

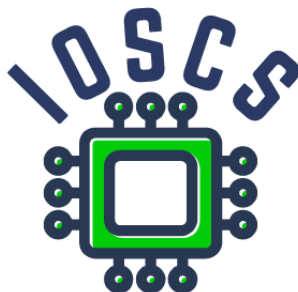
Project: Innovative Open Source Courses for Computer Science

Bezprzewodowe przetwarzanie sygnałów w środowisku GNU Radio Laboratoria

**Wojciech Sałabun
West Pomeranian University of Technology in Szczecin**

30.05.2021

Innovative Open Source Courses for Computer Science



This teaching material was written as one of the outputs of the project “Innovative Open Source Courses for Computer Science”, funded by the Erasmus+ grant no. 2019-1-PL01-KA203-065564. The project is coordinated by West Pomeranian University of Technology in Szczecin (Poland) and is implemented in partnership with Mendel University in Brno (Czech Republic) and University of Žilina (Slovak Republic). The project implementation timeline is September 2019 to December 2022.

Project information

Project was implemented under the Erasmus+.

Project name: **“Innovative Open Source courses for Computer Science curriculum”**

Project nr: **2019-1-PL01-KA203-065564**

Key Action: **KA2 – Cooperation for innovation and the exchange of good practices**

Action Type: **KA203 – Strategic Partnerships for higher education**

Consortium

ZACHODNIOPOMORSKI UNIWERSYTET TECHNOLOGICZNY W SZCZECINIE

MENDELOVA UNIVERZITA V BRNE

ZILINSKA UNIVERZITA V ZILINE

Erasmus+ Disclaimer

This project has been funded with support from the European Commission. This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Copyright Notice

This content was created by the IOSCS consortium: 2019–2022. The content is Copyrighted and distributed under Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0).



Co-funded by the
Erasmus+ Programme
of the European Union

Laboratorium 1
Wprowadzenie do środowiska GNU Radio

30.05.2021

GNU Radio to darmowe oprogramowanie, które umożliwia użytkownikowi projektowanie i implementację przetwarzanych sygnałów bez potrzeby angażowania sprzętów jak generator sygnałów bądź oscyloskop. Umożliwia generowanie kodu do języka Python oraz stanowi alternatywę dla programów takich jak Matlab czy LabView. Pliki projektu zapisywane są z rozszerzeniem .grc.

Oprogramowanie może zostać pobrane ze strony <https://www.gnuradio.org/>

Szczegółowa dokumentacja dostępna jest pod linkiem <https://www.gnuradio.org/docs/>

Po utworzeniu projektu, w obszarze roboczym pojawiają się 2 podstawowe komponenty

- Opcje projektu
- Zmienna określająca częstotliwość próbkowania systemu

Podstawowy element interfejsu GNU Radio

- Lista komponentów środowiska GRC – prawa strona
 - Komponenty dodaje się poprzez przeciągnięcie elementu z listy do obszaru głównego
- Konsola programu wraz z zadeklarowanymi zmiennymi – dolna część GUI
- Obszar roboczy wraz z paskiem ustawień – górna część GUI
 - uruchomienie programu
 - generowanie kodu do pliku w języku Python z rozszerzeniem .py
 - przerwanie działania programu

Podstawowe komponenty

- Options – główne parametry programu (tytuł, opis, ID projektu), należy ustawić pole Generate Options, w innym przypadku zostanie zwrócony błąd
- Variable – zmienna w programie, należy ustawić ID i wartość
- Signal Source (zakładka Waveform generators)– blok tworzący źródło sygnału o wybranych parametrach, określamy częstotliwość, amplitudę, offset, kształt sygnału, typ danych, należy określić typ danych jako float
- Multiply – mnożenie dwóch sygnałów, analogicznie można wykorzystywać pozostałe operatory matematyczne jak Add, Subtract, Divide oraz operatory logiczne jak And, Or, Xor, Not. Bloki łączymy za pomocą pojedynczego kliknięcia na jeden z nich a następnie na wybrany operator

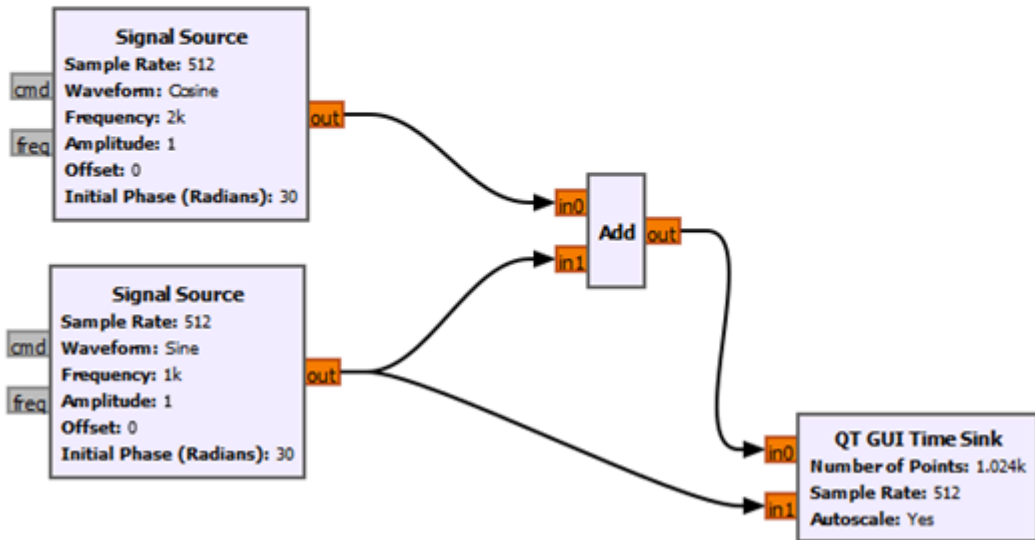
- Komponenty graficzne
 - Sink – umożliwiają wizualizację zaprojektowanego sygnału, należy określić typ danych jako float
 - Panel ustawień wykresu
- Rational Resampler – blok służący do zmiany częstotliwości próbkowania sygnału
- Throttle – blok wykorzystywany do generowania wykresów typu real time, ogranicza wykorzystaną liczbę bitów przechodzących przez komponent
- Audio Sink – do odtwarzania audio z sygnału za pomocą karty dźwiękowej, typowa wartość Sample Rate to 48KHz
- Operatory na plikach
 - Wav File Source – odczyt danych z pliku
 - Wav File Sink – zapis danych do pliku

Zadania do wykonania

- Zbudować 5 układy o zróżnicowanych właściwościach
 - wykorzystać operatory matematyczne i logiczne
 - wprowadzić różne ustawienia generowanego sygnału
- Wykorzystać Sink to wizualizacji układów
 - Wykorzystać różne ustawienia do wyświetlania wykresów

Options
Title: L1
Output Language: Python
Generate Options: QT GUI

Variable
Id: samp_rate
Value: 512



Rysunek 1: Przykładowy układ.

Laboratorium 2
Reprezentacje sygnałów w dziedzinie czasu i
częstotliwości

30.05.2021

Celem laboratorium jest zbadanie działania układów reprezentujących sygnały w dziedzinie czasu oraz w dziedzinie częstotliwości. Dodatkowo, należy dokonać analizy i porównania otrzymanych wyników względem siebie oraz zbadać właściwości sygnału przy ustalonych parametrach.

Szczegółowa dokumentacja dostępna jest pod linkiem <https://www.gnuradio.org/docs/>

W celu stworzenia układu do wizualizacji sygnału w dziedzinie czasu należy wykorzystać bloki poznane na poprzednich zajęciach. Natomiast, by utworzyć układ do reprezentacji sygnału w dziedzinie częstotliwości należy wykorzystać poniższe bloki funkcyjne.

- FFT – główny blok realizujący Szybką Transformatę Fouriera
 - FFT Size – liczba próbek użytych do obliczenia FFT
 - Forward/Reverse – czy użyć FFT czy iFFT
 - Window – jaki typ okna użyć dla każdego zbioru zanim zostanie wykonane FFT, domyślna opcja to blackmanharris, opisuje ile punktów ma znaleźć się w każdym oknie, rozmiar musi zgadzać się z wartością parametru FFT Size
 - Shift – czy wykonywać FFT shift, można wykonywać tylko dla złożonych danych wejściowych
 - Num Threads – ile wątków zaangażować do obliczenia FFT
- Frequency Sink – blok funkcyjny umożliwiający wyświetlanie sygnałów w dziedzinie częstotliwości
 - FFT Size - liczba próbek użytych do obliczenia FFT
 - Wintype – typ okna
 - Center Frequency – częstotliwość do której wycelowany będzie wykres
 - Bandwidth – szerokość pasma wyświetlanego sygnału
 - Name – tytuł wykresu

Podstawowy układ z wykorzystaniem FFT:

- Sygnał wejściowy
- Blok Throttle do generowania układów w czasie rzeczywistym
- Blok FFT
- Block Vector Sink/Virtual Sink do wizualizacji

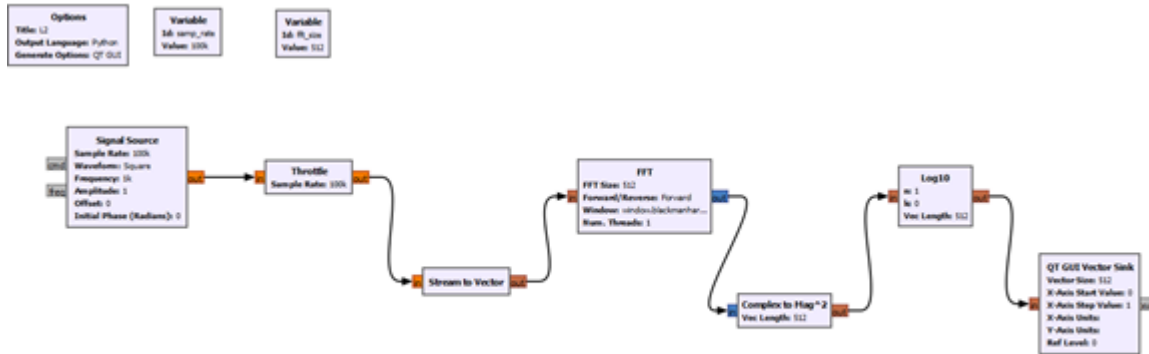
Podstawowy układ z wykorzystaniem Frequency Sink:

- Sygnał wejściowy
- Blok Throttle do generowania układów w czasie rzeczywistym
- Blok wizualizacji Frequency Sink

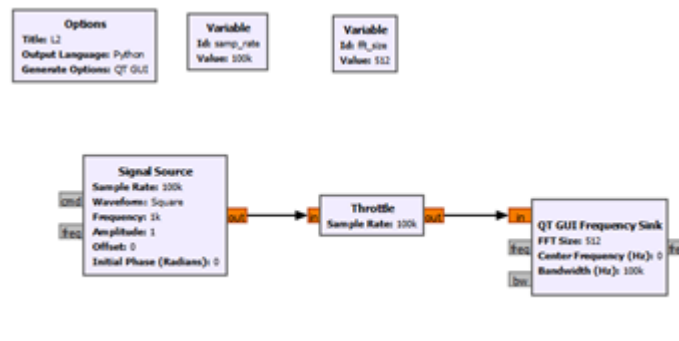
Układy ten można rozszerzać o dodatkowe wejścia sygnałów, bloki operacji arytmetycznych/logicznych, bloki modulacji (ASK, FSK, PSK)

Zadania

- Zbudować 2 układy o zróżnicowanych parametrach w dziedzinie czasu wykorzystując wiedzę i bloki funkcyjne z poprzednich zajęć
- Dodać możliwość wyświetlania i wizualizacji zaprojektowanego sygnału
- Zbudować 2 układy o zróżnicowanych parametrach w dziedzinie częstotliwości w oparciu o bloki funkcyjne z poprzednich zajęć oraz z bieżącej instrukcji opartych o blok FFT
- Zbudować 2 układy o zróżnicowanych parametrach w dziedzinie częstotliwości w oparciu o bloki funkcyjne z poprzednich zajęć oraz z bieżącej instrukcji opartych o blok Frequency Sink



Rysunek 2: Przykładowy układ z wykorzystaniem bloku FFT.



Rysunek 3: Przykładowy układ z wykorzystaniem bloku Frequency Sink

Laboratorium 3
Realizacja systemów modulacji ciągłych amplitudy
i kąta

30.05.2021

Celem laboratorium jest zbadanie wpływu użytych modulacji ciągłych na właściwości sygnału wyjściowego. Do tego wykorzystane zostaną modulacja amplitudy i kąta. Po zaprojektowaniu układów i zastosowaniu modulacji należy ocenić szerokość pasma sygnałów oraz dokonać analizy sygnałów w dziedzinie częstotliwości.

Szczegółowa dokumentacja dostępna jest pod linkiem <https://www.gnuradio.org/docs/>

Modulacja amplitudy w GNU Radio

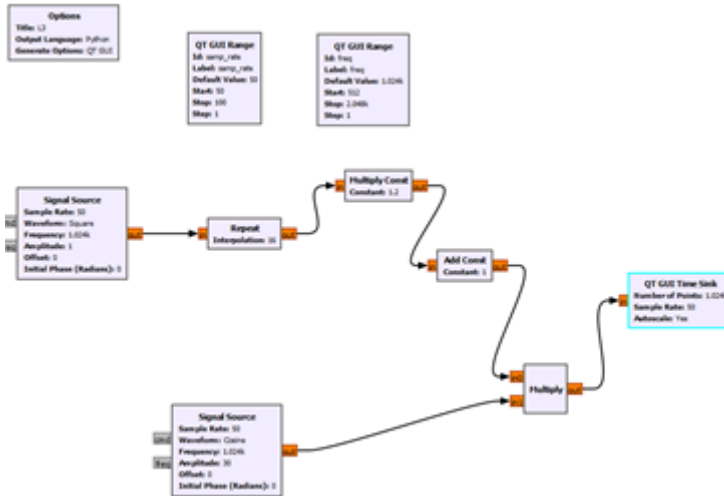
- Źródła sygnału wejściowego o różnych parametrach
- Operacja arytmetyczna (np. Multiply)
- Blok do wyświetlenia sygnału (Time Sink)

Modulacja kąta w GNU Radio

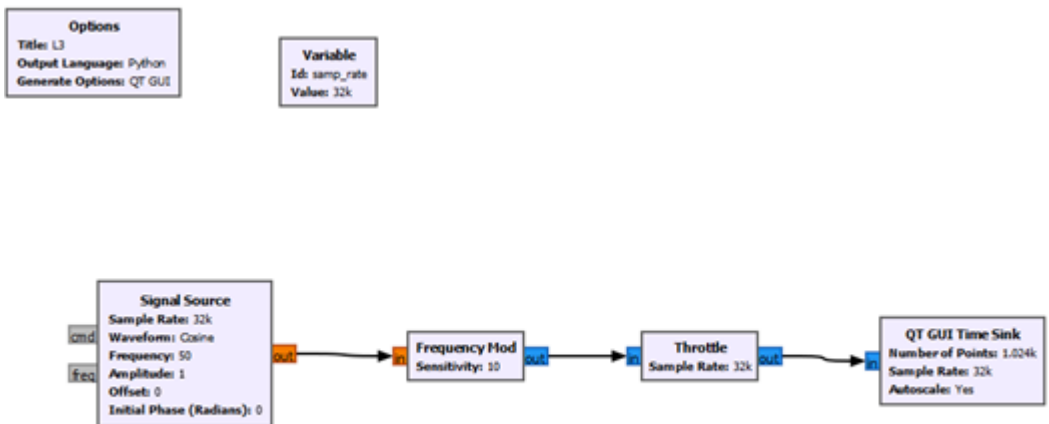
- Źródło sygnału wejściowego
- Blok Frequency Mod
- Blok Throttle
- Blok do wyświetlenia sygnału (Time Sink)

Zadania

- Zbudować 2 układy dla modulacji amplitudy
 - Zastosować różne parametry wejściowe dla sygnału
 - Zaprojektowane układy wyświetlić w dziedzinie czasu i w dziedzinie częstotliwości
 - Porównać otrzymane wyniki
- Zbudować 2 układy dla modulacji kąta
 - Zastosować różne parametry wejściowe dla sygnału
 - Zaprojektowane układy wyświetlić w dziedzinie czasu i w dziedzinie częstotliwości
 - Porównać otrzymane wyniki



Rysunek 4: Przykładowa realizacja układu dla AM.



Rysunek 5: Przykładowa realizacja układu dla FM.

Laboratorium 4
Kluczowanie amplitudy, fazy, częstotliwości

30.05.2021

Celem laboratorium jest zbadanie właściwości sygnału zmodulowanego przy użyciu modulacji ASK, FSK i PSK. Należy dokonać modulacji i demodulacji sygnału wejściowego za pomocą wybranych kluczozań. Następnie należy dokonać analizy otrzymanych przebiegów sygnału.

Szczegółowa dokumentacja dostępna jest pod linkiem <https://www.gnuradio.org/docs/>

Modulacja ASK:

On-Off Keying Modulation

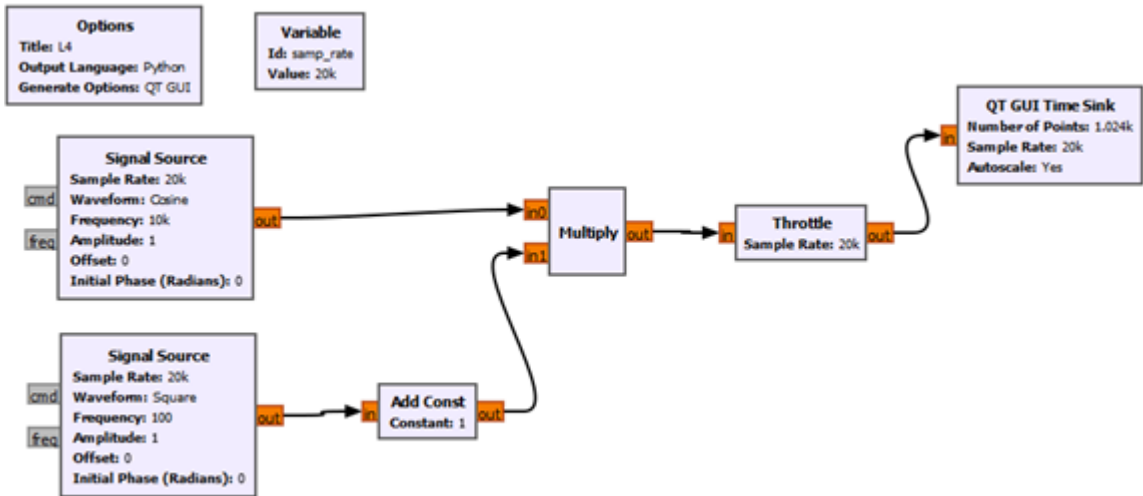
- Dwa sygnały wejściowe:
 - Sygnał nośny
 - * Typ: sinus
 - * Częstotliwość: 10KHz
 - Sygnał modulowany
 - * Typ: square
 - * Częstotliwość: 100 Hz
 - * Można dodać blok Slider dla zmiany wartości parametrów sygnałów
 - Blok operacji mnożenia Multiply
 - Blok Throttle
 - Wizualizacja Scope Sink

ASK

- Dla układu z On-Off Keying Modulation
 - Zmienić amplitudę modulowanego sygnału (przykładowo z 0,1 na 1,2) poprzez dodanie bloków „Add const”
 - Wizualizacja Scope Sink

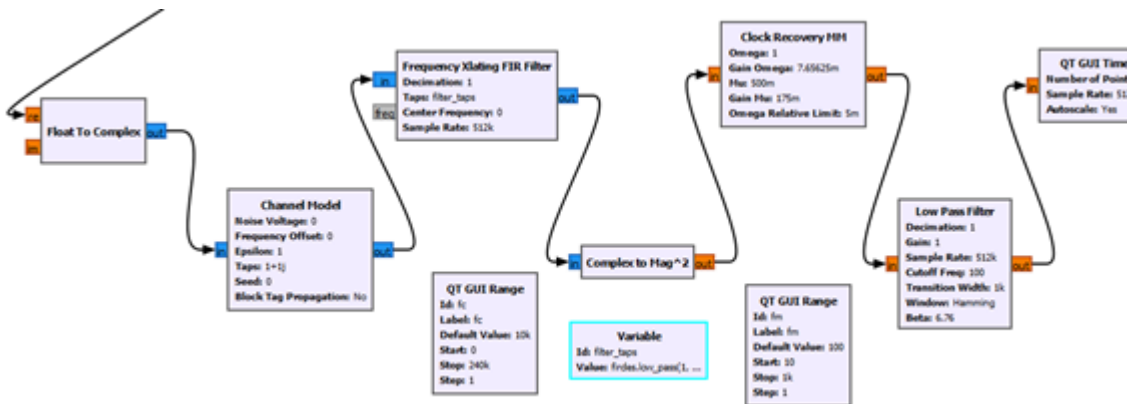
Demodulacja

- Blok Float To Complex do zmiany reprezentacji wartości
- Blok Channel Model – do retransmisji sygnał
- Blok filtra częstotliwości Xlating FIR Filter
 - Wymaga zdefiniowania zmiennej o podanej wartości:
firdes.low_pass(1, samp_rate, fc, 25000, firdes.WIN_HAMMING, 6.76)



Rysunek 6: Modulacja ASK.

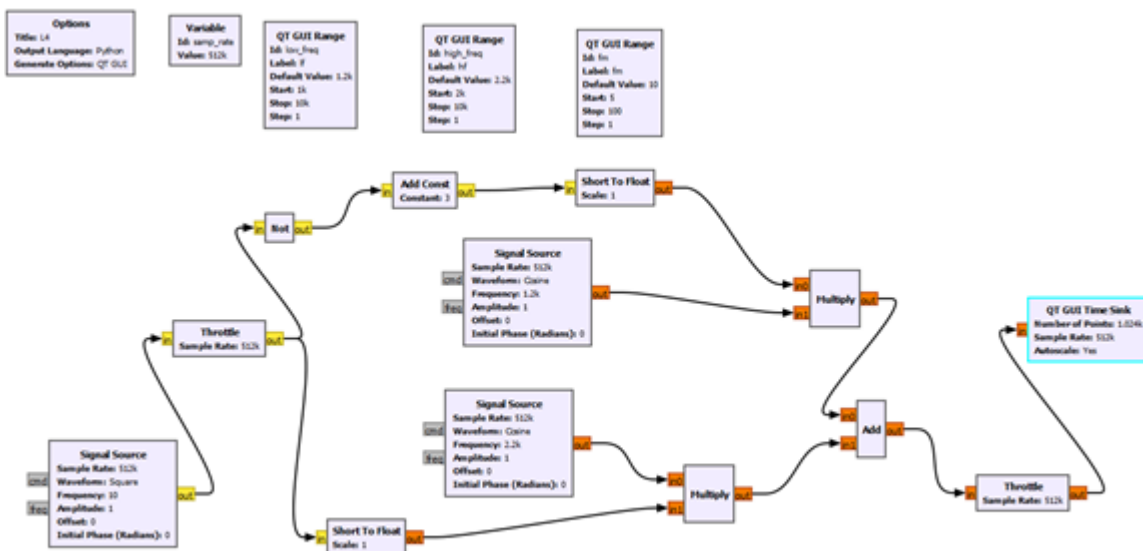
- Block Complex to Mag² do obliczenia części rzeczywistej i urojonej
- Blok Clock Recovery MM służący do dekodowania symboli
- Low Pass Filter w celu filtrowania sygnału demodulowanego
- Wizualizacja Scope Sink



Rysunek 7: Demodulacja ASK.

Modulacja FSK:

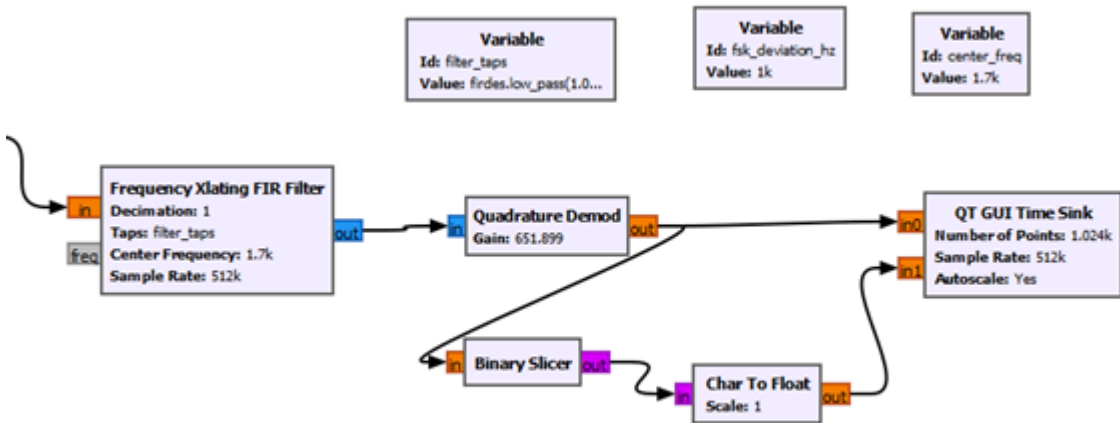
- Sygnał modulowany
 - Typ: float, square
 - Częstotliwość: 50 Hz
- Blok Throttle
- Konwersja danych blokiem Short To Float
- Dwa sygnały wejściowe połączone z rozgałęzionym sygnałem modulowanym za pomocą bloków Multiply
- Blok Add sumujący oba sygnały
- Blok Throttle
- Wizualizacja Scope Sink



Rysunek 8: Modulacja FSK.

Demodulacja

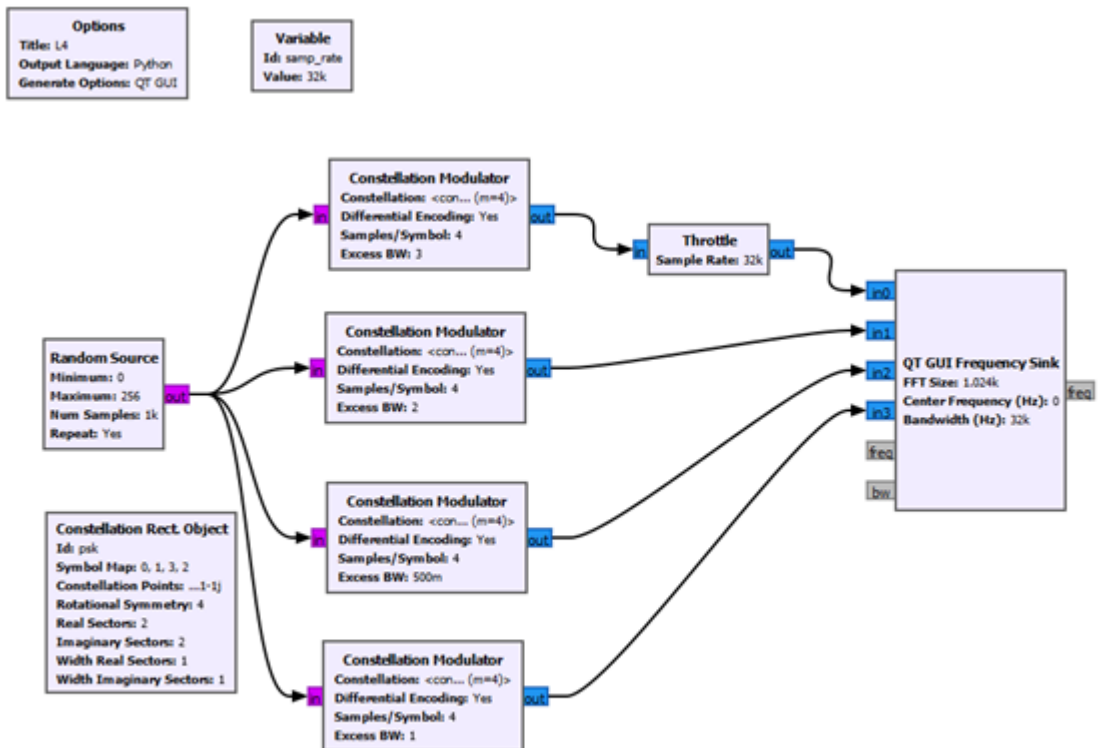
- Blok filtra częstotliwości Xlating FIR Filter
 - Wymaga zdefiniowania zmiennych o podanej wartości:
 - `firdes.low_pass(1.0,samp_rate,1000,400)`
 - $\text{CenterFrequency} = (\text{Frequency_Symbol_1} + \text{Frequency_Symbol_0}) / 2$
 - Gdzie `Frequency_Symbol_1/0` to nazwy zmiennych np. dla bloku `Slicer_ID`
 - $\text{fsk_deviation_hz} = \text{Frequency_Symbol_1} - \text{Frequency_Symbol_0}$
- Blok Quadrature Demod
- Blok Binary Slicer
- Konwersja Char to Float
- Wizualizacja Scope Sink z wejściami
 - Quadrature Demod
 - Sygnału po konwersji typu danych



Rysunek 9: Demodulacja FSK.

Modulacja PSK:

- Źródło sygnału wejściowego
- Blok Constellation Rect Object
- Blok Constellation Modulator
- Blok Throttle
- Wizualizacja Scope Sink/ Frequency Sink



Rysunek 10: Modulacja PSK.

Zadania

- Zaimplementować układ modulacji i demodulacji ASK i zbadać wpływ zmiany parametrów nastawnych na wyjście
- Zaimplementować układ modulacji i demodulacji FSK i zbadać wpływ zmiany parametrów nastawnych na wyjście
- Zaimplementować układ modulacji PSK i zbadać wpływ zmiany parametru Excess_BW na wynik końcowy

Laboratorium 5
Zniekształcenia w kanale transmisyjnym

30.05.2021

Celem laboratorium jest zbadanie wpływu zniekształceń w kanale transmisyjnym. Należy sprawdzić skuteczność przesyłania danych z wykorzystaniem poznanych już mechanizmów kluczenia.

Szczegółowa dokumentacja dostępna jest pod linkiem <https://www.gnuradio.org/docs/>

Do generowania interferencji i szumu w transmitowanym sygnale wykorzystać można blok funkcyjny Channel Model, gdzie jednym z parametrów nastawnych jest wielkość szumu.

Zadania

- Zaprojektować układ modulacji ASK w oparciu o poprzednie laboratoria
 - Z badać wpływ zmiany wartości parametru Noise Voltage na odtwarzany sygnał
 - Usunąć blok Low Pass Filter i sprawdzić działanie programu
 - Usunąć blok Clock Recovery MM I sprawdzić działanie programu
 - Usunąć blok Frequency Xlating FIR Filter i sprawdzić działanie programu
 - Na co wpłynęło usunięcie tych bloków?
- Zaprojektować układ modulacji FSK w oparciu o poprzednie laboratoria
 - Dodać blok Channel Model w odpowiednie miejsce w celu wprowadzenia szumu do odtwarzanego sygnału
 - Z badać zmiany funkcji wykorzystanej do generowanego sygnału na wyniki
 - Z badać wpływ wielkości szumu na otrzymane wykresy
 - Z badać wpływ zmiany wartości Low Frequency, High Frequency na wyniki

Laboratorium 6
Modulator kwadraturowy QPSK

30.05.2021

Celem laboratorium jest opracowanie układu do modulacji cyfrowej z wykorzystaniem konstelacji kodowych. W ramach laboratorium należy zrealizować układ modulacji QPSK. Dodatkowo, należy zbadać wpływ parametrów wejściowych na uzyskane wyniki.

Szczegółowa dokumentacja dostępna jest pod linkiem <https://www.gnuradio.org/docs/>

Blok Constellation Sink – blok do wizualizacji punktów konstelacji

Do zbudowania QPSK układu niezbędne będą bloki:

- Generujące sygnał w postaci bajtów
- Konwertujące bajty na liczby zmiennoprzecinkowe
- Dodające stałą wartość do sygnału
- Blok Root Raised Cosine Filter
- Throttle
- Konwertujące liczby zmiennoprzecinkowe do liczby zespolonej, gdzie jedno wejście pochodzi z jednego wygenerowanego sygnału, natomiast drugie z kolejnego wygenerowanego sygnału
- Wizualizacja konstelacji oraz wizualizacja w dziedzinie częstotliwości

Zadania

- W oparciu o wypisane komponenty zbudować układ QPSK
 - Sample rate – 25k – wykorzystane w Throttle, Root Raised Cosine Filter oraz w Frequency Sink
 - Bit rate – 5k – wykorzystany w Root Raised Cosine Filter
- Zbadać wpływ zmiany parametrów na działanie układu
- Do generowania sygnału wykorzystać
 - Signal Source
 - Random Source
 - Porównać źródła sygnału pod względem otrzymanych wykresów

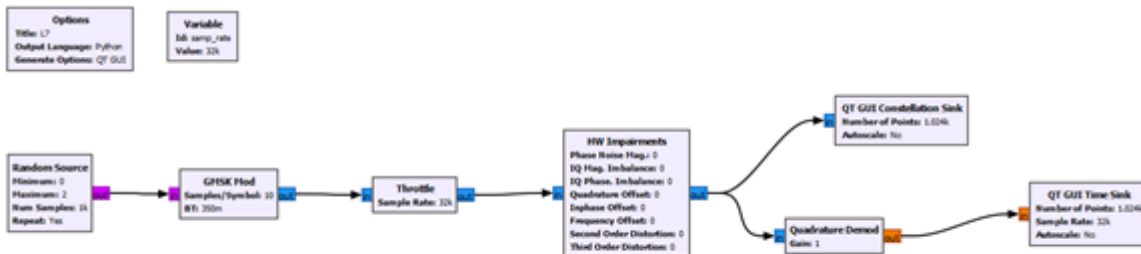
Laboratorium 7

Modulacja GMSK

30.05.2021

Celem laboratorium jest opracowanie systemu modulacji GMSK (Gaussian Minimal Shift Keying). Należy zaprojektować układ i zbadać wpływ parametrów wejściowych na uzyskiwane wyniki. Dodatkowo otrzymane wyniki należy porównać z wybranymi modulacjami w dziedzinie częstotliwości.

Szczegółowa dokumentacja dostępna jest pod linkiem <https://www.gnuradio.org/docs/>



Rysunek 11: Układ modulacji GMSK.

Zadania

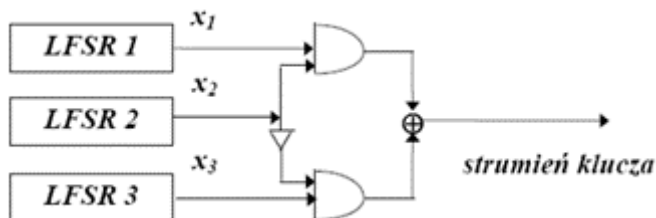
- Zaprojektować układ przedstawiony powyżej
- Wprowadzić zmienne modyfikowalne przez użytkownika do bloku HW Impairments
- Zbadać wpływ zmiany wartości parametrów na uzyskiwane wykresy
- Porównać otrzymane wyniki z modulacjami implementowanymi na wcześniejszych laboratoriach dotyczących kluczowania

Laboratorium 8
System rozpraszania widma metodą FHSS.
Generatory ciągów pseudolowych

30.05.2021

Celem laboratorium jest realizacja systemu rozpraszania widma za pomocą metody FHSS. Dodatkowo, należy zaimplementować generatory ciągów pseudolosowych z wykorzystaniem liniowych rejestrów przesuwających LFSR. LFSR to generatory ciągów binarnych, których struktura posiada dobre właściwości statystyczne.

Szczegółowa dokumentacja dostępna jest pod linkiem <https://www.gnuradio.org/docs/>



Rysunek 12: Budowa generatorów LFSR.

Zadania

- Zaimplementować generator Geffe'go
 - Wykorzystać bloki operacji logicznych
 - Jako źródło sygnału wykorzystać 2 różne rodzaje bloków
 - * Signal Source
 - * Random Source
 - Porównać działanie generatora w zależności od użytego wejścia
 - Do wizualizacji wykorzystać Time Sink
 - Działania przeprowadzać na typie byte
- Wykorzystać wbudowany blok GLFSR i porównać wyniki z zaimplementowanym generatorem Geffe'go

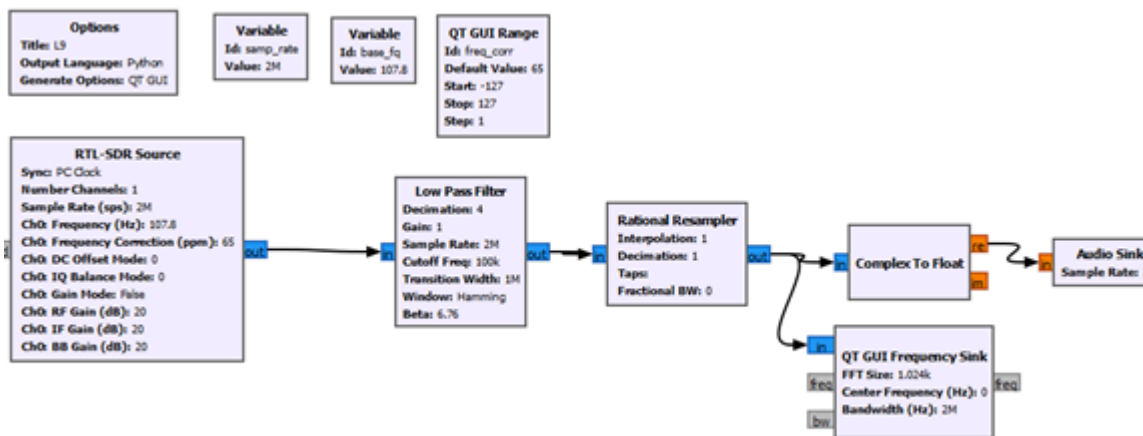
Laboratorium 9

Symulacja toru radiowego

30.05.2021

Celem laboratorium jest symulacja przykładowego toru radiowego z wybranymi schematami modulacji. Dodatkowo, należy dokonać analizy wybranych grup modulacji cyfrowej pod względem uzyskanych wyników.

Szczegółowa dokumentacja dostępna jest pod linkiem <https://www.gnuradio.org/docs/>



Rysunek 13: Podstawowy układ toru radiowego z wykorzystaniem bloku RTL-SDR.

Zadania

- Zaimplementować podstawowy układ toru radiowego
- Zbadać zmiany parametrów wejściowych na wynik
- Do podstawowego układu dodać bloki odpowiadające wybranym dwóm modulacjom cyfrowym
- Porównać wyniki otrzymane dla zaimplementowanych modulacji cyfrowych

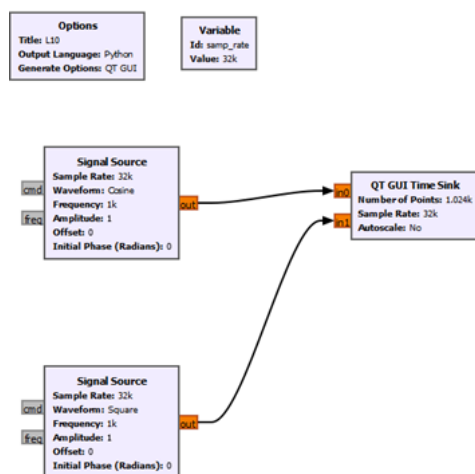
Laboratorium 10
System transmisyjny GNU Radio w języku Python

30.05.2021

Celem laboratorium jest realizacja systemu transmisyjnego zaprojektowanego w GNU Radio, a następnie wygenerowaniu kodu do języka Python i wprowadzeniu zmian w utworzonym modelu.

Szczegółowa dokumentacja dostępna jest pod linkiem <https://www.gnuradio.org/docs/>

W zależności od wykorzystywanego systemu operacyjnego (Windows lub Linux), należy zaimplementować poniższy układ, a następnie wygenerować i zapoznać się z kodem w języku Python.



Rysunek 14: Przykładowy układ z dwoma wejściami.

W przypadku systemu Windows, plik ten utworzy się sam po uruchomieniu zaprojektowanego modelu w programie GnuRadio. Plik .py będzie dostępny pod nazwą wskazaną w opcjach projektu we wskazanej wcześniej lokalizacji.

W przypadku systemu Linux należy po zaimplementowaniu układu nacisnąć przycisk odpowiedzialny za generowanie kodu do języka Python. Plik ten również będzie znajdował się we wcześniej wskazanej lokalizacji, w której zapisywane były dotychczas realizowane projekty.

Zadania

- Zaimplementować układ wskazany powyżej
- Zapoznać się z wygenerowanym kodem w języku Python

- Zmienić parametry nastawne w wygenerowanym kodzie oraz uruchomić go i porównać z wynikami otrzymanymi z programu GNU Radio
 - W przypadku braku modułów w języku Python, doinstalować je
- Następnie wybrać 2 układy realizowane na zajęciach
 - wygenerować kod w języku Python
 - dokonać zmiany wybranych parametrów
 - porównać otrzymane wyniki z uzyskanymi w wyniku działania modeli w GNU Radio