



WYDZIAŁ ELEKTRONIKI,  
TELEKOMUNIKACJI  
I INFORMATYKI



# ***TECHNIKA RADIA PROGRAMOWALNEGO***

INSTRUKCJA LABORATORYJNA

## ***ĆWICZENIE***

### **SYMULACJA TRANSMISJI Z WYKORZYSTANIEM ŚRODOWISKA GRC**

Wersja dokumentu: 1.01 z dnia 3.02.2023

# 1. Cel ćwiczenia

Celem ćwiczenia jest zapoznanie studentów z zasadami analizy widma sygnału przy użyciu programu GNU Radio Companion. Podczas laboratorium wykorzystywany jest plik danych zawierający zarejestrowany uprzednio materiał dźwiękowy.

# 2. Wprowadzenie

GNU Radio jest darmowym pakietem narzędzi programistycznych umożliwiających implementację radia programowalnego przy użyciu bloków przetwarzania sygnałów. Stanowi samodzielne środowisko symulacyjne, zapewniające współpracę z zewnętrznymi modułami RF.

GNU Radio Companion (GRC) jest graficznym narzędziem pozwalającym na konstrukcję schematów przepływu informacji, wizualizację wyników w postaci wykresów oraz generację kodu źródłowego w języku programowania Python.

# 3. Zadania do wykonania



Uruchom program GRC używając skrótu w lewym panelu bocznym.

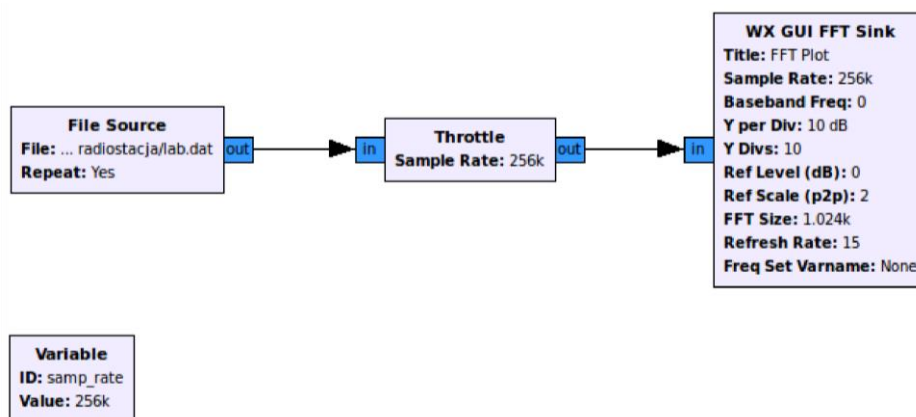
Skonstruuj schemat składający się z następujących elementów: *File Source*, *Throttle*, *FFT Sink*.

Bloki dodaje się poprzez wybór elementu z odpowiedniej kategorii oraz podwójne kliknięcie jego nazwy lub przeciągnięcie na obszar roboczy. Innym sposobem jest wyszukanie elementu za pomocą kombinacji klawiszy *CTRL + F*. Połączenie elementów za pomocą strzałek odbywa się poprzez kliknięcie wyjścia pierwszego, a następnie wejścia drugiego.



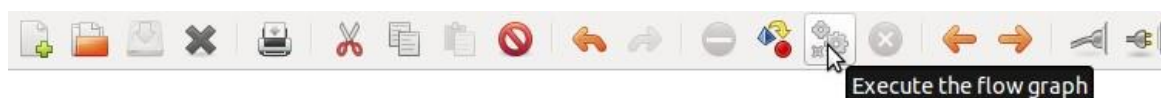
Wczytaj plik z danymi klikając podwójnie na blok *File Source* oraz wybierając ścieżkę (...).

Ustaw częstotliwość próbkowania na 256000[Hz] klikając podwójnie na blok *Variable* oraz edytując parametr *Value*.



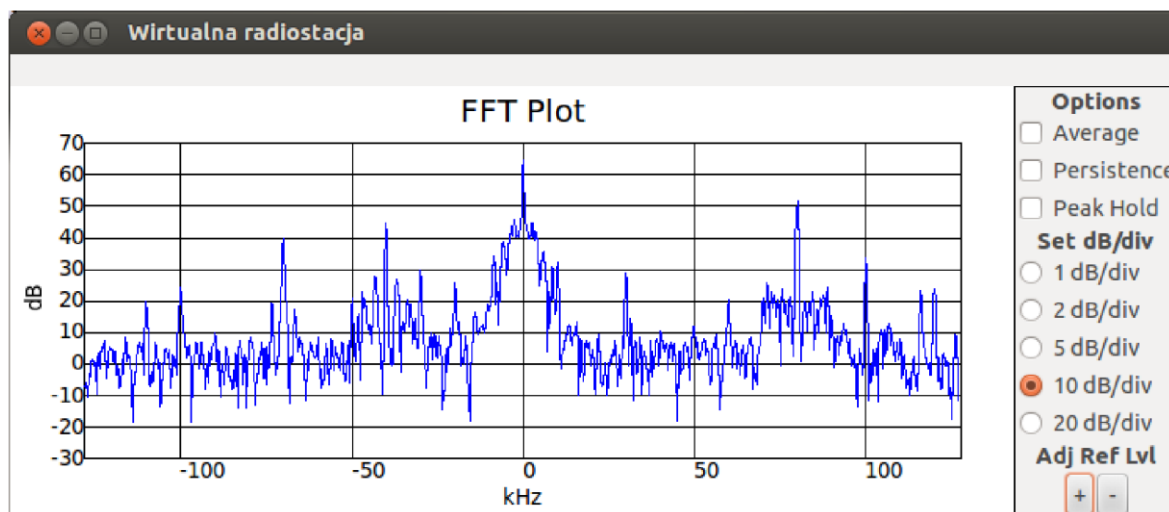
Rys. 1. Wczytywanie i analiza danych.

Zapisz oraz uruchom projekt.



Rys. 2. Uruchamianie projektu.

Powinieneś zaobserwować wykres widma, dostosuj jego poziom za pomocą przycisków *Adj Ref Level*.

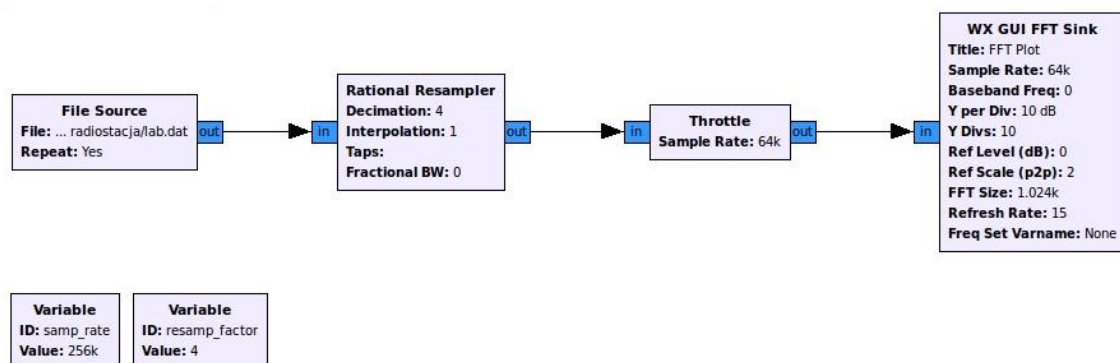


Rys. 3. Wykres widma.

Dane zostały zarejestrowane dla częstotliwości środkowej wynoszącej 710[kHz]. Pomimo wskazania 0[kHz] na środku osi, rzeczywista częstotliwość równa jest 710[kHz]. Zauważ, iż zakres częstotliwości (span) zawarty jest w przedziale od ok. -120[kHz] do +120[kHz]. Wynosi on dokładnie tyle, ile częstotliwość próbkowania. Aby dokładniej zaobserwować sygnały, należy zwiększyć rozdzielczość. W tym celu konieczne jest zmniejszenie częstotliwości próbkowania, np. poprzez operację decymacji.

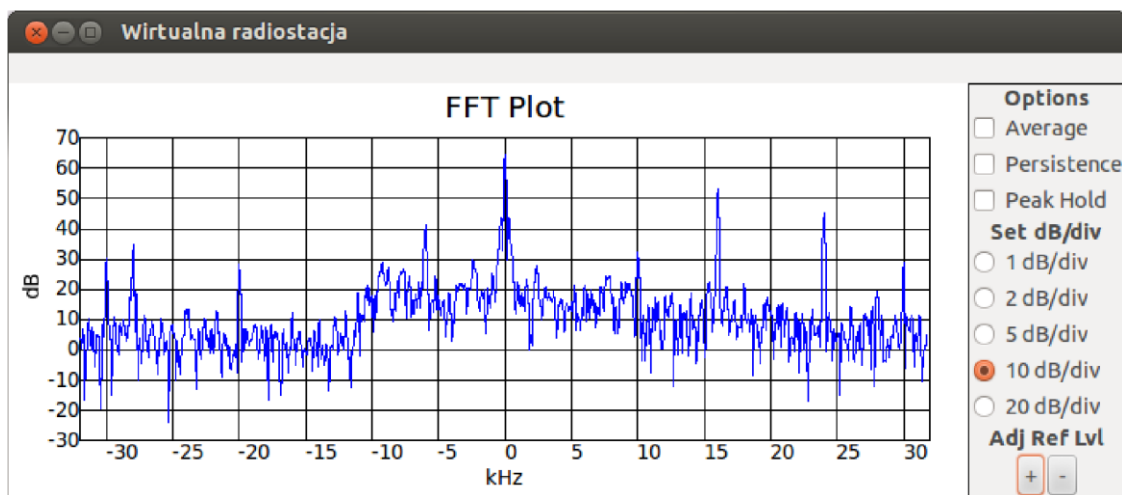
Dodaj blok *Variable* z kolejną zmienną, nadając jej nazwę *ID* „*resamp\_factor*” oraz ustawiając wartość *Value* na „4”. Wstaw blok *Rational Resampler* pomiędzy *File Source* a *Throttle*, w polu *Decimation* wpisz „*resamp\_factor*”. Poprzez wykonanie tej operacji każdy nadchodzący strumień danych [próbki/sekundę] będzie podpróbkowany (próbkowany z mniejszą częstotliwością), w tym wypadku  $256[\text{kHz}] / 4 = 64[\text{kHz}]$ .

W celu dokonania właściwej interpretacji strumienia o nowej przepustowości konieczna jest zmiana częstotliwości próbkowania w blokach *Throttle* oraz *FFT Sink*. Zamień wartość częstotliwości próbkowania *Sample Rate* na „*samp\_rate/resamp\_factor*”. Dzięki temu po każdej edycji parametru decymacji „*resamp\_factor*” zmiany zostaną odwzorowane we wszystkich blokach.



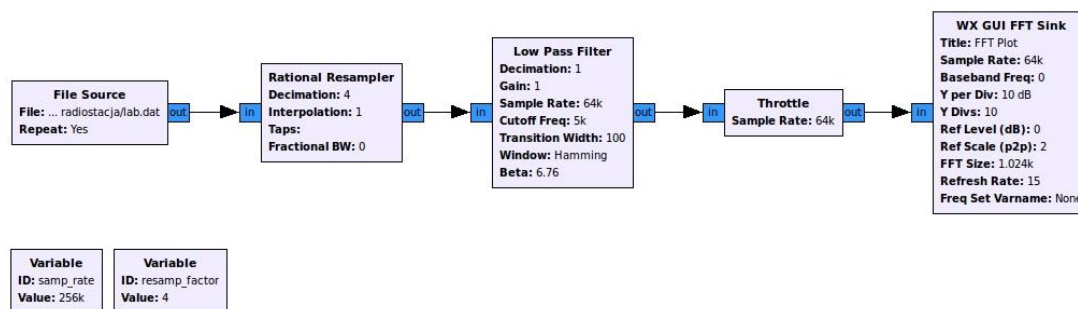
Rys. 4. Zmiana częstotliwości próbkowania.

Zapisz i uruchom projekt. Zauważ, iż span równy jest dokładnie 64[kHz], zaś pasmo sygnału AM wynosi ok. 10[kHz] (+/-5[kHz] od częstotliwości nośnej).



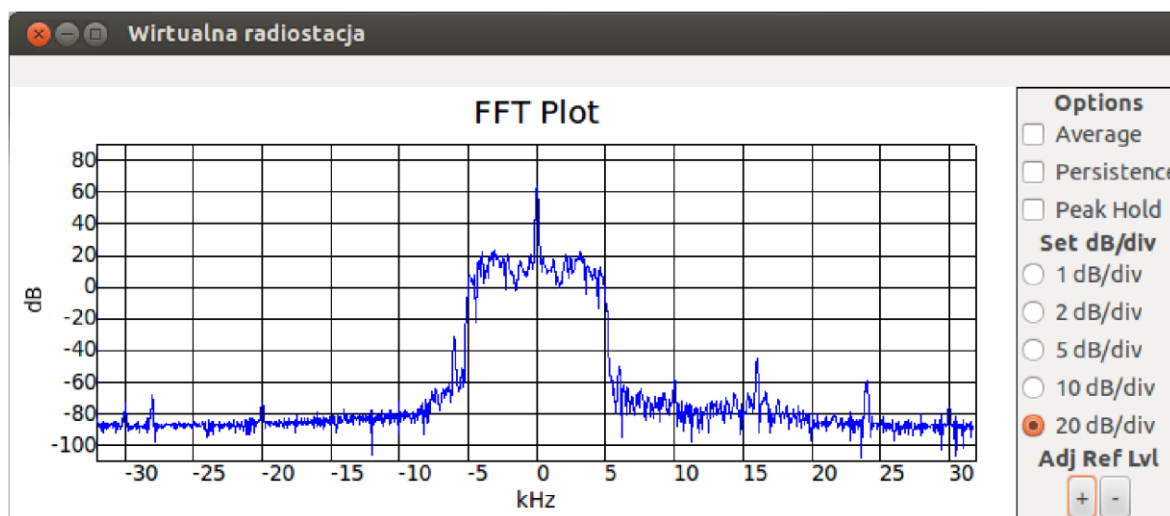
Rys. 5. Wykres widma po zmianie częstotliwości próbkowania.

Aby odebrać sygnał nadawany na częstotliwości 710[kHz] (0[kHz]) konieczne jest zastosowanie filtra eliminującego składowe niepożądane. W tym celu należy umieścić filtr dolnoprzepustowy *Low Pass Filter* o częstotliwości odcięcia 5[kHz] pomiędzy blokami *Rational Resampler* a *Throttle*. Ustaw wartość parametru *Sample Rate* na „*samp\_rate/resamp\_factor*”, *Cutoff Freq* jako „5000”, *Transition Width* „100”, pozostałe pozostaw bez zmian.



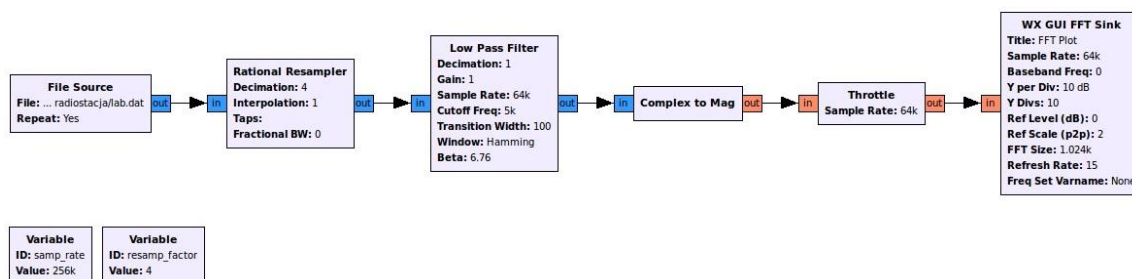
Rys. 6. Filtracja sygnału.

Zapisz i uruchom projekt. Pozostała wyłącznie jedna stacja pomiędzy -5[kHz] a +5[kHz], reszta składowych została odfiltrowana.



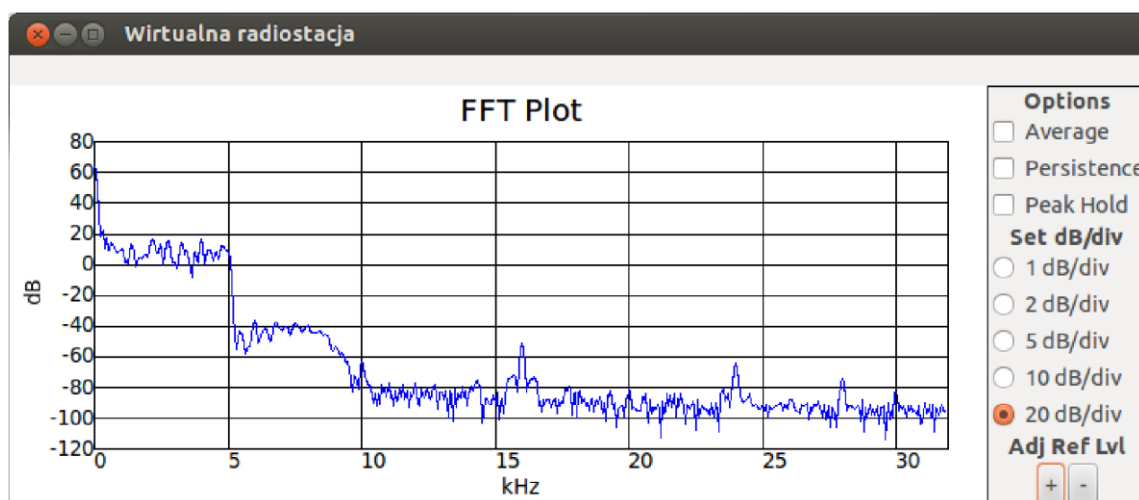
Rys. 7. Wykres widma po filtracji.

Dokonaj demodulacji sygnału umieszczając blok *Complex to Mag* pomiędzy elementami *Low Pass Filter* a *Throttle*. Spróbuj uruchomić projekt. Wystąpił błąd, gdyż nowo dodany element podaje na wyjście dane w postaci rzeczywistej (float) a nie zespolonej (complex). Zmień typy danych w *Throttle* oraz *FFT Sink* z „Complex” na „Float” i uruchom ponownie projekt.



Rys. 8. Demodulacja sygnału.

Zauważ, iż wykres widma obejmuje jedynie wartości dodatnie.



Rys. 9. Wykres części rzeczywistej widma po demodulacji.

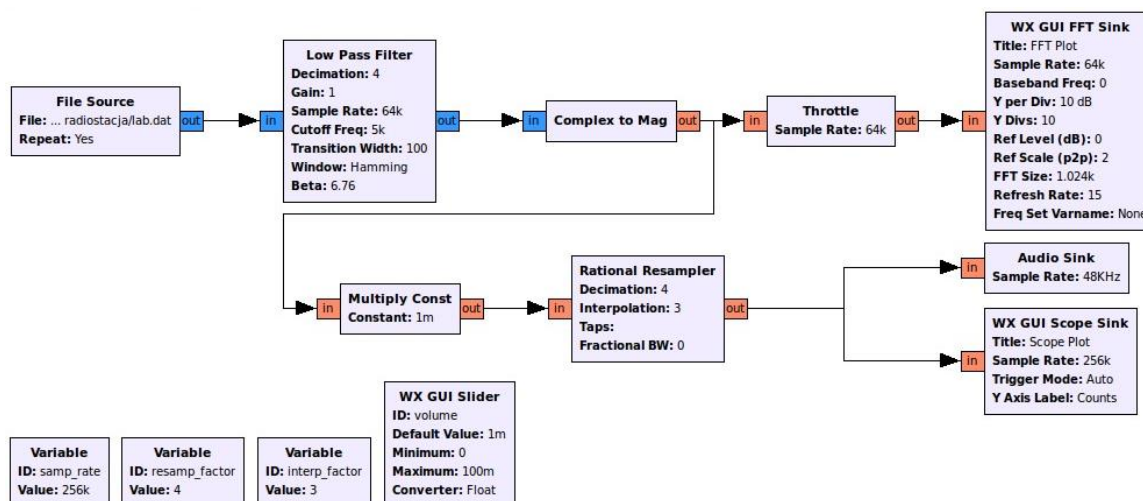
Aby posłuchać sygnału zdemodulowanego dodaj element *Audio Sink* po bloku *Complex to Mag*. Program domyślnie ustawił częstotliwość próbkowania zgodnie ze zmienną „*samp\_rate*”. Większość kart dźwiękowych nie obsługuje takich wartości. Zmień częstotliwość próbkowania w bloku *Audio Sink* na 48[kHz]. W celu zaobserwowania sygnału audio umieść blok *Scope Sink* równolegle do *Audio Sink*. Zapisz i uruchom projekt. Jak oceniasz odbiór?

Na wyjściu bloku *Complex to Mag* częstotliwość próbkowania wynosi 64[kHz], zaś wartość przypisana dla karty dźwiękowej 48[kHz]. Potrzebna jest zatem pewnego rodzaju konwersja strumienia (decymacja i interpolacja).

Dodaj nową zmienną *Variable* o nazwie ID „*interp\_factor*”, przypisując jej wartość *Value* „3”. Następnie umieść kolejny blok *Rational Resampler* pomiędzy *Complex to Mag* a *Audio Sink* oraz zmień typ danych z „*Complex*” na „*Float*”. Ustaw współczynniki *Decimation* i *Interpolation* w obu blokach *Rational Resampler* odpowiednio na „*resamp\_factor*” oraz „*interp\_factor*”. Zauważ, iż filtr *Low Pass Filter* ma także funkcję podpróbkowania (parametr *Decimation*). Przypisz jej wartość „*resamp\_factor*”, a następnie usuń pierwszy *Rational Resampler*. Czy obliczenia się teraz zgadzają? Czy słyszysz wyraźny dźwięk?

Sygnał nie jest wyraźny, ponieważ wartości niektórych próbek przekraczają te obsługiwane przez głośniki. Potrzebny jest zatem mechanizm kontrolny, działający na zasadzie tłumika, który pozwalałby sterować nimi. Dodaj blok *Slider* o nazwie ID „*volume*” oraz parametrach *Default Value* „0.001”, *Minimum* „0”, *Maximum* „0.1”, *Num Steps* „100”. Następnie umieść element *Multiply Const*, pomiędzy *Complex to Mag* a *Rational Resampler*, zmieniając jego typ danych na „*Float*” oraz przypisując w polu *Constant* wartość „*volume*”. Zapisz i uruchom projekt.



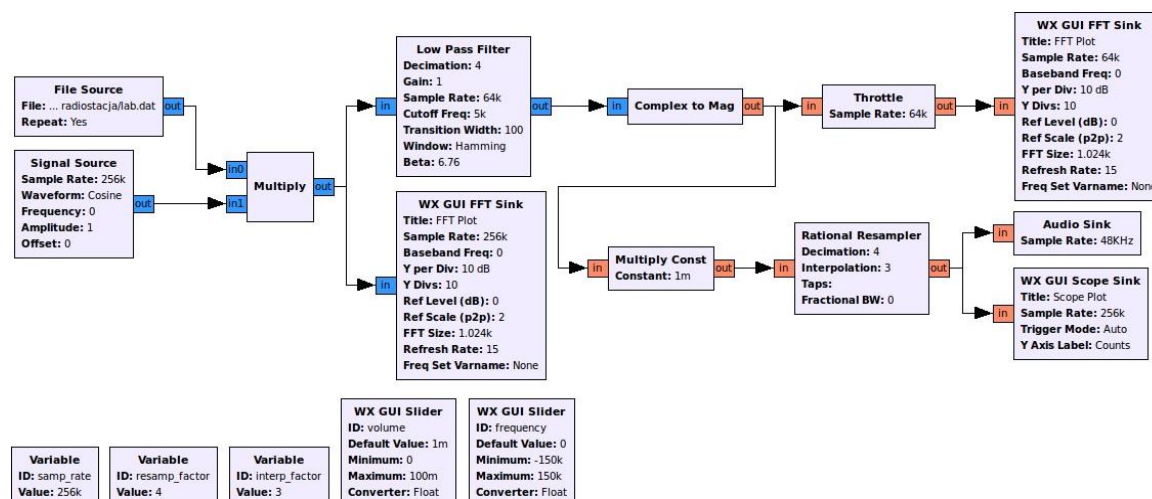


Rys. 10. Odsłuch sygnału audio.

Przypomnij sobie listę częstotliwości pozostałych stacji radiowych. Aby możliwy był odbiór dowolnej dostępnej stacji potrzebny jest sygnał, np. kosinusoidalny, o „ujemnej częstotliwości”. Taki sygnał pozwoliłby na przejście do „zerowej częstotliwości” (mowa tu o wykresie widma), co zapewniłoby skuteczne odfiltrowanie składowych niepożądanych.

Dodaj kolejny *Slider* o nazwie *ID „frequency”* oraz parametrach *Default Value „0”*, *Minimum „-150000”*, *Maximum „150000”*, *Num Steps „1000”*. Umieść blok *Signal Source* i przypisz wielkości *Frequency* wartość „frequency”, aby zapewnić sprawny dobór częstotliwości. Następnie wstaw blok *Multiply*, do którego wejścia podłącz wyjścia z *File Source* oraz *Signal Source*, zaś wyjście połącz z filtrem *Low Pass Filter*. Zaobserwuj sygnał wynikowy po przemnożeniu umieszczając dodatkowy blok *FFT Sink* na wyjściu bloku *Multiply*. Zapisz i uruchom projekt.

Czy wszystkie stacje są jednakowo wyraźne? Dostosuj poziom głośności za pomocą suwaka.



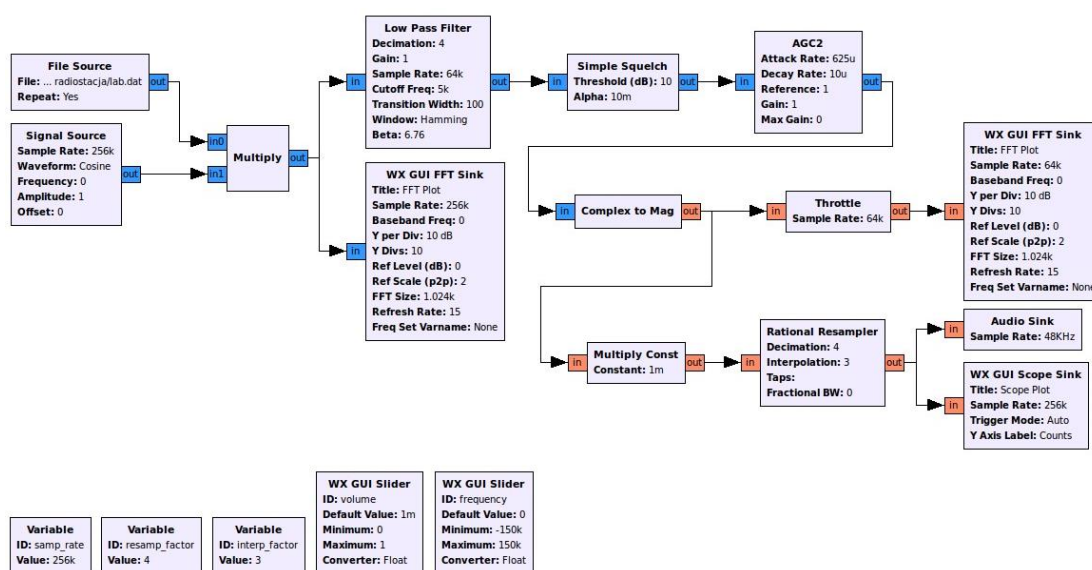
Rys. 11. Odbiór wielu stacji radiowych.

Poziom głośności może być regulowany automatycznie za pomocą modułu ARW

(Automatycznej Regulacji Wzmocnienia), którego zasada działania oparta jest na próbkowaniu wartości podawanych na wyjście oraz dostosowaniu wzmocnienia w celu zapewnienia odpowiedniego średniego poziomu tych wartości. Moduł ten umożliwia dokonanie „miękkiego przejścia” pomiędzy stacjami, jest szeroko stosowany w odbiornikach radiowych i telewizyjnych.

Umieść bloki *Simple Squelch* oraz *AGC2* pomiędzy *Low Pass Filter* a *Complex to Mag*. Element *Simple Squelch* za pomocą wielkości *Threshold* określa granicę, w obrębie której następuje aktywacja układu regulacji wzmacnienia, z kolei *Alpha* definiuje jak długo będzie otwarty przepływ (hang time) po zaniku sygnału. Na podstawie wykresu widma określ optymalną wartość parametru *Threshold*, zaś *Alpha* ustaw jako „0.01”. W przypadku *AGC2* przypisz wielkościom *Attack Rate* oraz *Decay Rate* odpowiednio „625e-6” oraz „1e-5”. Zwiększ maksymalną głośność z „0.1” do „1.0”.

## Zapisz i uruchom projekt.



**Rys. 12. Odbiór wielu stacji radiowych z układem ARW.**



## ***Literatura***

- [1] <http://gnuradio.org/redmine/projects/gnuradio/wiki>.
- [2] Notatki z wykładów.