Laboratorium 1

Wprowadzenie do środowiska GnuRadio

I. Wprowadzenie

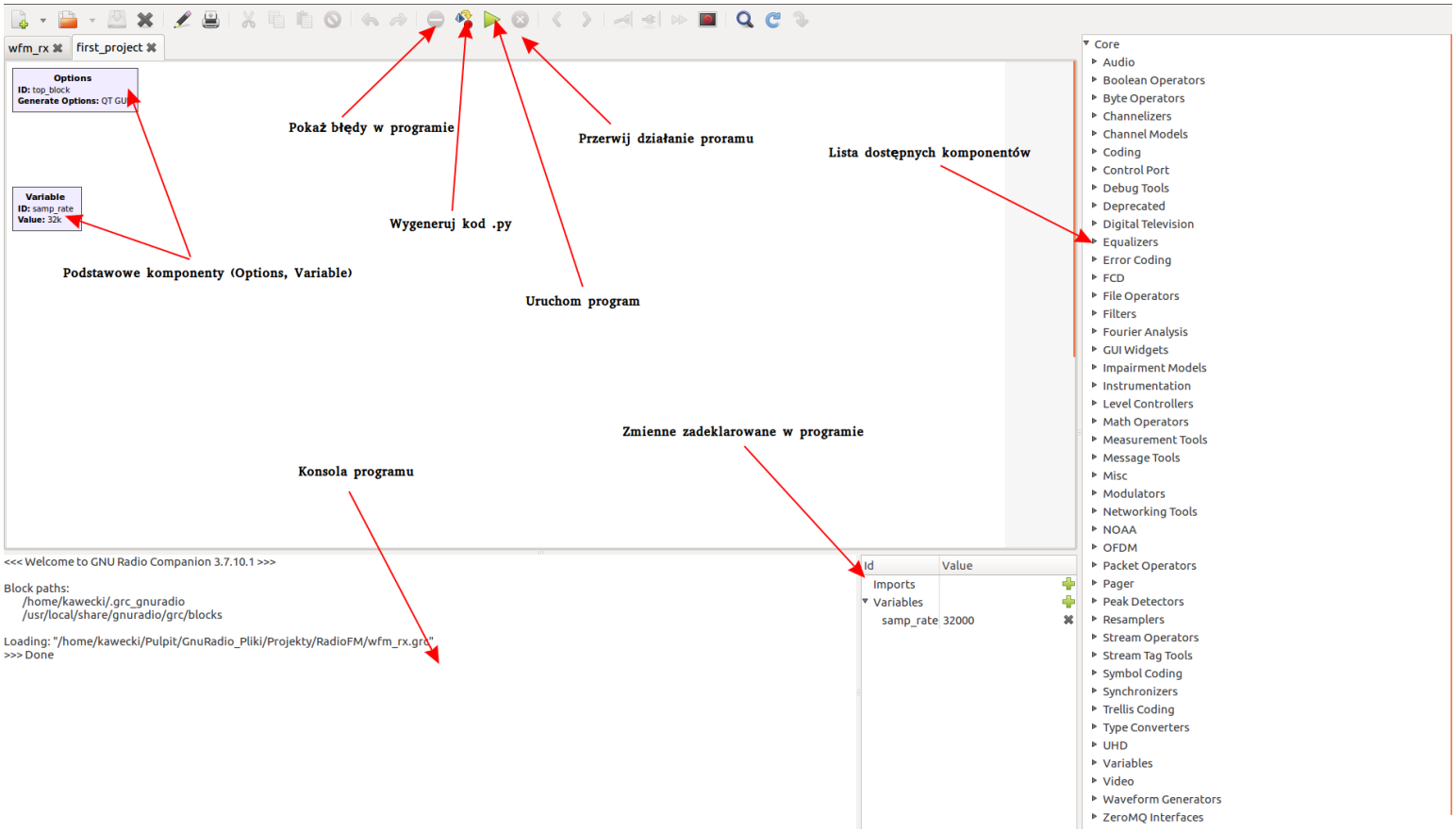
GnuRadio jest darmowym oprogramowaniem wydanym w oparciu o licencję „General Public License". Umożliwia użytkownikowi projektowanie oraz implementację przetwarzanych sygnałów, bez konieczności używania specjalistycznych sprzętów takich jak np. generator sygnałów badź oscyloskop. Sygnały możemy generować za pomocą dostarczonych komponentów programu GnuRadio Companion.

GnuRadio posiada graficzny interfejs użytkownika, który umożliwia na projektowanie oraz modelowanie operacji wymaganych podczas przetwarzania sygnałów. GnuRadio umożliwia wygenerowanie kodu w języku Python, który uważany jest za wszechobecny w użyciu na świecie. Gnu Radio stanowi alternatywę dla programów takich jak Matlab oraz LabView. Oprogramowanie to jest on bardzo intuicyjne oraz pomocne podczas zdobywania wiedzy z zakresu tematyki przetwarzania sygnałów analogowych oraz cyfrowych. Oprogramowanie jest typu „open source" więc bez ograniczeń licencyjnych można z niego korzystać jak również wzbogacać go w nowe komponenty.

Plik projektu, który stworzymy w programie Gnu Radio Companion zapisywany jest z rozszerzeniem .grc. Możemy wygenerować również kod w języku Python. Należy jedynie pamiętać, że zapisany projekt przenoszony jest na inne urządzenia w pliku z rozszerzeniem .grc.

II. Pierwszy projekt

Po uruchomieniu programu „GRC" utworzony zostanie nowy projekt wraz z dwoma podstawowymi komponentami. Pierwszy z nich to opcje projektu, a kolejny to zmienna zadeklarowana jako częstotliwość próbkowania systemu.
Z prawej strony znajduje się lista komponentów środowiska GRC. Aby umieścić komponent należy go przeciągnąć z listy na canvas. Poniżej zaprezentowano główny widok programu, który dostępny jest po utworzeniu nowego projektu. Wyjaśnione jest również znaczenie podstawowych elementów niezbędnych do wykorzystania podczas pracy nad pierwszym ćwiczeniem.



Zadanie

Celem ćwiczenia jest zaprojektowanie i implementacja iloczynu sygnału sinusoidalnego i cosinusoidalnego oraz wyświetlenie jego przebiegu na wykresie. Użytkownik musi mieć możliwość sterowania amplitudą danego sygnału.

1. Uruchom program „GRC”.

2. Przejdź do właściwości bloku „Options” poprzez dwukrotne kliknięcie. Podaj ID swojego programu oraz wybierz typ graficznego interfejsu użytkownika (np. WX GUI).

3. Z listy komponentów przesuń na canvas dwa źródła sygnału „Signal source”. Pierwszym z nich jest źródło o przebiegu cosinusoidalnym, częstotliwości 2000Hz oraz amplitudzie 1V. W polu częstotliwość próbkowania podaj nazwę zadeklarowanej zmiennej (samp_rate).

Drugim źródłem sygnału jest sinus o częstotliwości 1000Hz oraz amplitudzie 1V. W polu częstotliwość próbkowania podaj nazwę zadeklarowanej zmiennej (samp_rate).

Uwaga! Pamiętaj aby ustawić odpowiednie typach danych wejściowych oraz wyjściowych w komponentach. Dostępne typy danych to: Complex, Float, Int, Short, Byte. Użyj typu float.

4. Dodaj operator matematyczny „Multiply” oraz wykonaj iloczyn obu sygnałów. Na wyjściu powinien zostać wygenerowany iloczyn obu sygnałów. Komponenty łączymy ze sobą za pomocą pojedynczego kliknięcia w miejscu „in/out” każdego z nich.

5. Dodaj komponent throttle na wyjściu sygnału z bloku multiply. Throttle ogranicza liczbę bitów przechodzących przez ten komponent.

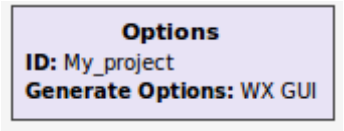
6. W celu wyświetlenia danych przeciągnij komponent „WX GUI Scope Sink” oraz zdefiniuj liczbę sygnałów wejściowych w polu „Num Inputs”.

7. Aby umożliwić nasłuch sygnału dodaj komponent Audio Sink oraz zdefiniuj częstotliwość równą 48KHz.

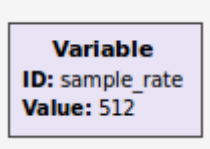
8. W celu zmiany wartości amplitudy sygnału sinusoidalnego przez użytkownika dodaj komponent o nazwie *WX GUI Slider* oraz nadaj mu id o nazwie np. *amplitude_slider*. Zadeklarowane id ustaw w wygenerowanym wcześniej bloku *Signal Source* (w polu *Amplitude),* który jest źródłem sygnału sinusoidalnego.

III. Podstawowe komponenty, bloki, używane podczas laboratoriów:

1) *Options* – główne parametry programu. Możemy ustawić ID naszego projektu, autora, tytuł oraz opis. ID które ustawimy pozwoli na zapis programu w języku python o takiej samej nazwie wraz z rozszerzeniem .py. Bardzo ważne jest wybranie w polu „Generate Options" typu interfejsu graficznego, który będzie wykorzystany w naszych aplikacjach. Należy zwrócić uwagę na QT GUI oraz WX GUI. W przypadku nie ustawienia tego pola zostanie wyświetlony błąd (przeważnie podczas generowania wykresów)

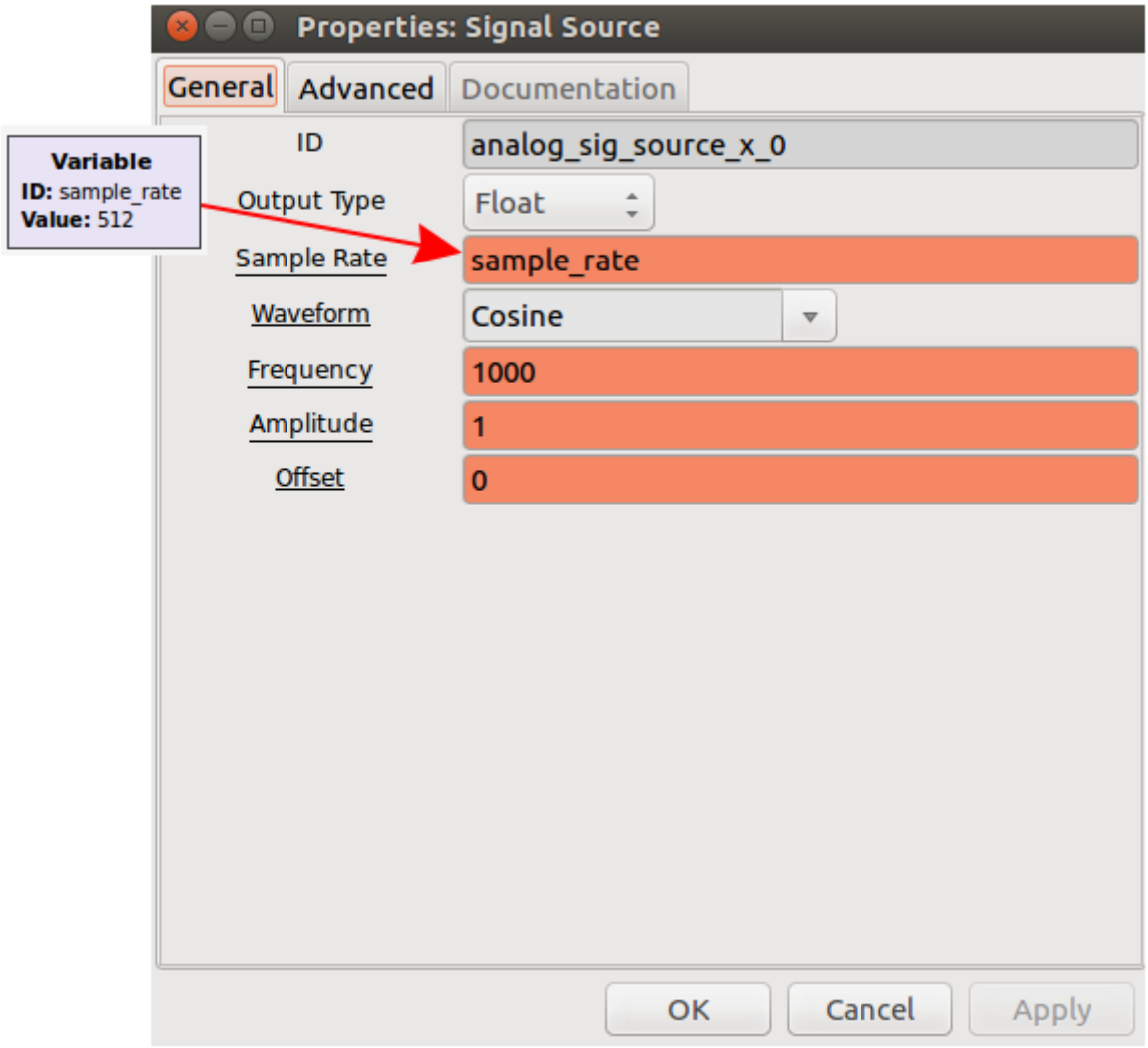


2) *Variable* – jest to zmienna, którą możemy ustawić w projekcie. Konieczne jest ustawienie ID zmiennej oraz podanie wartości.



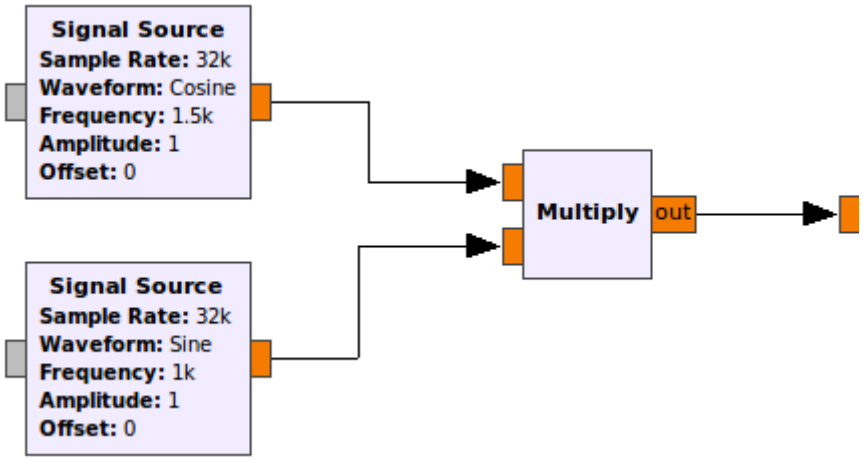
3) *Signal Source* – blok pozwalający na utworzenie źródła sygnału o wybranych parametrach. Definiujemy częstotliwość próbkowania, kształt naszego sygnału np. sinus, cosinus, częstotliwość, amplitudę oraz offset. Pamiętaj aby podczas definiowania wartości wykorzystywać zmienne.

Przykład: Zmienna „sample_rate" umieszczona w polu „Sample Rate".



4) *Multiply* – operator mnożenia. Bloki łączymy ze sobą za pomocą pojedynczego kliknięcia na jeden z nich, np. klikamy na Signal Source oraz na operator Multiply. W poniższym przykładzie używamy operatora do wymnożenia dwóch sygnałów Sinusa oraz Cosinusa.

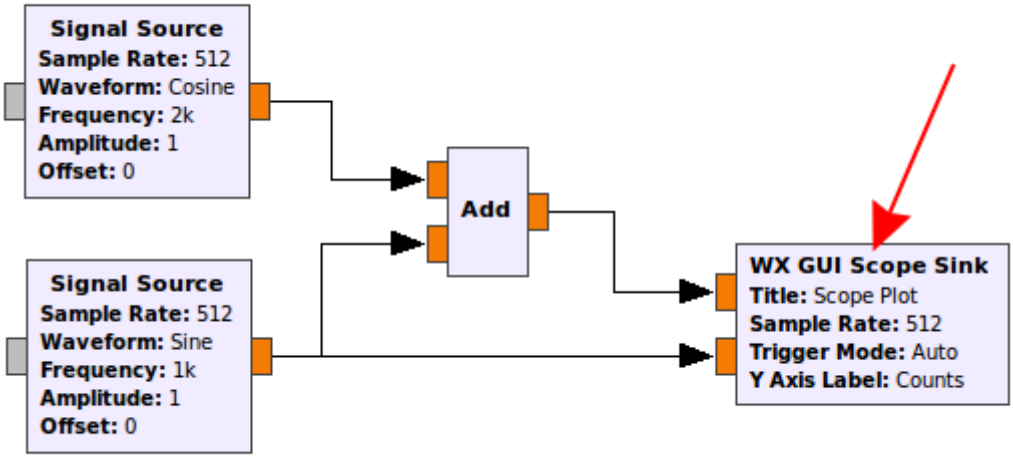
Podobnie możemy używać operatorów matematycznych „Math Operators" mn. takich jak: Add, Subtract, Divide. Występują również operatory logiczne „Boolean Opeators" tj. And, And Const, Not, Or, Xor.



5) Komponenty do obsługi GUI – Graficznego Interfejsu Użytkownika.

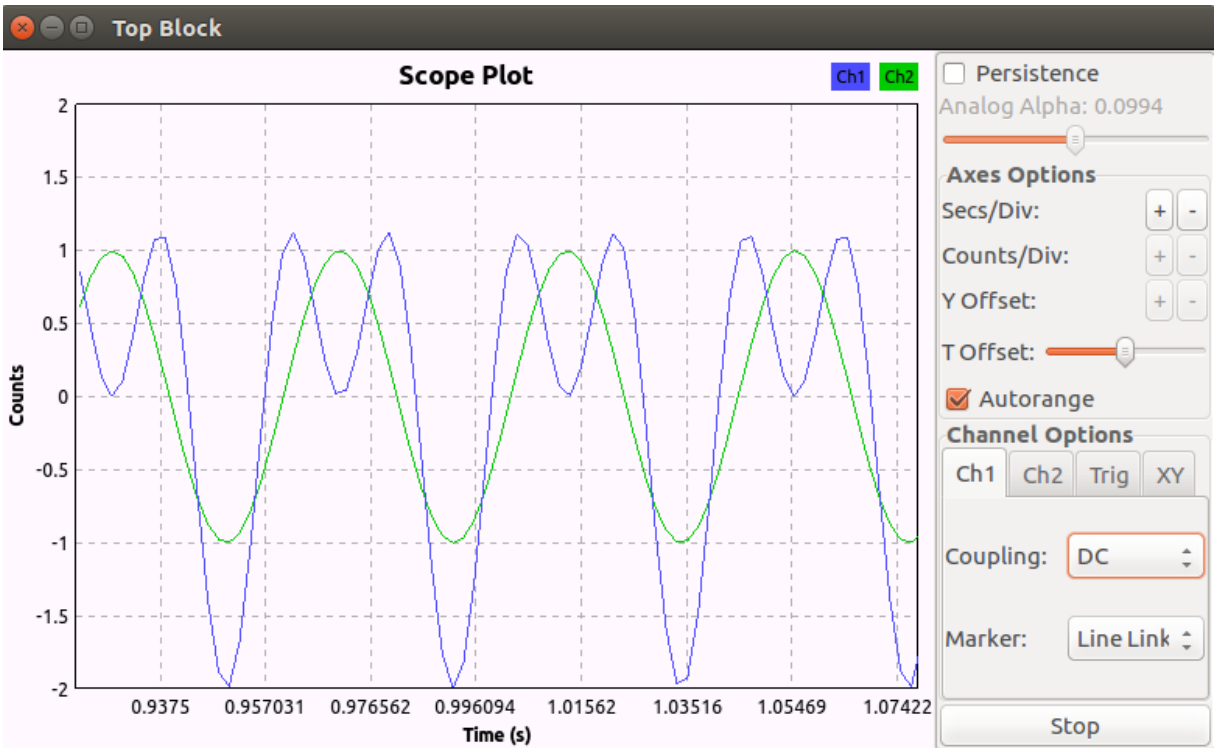
Wykorzystanie biblioteki graficznej WX GUI:

W*X GUI Scope Sink:* Pozwala na wizualizację powstałego sygnału poprzez dodanie do siebie sygnałów (input 1) oraz sygnału o przebiegu sinusoidalnym (input 2)

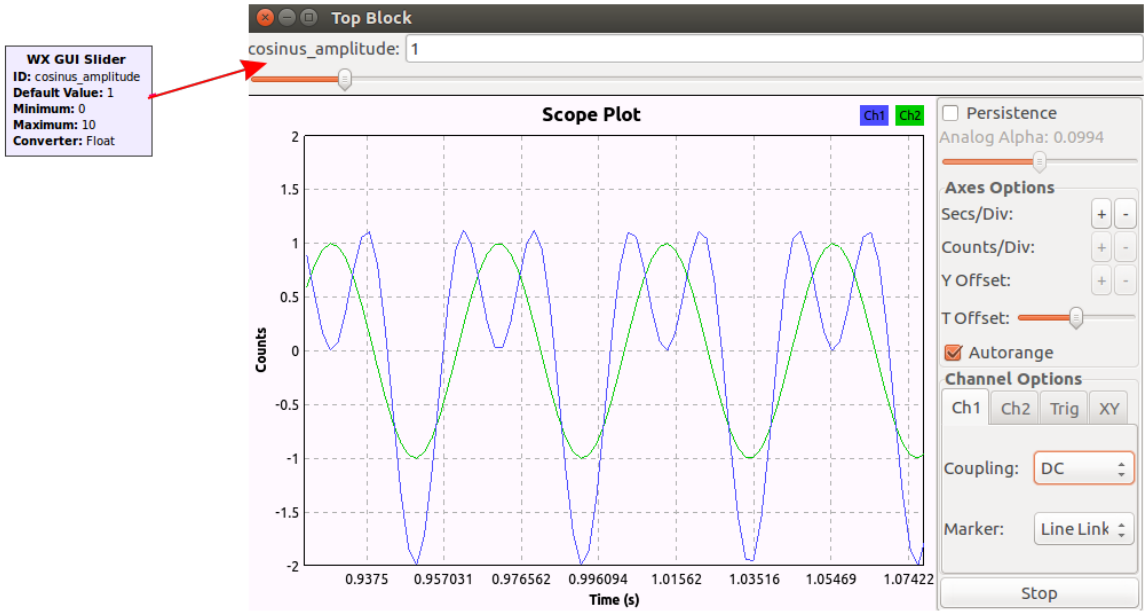


Dzięki powyższemu komponentowi WX GUI Scope Sink możemy odtworzyć wykres dodanych do siebie sygnałów (input 1) oraz wykres sinusa (input 2).

Poniżej zaprezentowano wizualizację graficzną powyższej operacji.



WX GUI Slider: suwak, który służy do zmiany wybranej wartości w GUI użytkownika. Stosując przykładowy suwak wymagane jest aby został wcześniej utworzony blok W*X GUI Scope Sink,* tak by owy slider mógł zostać wyświetlony.



Korzystając z QT GUI :

QT GUI Sink: Wykres danych wejściowych

QT GUI Check Box: check box

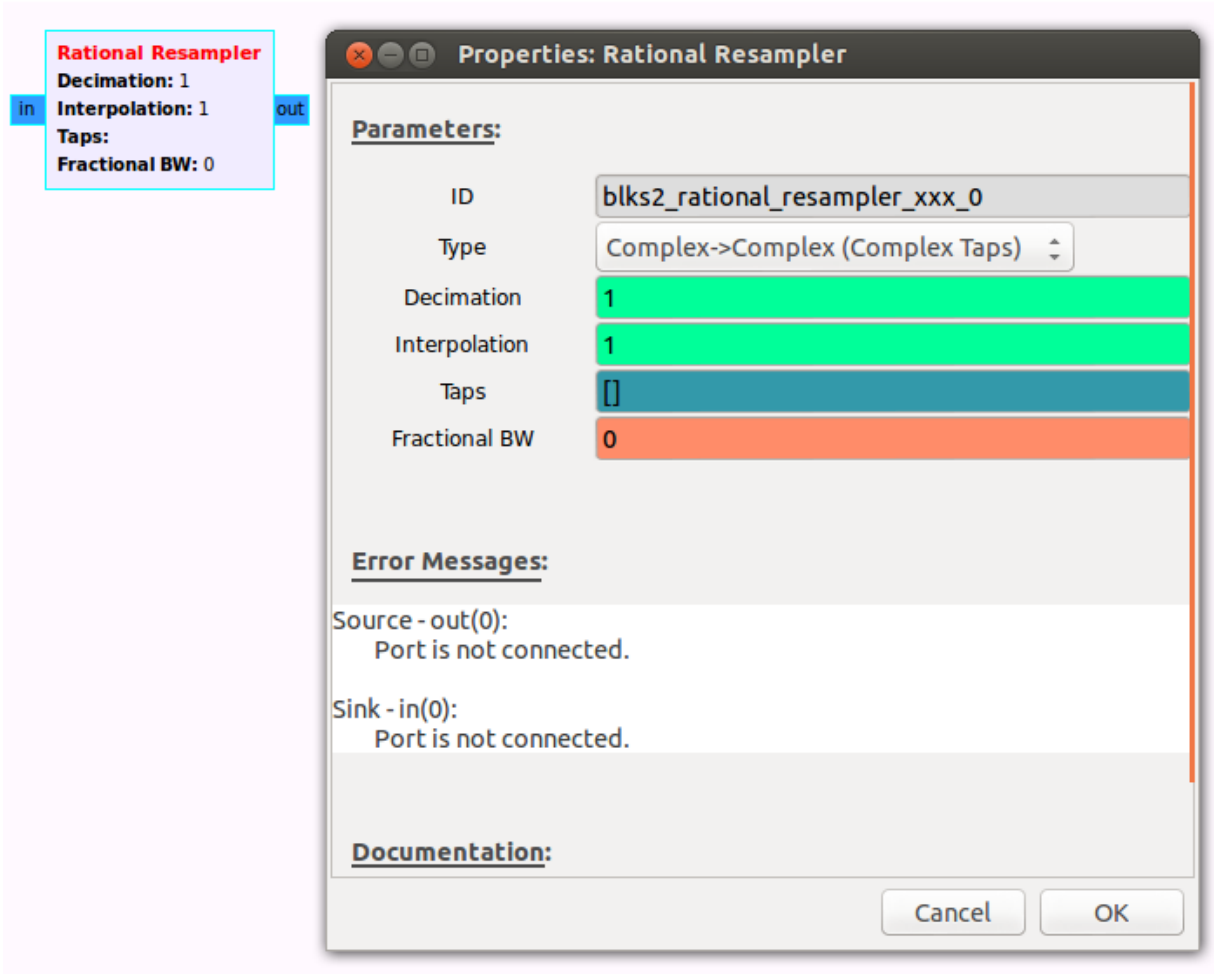
QT GUI Chooser: możliwość wybrania kilku opcji

QT GUI Label: etykieta

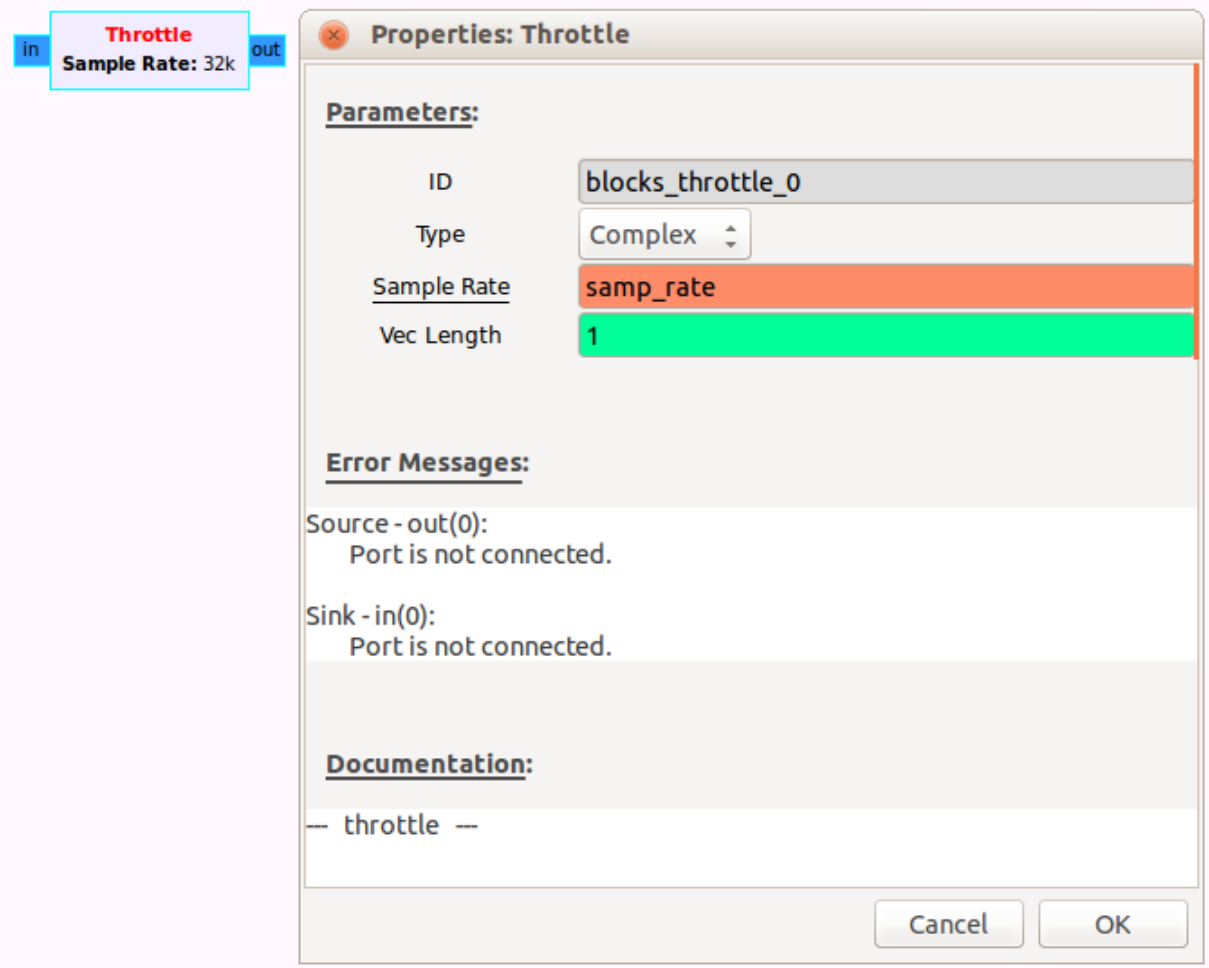
QT GUI Push Button: przycisk

QT GUI Tab Widget: zakładka

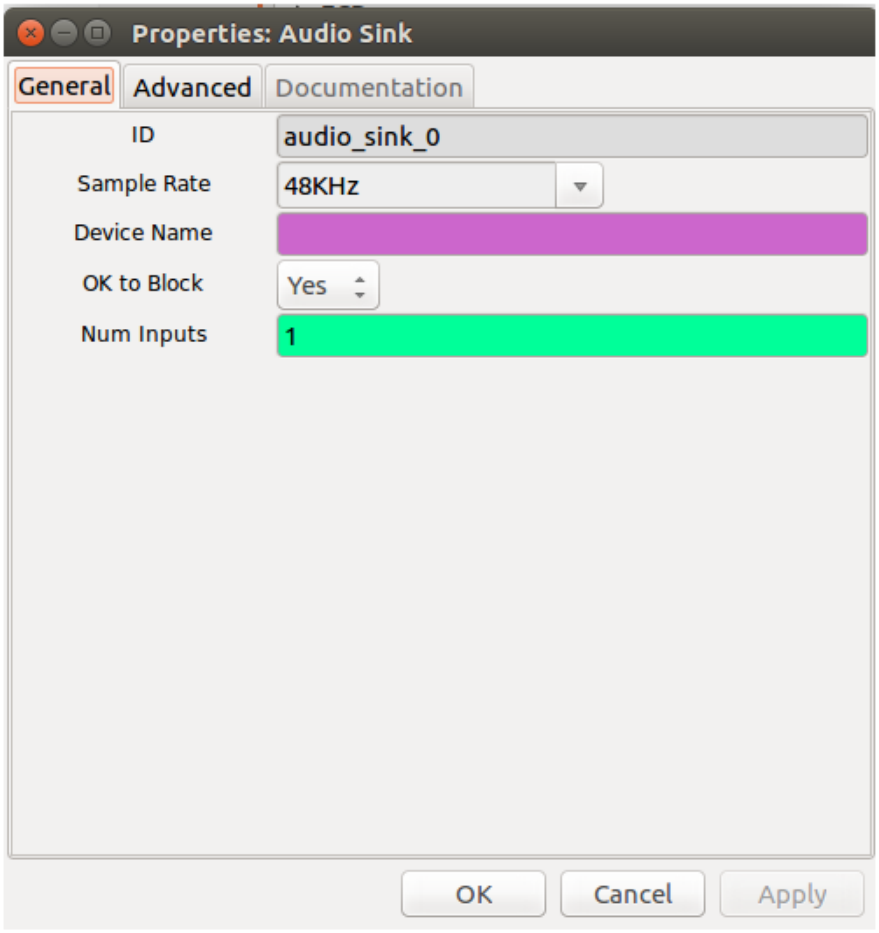
6) *Rational Resampler* – blok odpowiadający za zmianę częstotliwości próbkowania sygnału. Częstotliwość próbkowania sygnału na wyjściu opisana jest zależnością: Fs_out = Fs_in * Interpolation/Decimation



7) Throttle – blok wykorzystywany podczas generowania wykresów typu real time. Ogranicza liczbę bitów przechodzących przez ten komponent. Bardzo często częstotliwość ta jest dopasowywana do częstotliwości próbkowania sygnału. Pominięcie tego bloku spowoduje, że przetwarzane dane będą rysowane na wykresie z maksymalną szybkością na jaką pozwala procesor. Wszystkie zasoby procesora zostaną zużyte jedynie na rysowanie wykresów.



8) Audio Sink – komponent umożliwia odtwarzanie audio sygnału z wykorzystaniem karty dźwiękowej komputera. W polu „Num Inputs” definiujemy liczbę kanałów wejściowych. Dla typowych aplikacji w polu „Sample Rate” możemy ustawić 48KHz.



9) File Operators:

Wav File Sink – Zapis danych o przebiegu sygnału do pliku

Wav File Source - Odczyt danych z pliku dotyczących przebiegu sygnału

IV. Ćwiczenia :

Zadanie 1.

Zapisz do pliku z rozszerzeniem .wav przebieg sygnału powstały z ćwiczenia wprowadzającego. Skorzystaj z komponentu „Wav File Sink".

Zadanie 2.

Wykonaj program odtwarzający zapisany w Zadaniu 1 sygnał. Przedstaw sygnał na wykresie oraz odtwórz go za pomocą komponentu Audio Sink oraz karty dźwiękowej komputera.