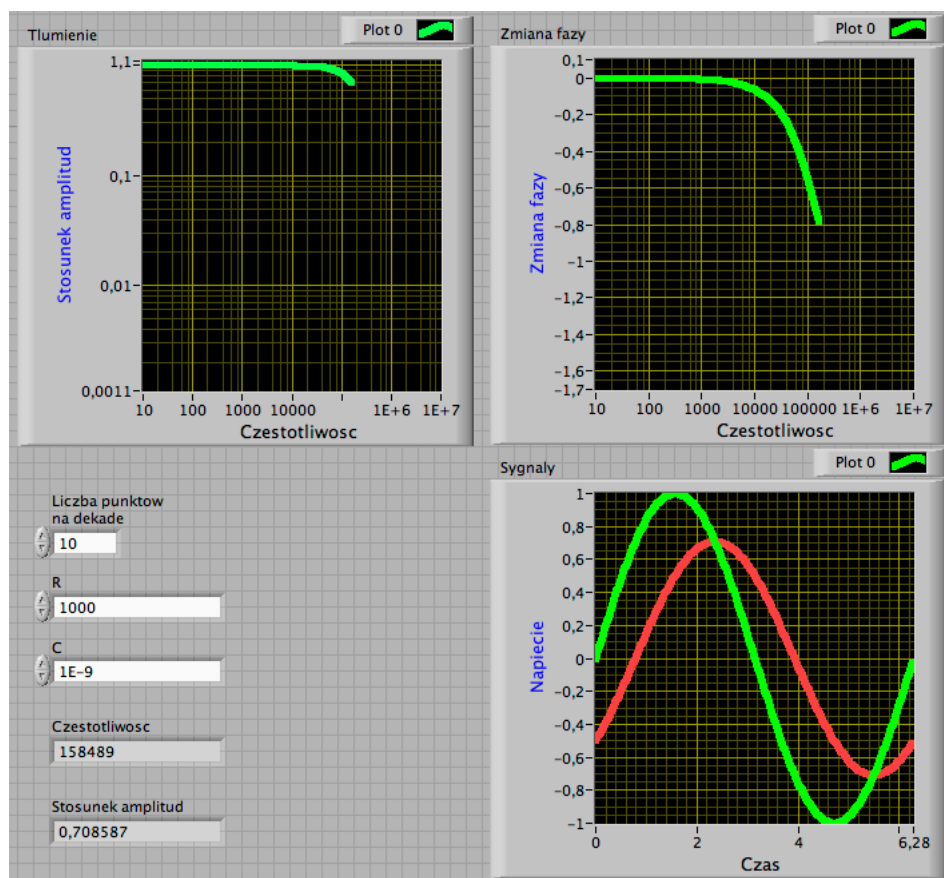
Zad. 11. Znając teoretyczne wartości tłumienia

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

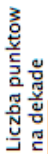
i przesunięcia fazy

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

sygnału wyjściowego w filtrze dolnoprzepustowym RC napisz program teoria RC (rys. 30 i 31) będącego komputerową symulacją doświadczenia pomiaru charakterystyki.



Rysunek 30: Wygląd panelu sterującego programu teoria RC.

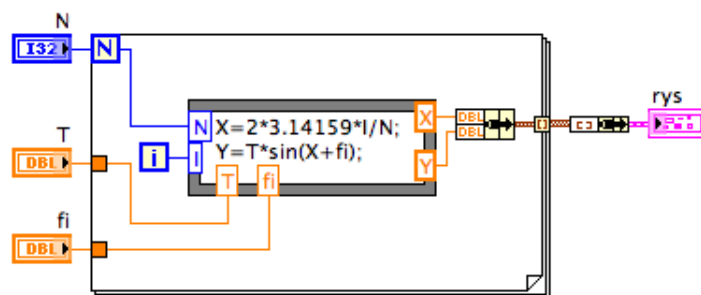


Rysunek 31: Schemat programu teoria RC.

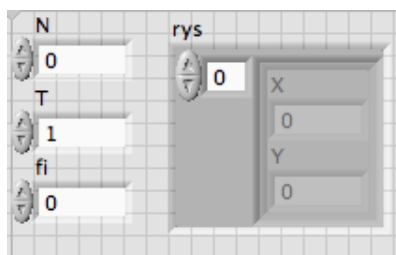
W celu poprawienia czytelności warto zapisać program z wykorzystaniem podprogramów. Przygotowanie danych wykresu funkcji

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

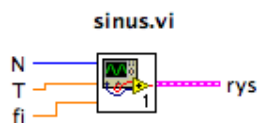
zapisujemy w postaci podprogramu **sinus** (rys. 32, 33, 34).



Rysunek 32: Schemat podprogramu **sinus**.

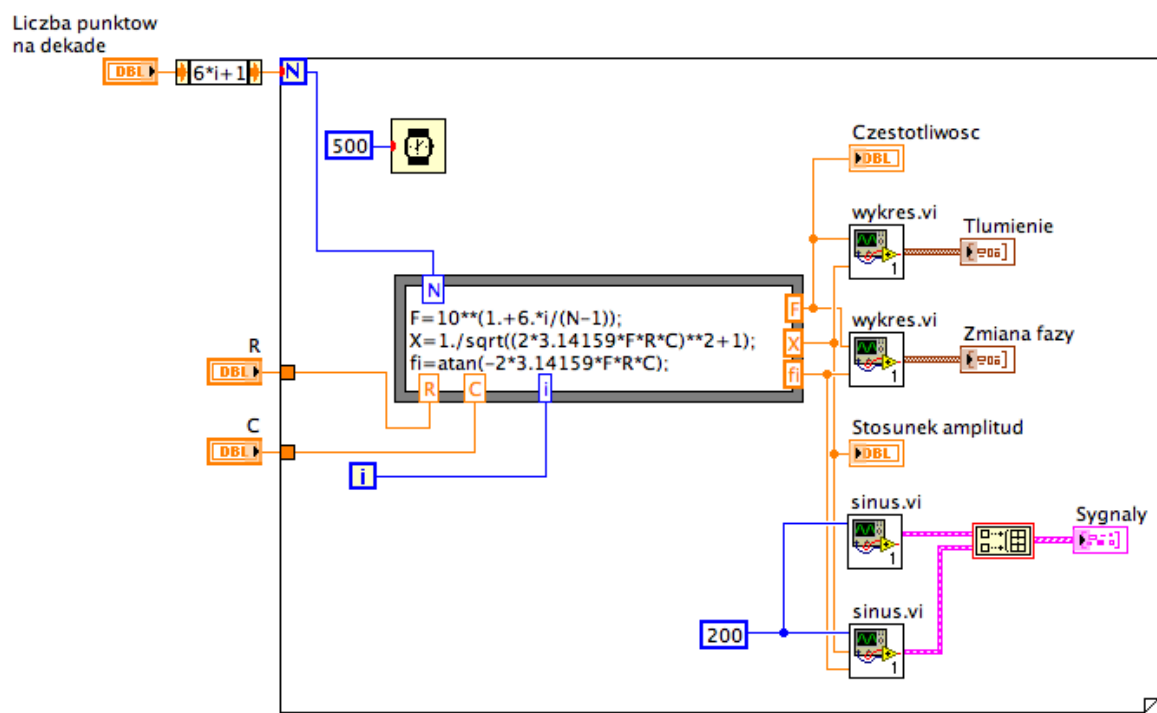


Rysunek 33: Wygląd panelu sterującego podprogramu **sinus**.



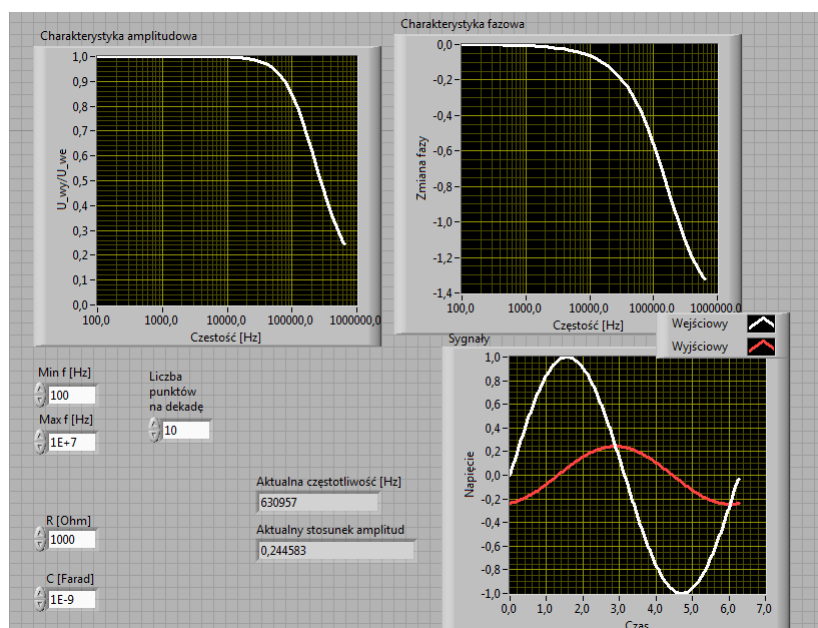
Rysunek 34: Wygląd ikony podprogramu **sinus**.

Przygotowanie danych dla wykresu tworzonego równolegle z obliczeniami zapisujemy w postaci podprogramu **wykres** (rys. 35, 36, 37). Ponieważ będziemy używali dwóch instancji podprogramu jednocześnie należy zaznaczyć opcję VI Properties → Execution → Reentrant Execution.

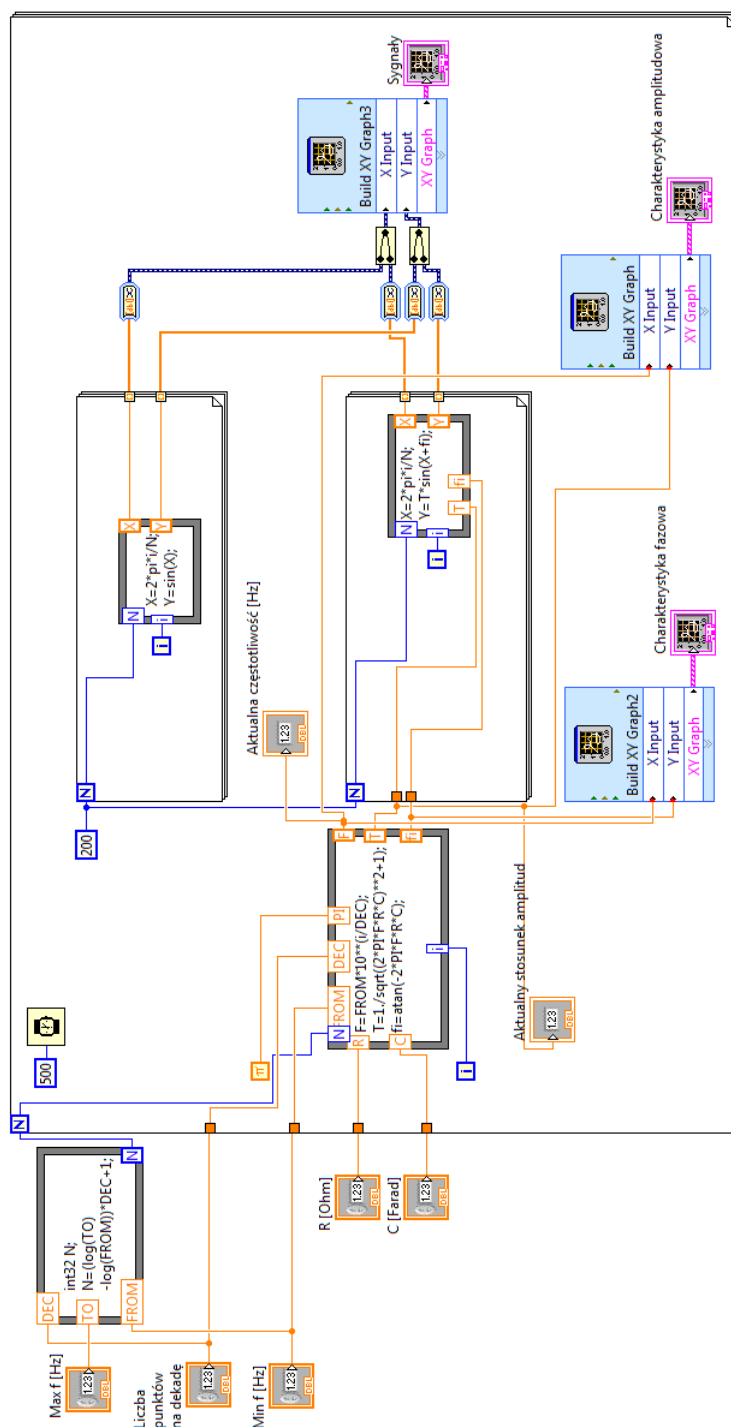


Rysunek 38: Schemat programu teoria RC zapisanego z wykorzystaniem pod-programów sinus i wykres.

Druga wersja programu, przy użyciu obiektów z palety Express (rys. 39 i 40).



Rysunek 39: Wygląd panelu sterującego programu teoria_RC_express.

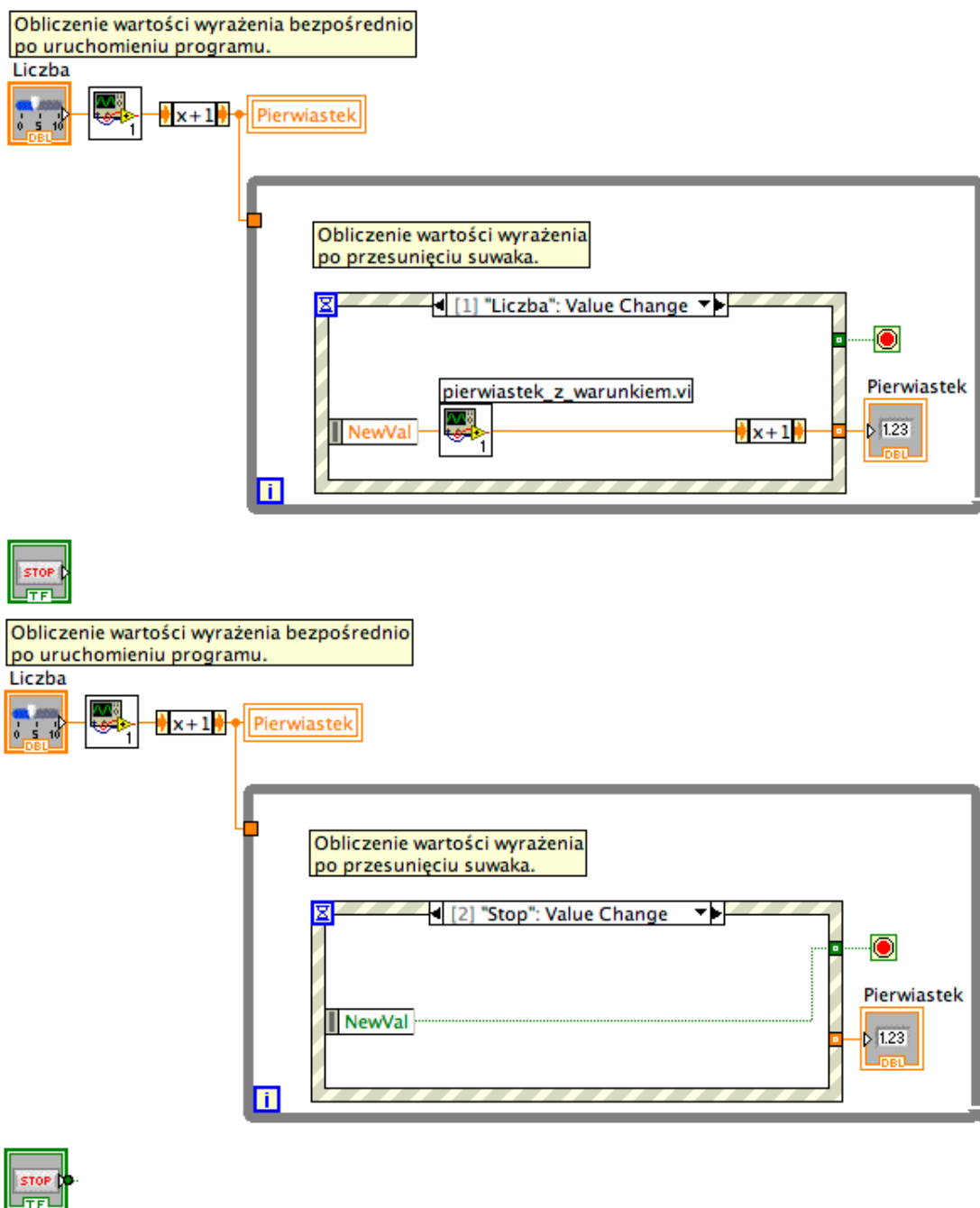


Rysunek 40: Schemat programu teoria_RC_express.

Zad. 12. W programie *pierwiastek* w pętli wykorzystaj mechanizm obsługi zdarzeń (rys. 41). Skorzystaj ze zdarzenia *zmiana wartości* (ang. *value changed*) związanego z suwakiem *Liczba*. Zadbaj aby bezpośrednio po uruchomieniu programu nastąpiło obliczenie wartości $1 + \sqrt{x}$ dla początkowego ustawienia suwaka.

W schemacie programu skorzystaj z obiektu

- obsługi zdarzeń (*Programming* → *Structures* → *Event Structure*).



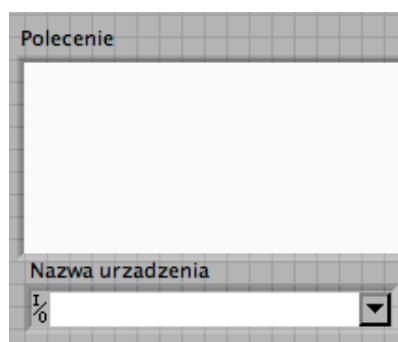
Rysunek 41: Schemat programu pierwiastek w pętli wykorzystującego obsługę zdarzeń.

Zad. 13. Napisz programy polecenie (rys. 42 i 43) i zapytanie (rys. 44 i 45)

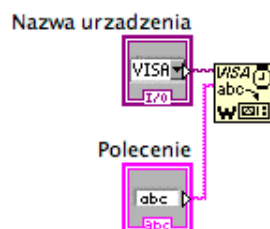
przesyłające polecenie (odpowiednio zapytanie) do urządzenia pomiarowego i (w przypadku zapytania) odczytujące odesłaną odpowiedź.

W schemacie programu skorzystaj z obiektów

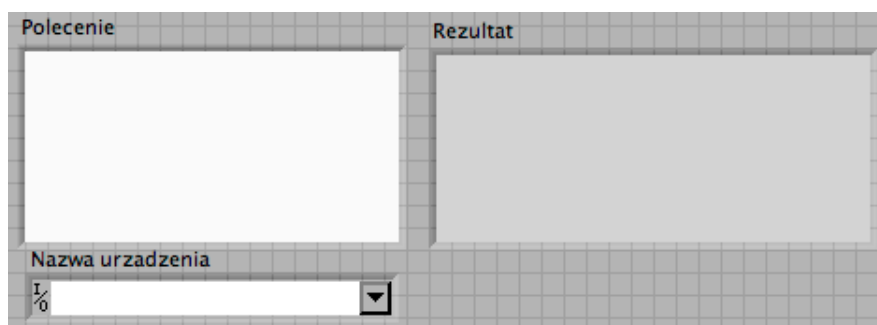
- przesłania polecenia do urządzenia (Instrument I/O → VISA → VISA Write),
- odczytania odpowiedzi z urządzenia (Instrument I/O → VISA → VISA Read).



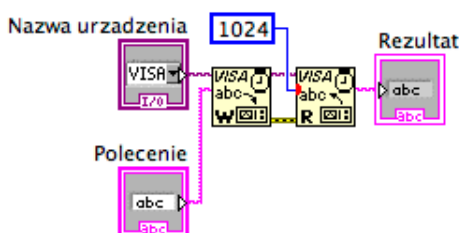
Rysunek 42: Wygląd panelu sterującego programem polecenie.



Rysunek 43: Schemat programu polecenie.



Rysunek 44: Wygląd panelu sterującego programu zapytanie.



Rysunek 45: Schemat programu zapytanie.

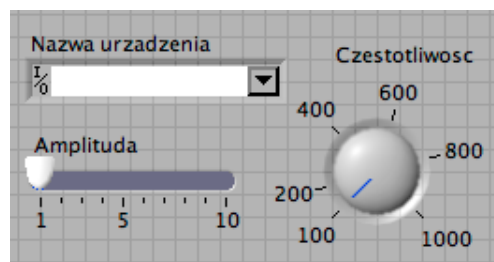
Zad. 14. Bez wykorzystywania mechanizmu obsługi zdarzeń napisz program generator (rys. 46 i 47) sterujący pracą generatora dostępnego na pracowni i umożliwiający ustawienie amplitudy sygnału w zakresie $1 \dots 10$ V oraz częstotliwości w zakresie $100 \dots 1000$ Hz. Podłączając sygnał generatora do oscyloskopu sprawdź użyteczność programu.

W panelu sterującym skorzystaj z obiektu

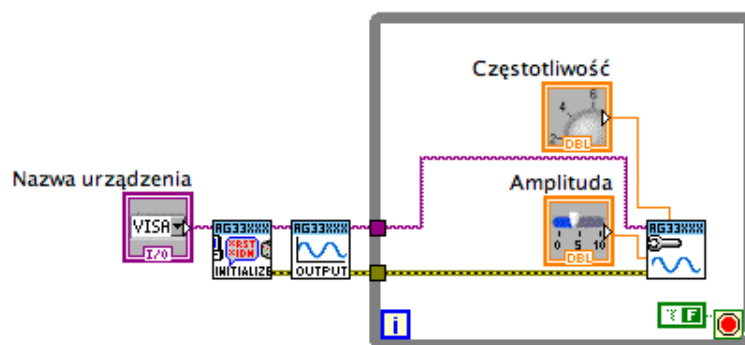
- reprezentującego liczbę (Express → Numeric Controls → Dial).

W schemacie programu skorzystaj z obiektów sterownika generatora (Instrument I/O → Instrument Drivers → Agilent 33XXX Series)

- przygotowującego generator do pracy (Initialize),
- załączającego sygnał wyjściowy (Action–Status → Enable Output),
- ustalającego parametry generowanego sygnału (Configure → Configure Standard Waveform).

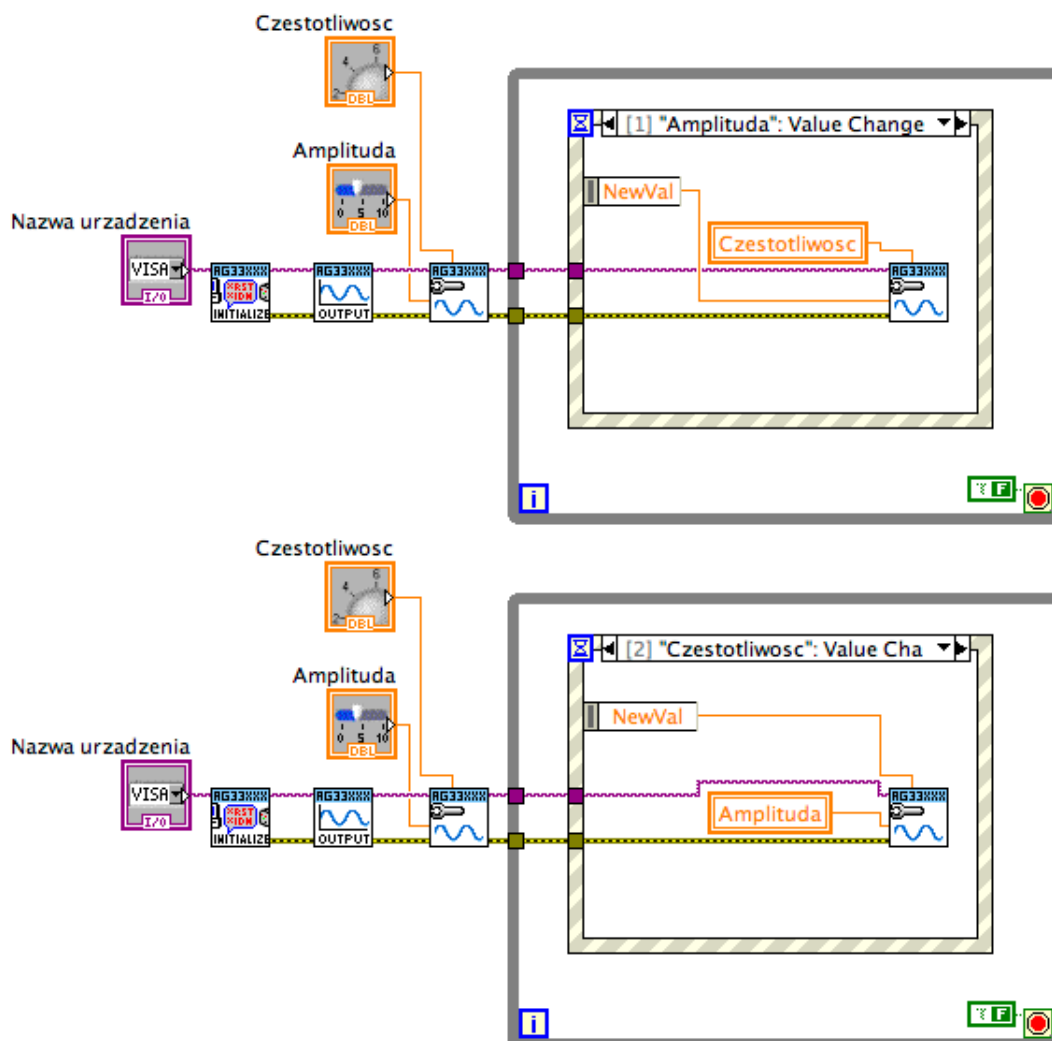


Rysunek 46: Wygląd panelu sterującego programu generator.



Rysunek 47: Schemat programu generator.

Następnie zmodyfikuj program (rys. 48) wprowadzając obsługę zdarzeń *zmiana wartości* (ang. *value changed*) związanych z suwakiem *Amplituda* i pokrętkiem *Częstotliwość*. Porównaj efektywność sterowania generatorem z poprzednią wersją programu.

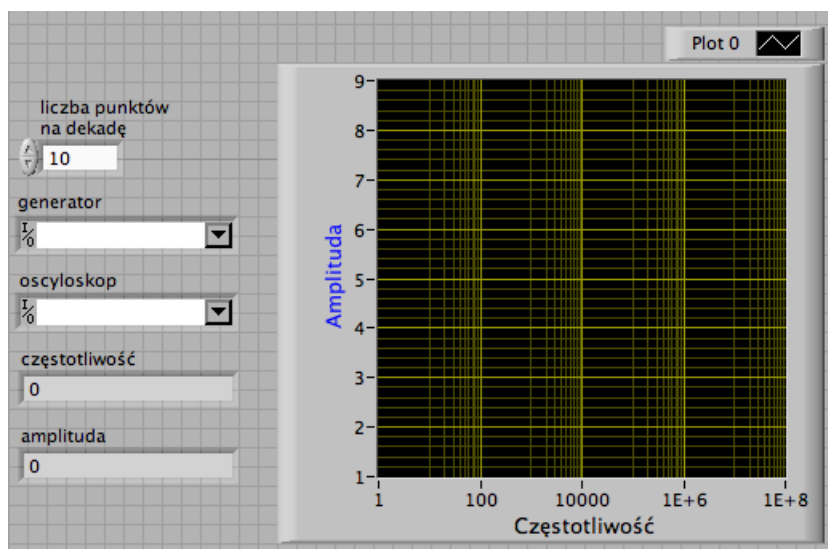


Rysunek 48: Schemat programu generator wykorzystującego obsługę zdarzeń.

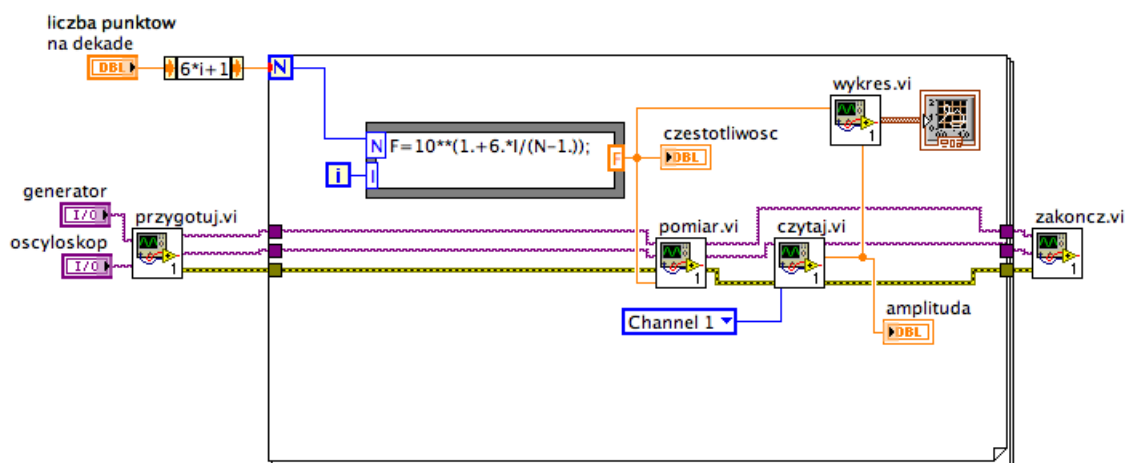
Zad. 15. Napisz program pomiar amplitudy (rys. 49 i 50) realizujący pomiar amplitudy sygnału generatora podłączonego do kanału 1 oscyloskopu w zakresie częstotliwości od 10 Hz do 10 MHz.

- Czynności związane z przygotowaniem urządzeń pomiarowych zapisz w postaci podprogramu **przygotuj**.
- Czynności związane z wykonaniem pojedynczego pomiaru amplitudy i fazy sygnału o częstotliwości f zapisz w postaci podprogramu **pomiar**.

- Czynności związane ze zwolnieniem urządzeń pomiarowych zapisz w postaci podprogramu zakończ.

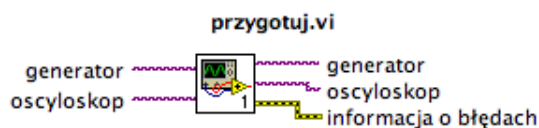


Rysunek 49: Wygląd panelu sterującego programu pomiar amplitudy.



Rysunek 50: Schemat programu pomiar amplitudy.

Podprogram przygotuj

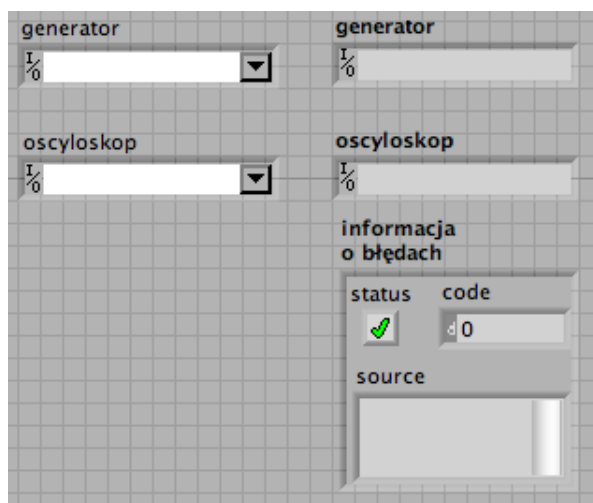


Rysunek 51 przedstawia wygląd panelu czołowego podprogramu. Podprogram wymaga dwóch danych wejściowych

1. identyfikatora generatora,
2. identyfikatora oscyloskopu.

Po zakończeniu pracy udostępnia trzy rezultaty

1. identyfikator generatora,
2. identyfikator oscyloskopu,
3. informację o błędach wykonania podprogramu.



Rysunek 51: Wygląd panelu sterującego podprogramu **przygotuj**. Wejścia sygnałów zostały umieszczone po lewej stronie panelu a wyjścia po prawej stronie.

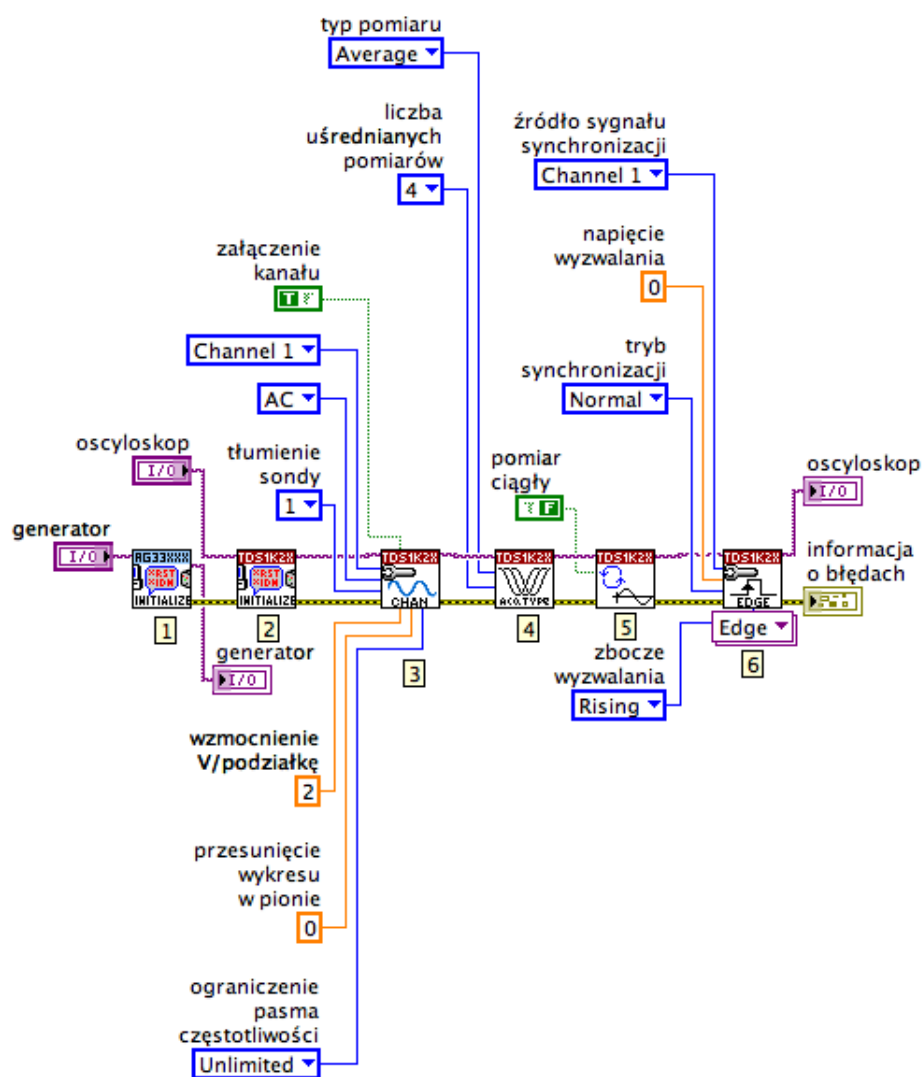
Rysunek 52 przedstawia schemat podprogramu, na który składają się następujące czynności.

1. Inicjalizacja generatora (Initialize¹).

¹Instrument I/O → Instrument Drivers → Agilent 33XXX Series

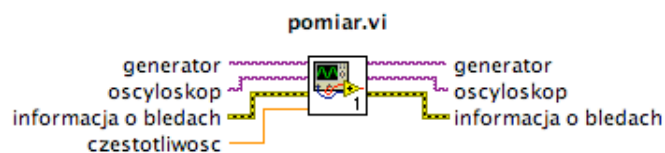
-
2. Inicjalizacja oscyloskopu (Initialize²).
 3. Konfiguracja kanału 1 oscyloskopu (Configure → Configure Channel³)
 - (a) pomiar bez składowej stałej AC,
 - (b) tłumienie sondy 1,
 - (c) czułość 2 V/„podziałkę”,
 - (d) przesunięcie w pionie 0,
 - (e) bez ograniczenia pasma częstotliwości mierzonego sygnału.
 4. Konfiguracja sposobu rejestracji: uśrednianie 4 kolejnych pomiarów (Configure → Configure Acquisition Type³).
 5. Konfiguracja pomiaru: pojedynczy pomiar (Configure → Configure Continuous Acquisition³).
 6. Konfiguracja synchronizacji (Configure → Configure Trigger³)
 - (a) źródło sygnału synchronizacji kanał 1,
 - (b) napięcie wyzwalania 0 V,
 - (c) wyzwalanie tylko w obecności sygnału,
 - (d) wyzwalanie narastającym zboczem.

²Instrument I/O → Instrument Drivers → Tektronix TDS 200 1000 2000 Series



Rysunek 52: Schemat podprogramu przygotuj.

Podprogram pomiar

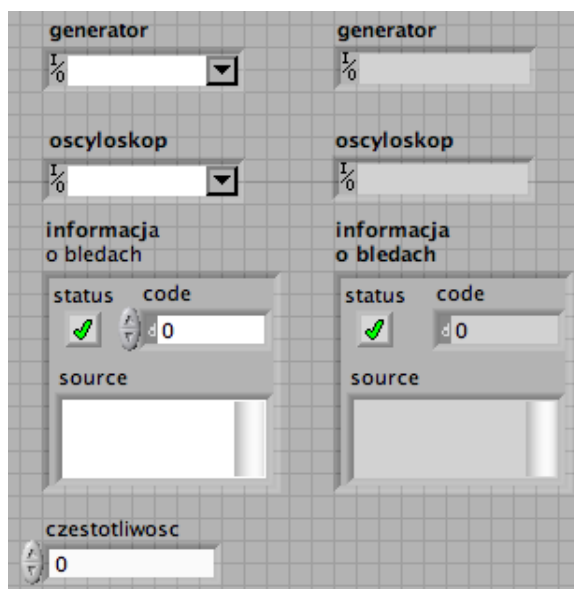


Rysunek 53 przedstawia wygląd panelu czołowego podprogramu. Podprogram wymaga czterech danych wejściowych:

1. identyfikatora generatora,
2. identyfikatora oscyloskopu,
3. informacji błędach wykonania poprzedzających elementów,
4. częstotliwości sygnału.

Po zakończeniu pracy udostępnia trzy rezultaty:

1. identyfikator generatora,
2. identyfikator oscyloskopu,
3. informację o błędach wykonania podprogramu.

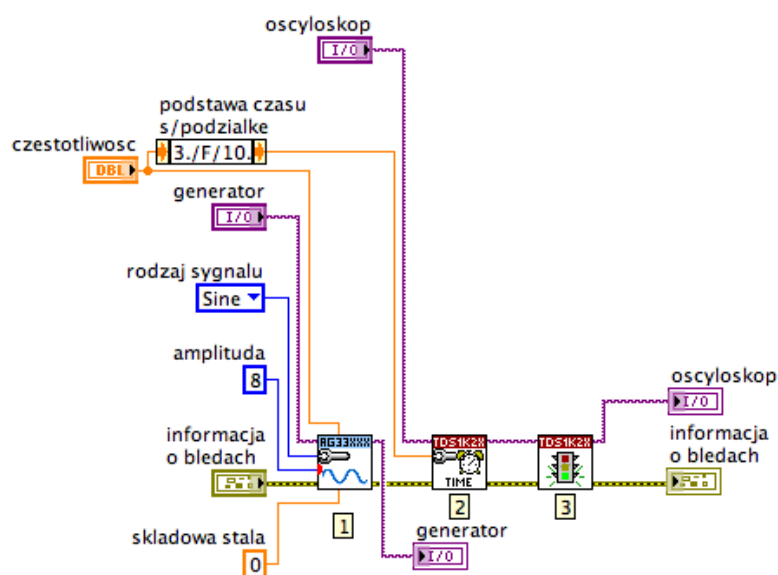


Rysunek 53: Wygląd panelu sterującego podprogramu **pomiar**. Wejścia sygnałów zostały umieszczone po lewej stronie panelu, a wyjścia po prawej stronie.

Rysunek 54 przedstawia schemat podprogramu, na który składają się następujące czynności.

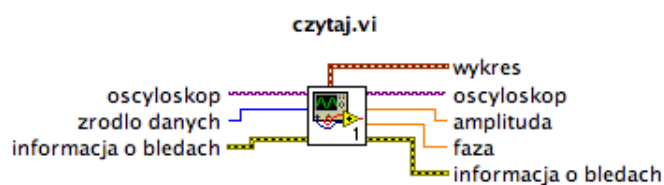
1. Ustawienie parametrów sygnału (Config → Configure Standard Waveform²)
 - (a) przebieg sinusoidalny,

- (b) częstotliwość,
 - (c) amplituda 8 V,
 - (d) składowa stała 0 V.
2. Ustawienie szybkości podstawy czasu oscyloskopu równej $\frac{3}{10f}$ s, tak aby na ekranie oscyloskopu widoczne były mniej więcej 3 pełne przebiegi sygnału (Configure → Configure Timebase³).
 3. Wykonanie pomiaru (Data → Low Level → Initialize³).



Rysunek 54: Schemat podprogramu pomiar.

Podprogram czytaj



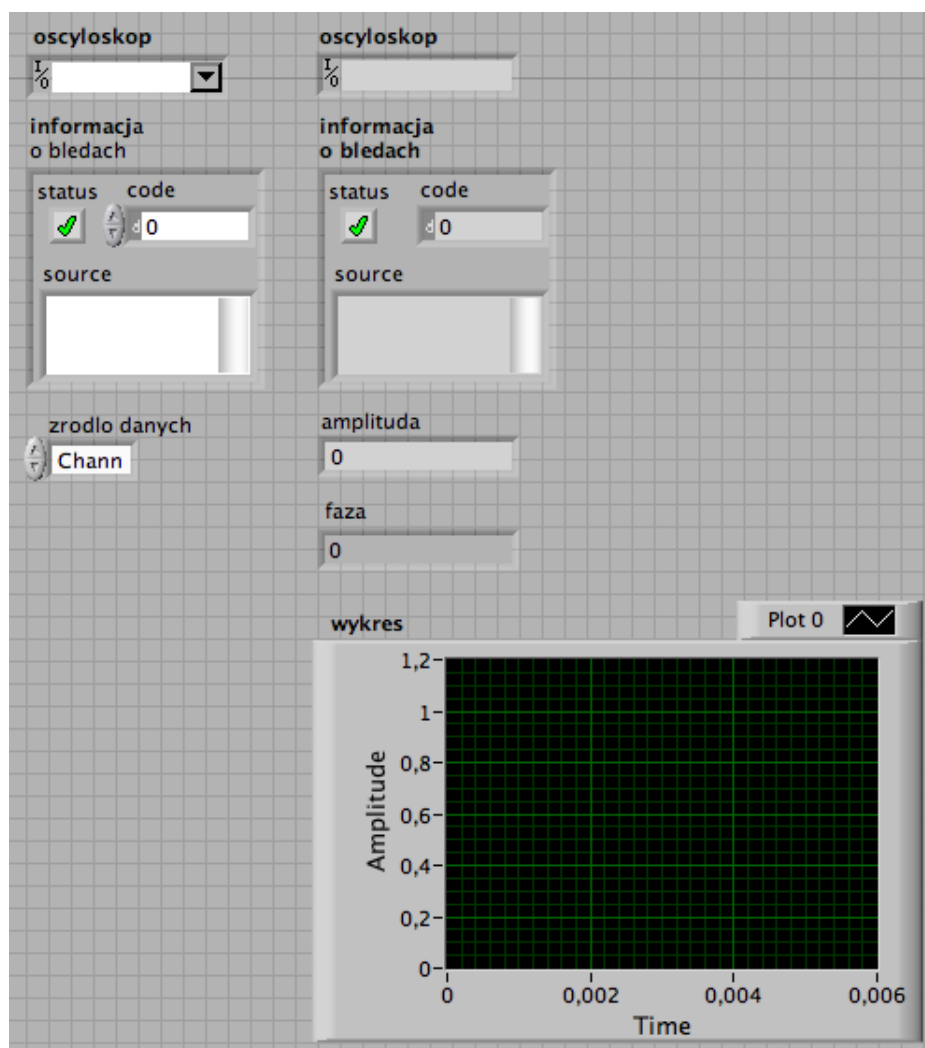
Rysunek 55 przedstawia wygląd panelu czołowego podprogramu. Podprogram wymaga trzech danych wejściowych:

1. identyfikatora oscyloskopu,

2. informacji błędach wykonania poprzedzających elementów,
3. numeru kanału oscyloskopu, z którego zostaną odczytane dane.

Po zakończeniu pracy udostępnia pięć rezultatów:

1. identyfikator oscyloskopu,
2. informację o błędach wykonania podprogramu,
3. zmierzoną amplitudę sygnału,
4. zmierzoną fazę sygnału,
5. wykres sygnału.

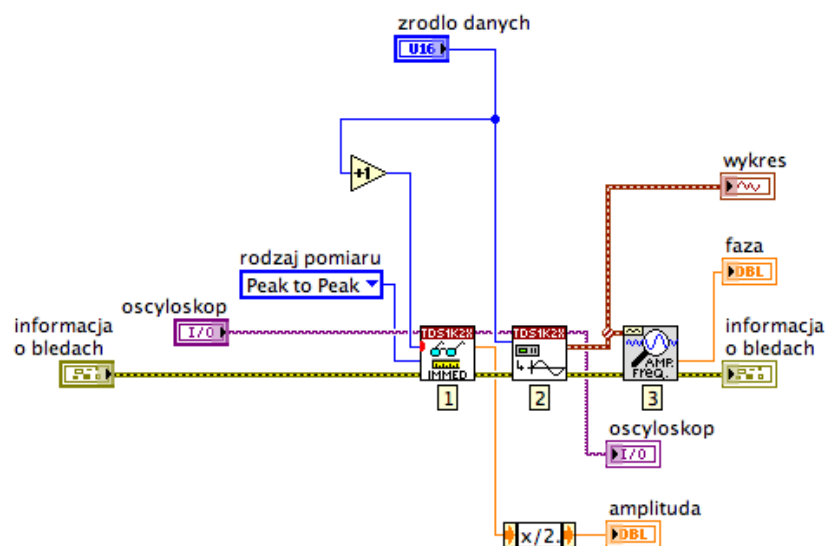


Rysunek 55: Wygląd panelu sterującego podprogramu *czytaj*. Wejścia sygnałów zostały umieszczone po lewej stronie panelu, a wyjścia po prawej stronie.

Rysunek 56 przedstawia schemat podprogramu, na który składają się następujące czynności.

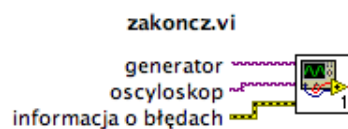
1. Odczytanie napięcia międzyszczytowego zarejestrowanego sygnału (Data → Read Measurement³).
2. Odczytanie zarejestrowanego przebiegu (Data → Low level → Fetch Waveform³).
3. Obliczenie fazy (Signal processing → Waveform Measurements → Extract

Single Tone Information).



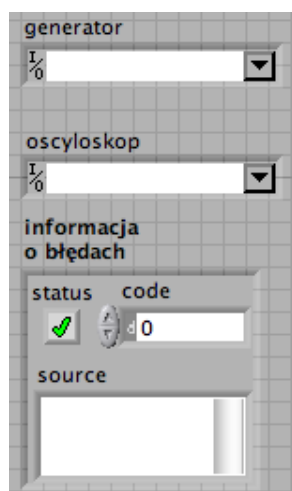
Rysunek 56: Schemat podprogramu czytają.

Podprogram zakończ



Rysunek 57 przedstawia wygląd panelu czołowego podprogramu. Podprogram wymaga trzech danych wejściowych:

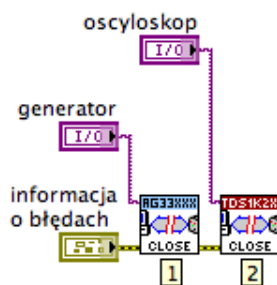
1. identyfikatora generatora,
2. identyfikatora oscyloskopu,
3. informacji o błędach wykonania poprzedzających elementów.



Rysunek 57: Wygląd panelu sterującego podprogramu zakończ.

Rysunek 58 przedstawia schemat podprogramu, na który składają się następujące czynności.

1. Zwolnienie zasobów związanych z generatorem (Close^2).
2. Zwolnienie zasobów związanych z oscyloskopem (Close^3).



Rysunek 58: Schemat podprogramu zakończ.

Rozwiązania zadań

Zad. 1. Powrót do treści zadania

- `pierwiastek_i_kwadrat_wersja1.vi`
- `pierwiastek_i_kwadrat_wersja2.vi`
- `pierwiastek_i_kwadrat_wersja3.vi`

Zad. 2. Powrót do treści zadania

- `kwadraty.vi`

Zad. 3. Powrót do treści zadania

- `pierwiastek_z_warunkiem.vi`

Zad. 4. Powrót do treści zadania

- `kwadraty_z_pomiarem_czasu.vi`
- `kwadraty_z_pomiarem_czasu_ze_zmienna_lokalna.vi`

Zad. 5. Powrót do treści zadania

- `pierwiastek_w_petli.vi`

Zad. 6. Powrót do treści zadania

- `kwadraty_z_zapisem_do_pliku.vi`
- `kwadraty_z_zapisem_do_pliku/kwadraty_z_zapisem_do_pliku.vi`
- `kwadraty_z_zapisem_do_pliku/pisz.vi`

Zad. 7. Powrót do treści zadania

- `fibonacci.vi`

Zad. 8. Powrót do treści zadania

- wykres1.vi
- wykres3.vi
- wykres1_na_biezaco.vi

Zad. 9. Powrót do treści zadania

- wykres1_express.vi
- wykres3_express.vi
- wykres1_na_biezaco_express.vi

Zad. 10. Powrót do treści zadania

- analiza_sygnalu.vi

Zad. 11. Powrót do treści zadania

- teoria_rc.vi
- teoria_rc/teoria_rc.vi
- teoria_rc/sinus.vi
- teoria_rc/wykres.vi
- teoria_rc_express.vi

Zad. 12. Powrót do treści zadania

- pierwiastek_w_petli_zdarzenia.vi

Zad. 13. Powrót do treści zadania

- polecenie.vi
- zapytanie.vi

Zad. 14. Powrót do treści zadania

- generator.vi
- generator_zdarzenia.vi

Zad. 15. Powrót do treści zadania

- pomiar_amplitudy.vi
- przygotuj.vi
- pomiar.vi
- czytaj.vi
- zakoncz.vi
- wykres.vi