Politechnika Gdańska

Wydział Elektroniki, Telekomunikacji i Informatyki

Katedra Systemów i Sieci Radiokomunikacyjnych

Technika Radia Programowalnego

INSTRUKCJA LABORATORYJNA

Ćwiczenie nr 5

ODBIORNIK OFDM

Opracował: dr inż. Andrzej Marczak

1. Cel ćwiczenia

Celem ćwiczenia jest zapoznanie studentów z budową i działaniem odbiornika cyfrowego łącza radiowego OFDM (ang. Orthogonal Frequency Division Multiplexing) zrealizowanych przy użyciu oprogramowania GNU Radio Companion i uruchomionych na platformie radia programowalnego SDR. Podczas laboratorium są wykorzystywane dwa stanowiska komputerowe.

2. Wprowadzenie

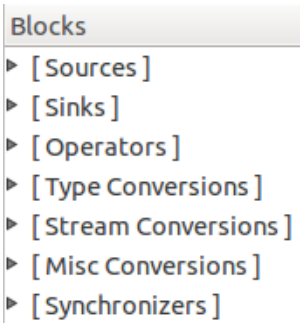
GNU Radio jest pakietem narzędzi programistycznych umożliwiających implementację radia programowalnego przy użyciu bloków przetwarzania sygnałów. Stanowi samodzielne środowisko symulacyjne, zapewniające współpracę z zewnętrznymi modułami RF.

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.

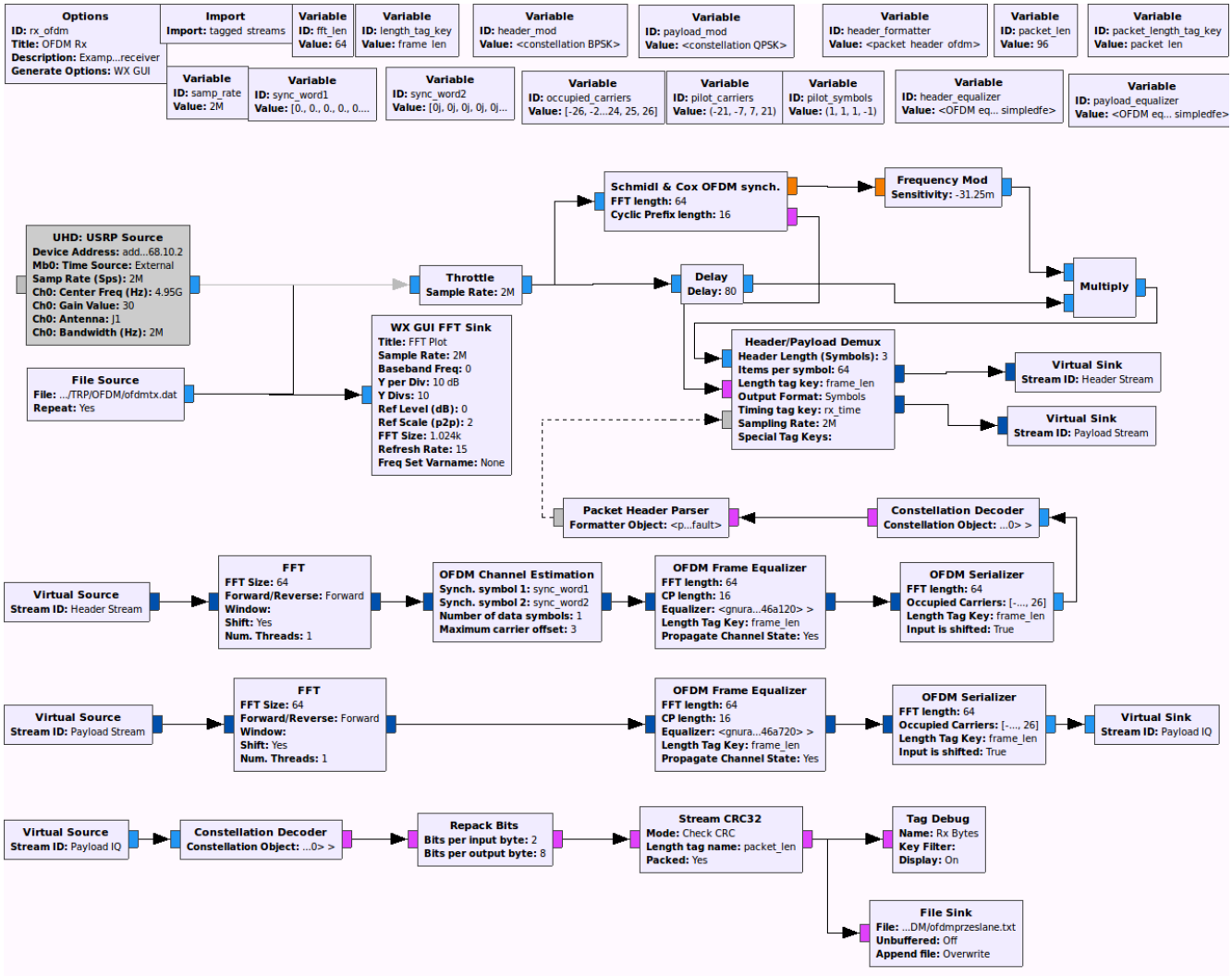
Przeprowadź diagnostykę łącza radiowego, obejmującej urządzenie USRP oraz komputer PC. Następnie skonstruuj schemat przepływu informacji części nadawczej i odbiorczej.



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.

3.1 Odbiornik OFDM

Skonstruuj schemat składający się z następujących elementów: *USRP Source/File Source, Throttle, Schmidl & Cox OFDM synch., Delay, Frequency Mod, Multiply Header/Payload Demux Virtual Sink, FFT, OFDM Channel Estimator, OFDM Frame Equalizer, OFDM Serializer, Constalation Decoder, Packet Header Parser, Repack Bits, File Sink, Tag debug* według schematu zamieszczonego na rysunku 1. Jako plik z sygnałem wejściowym użyj plik **ofdmtx.dat** umieszczony w serwisie enauczanie.



Rysunek 1 Schemat blokowy odbiornika OFDM

Skonfiguruj poszczególne bloki odbiornika zaprezentowanego na rysunku 1:

Variable **fft_len** **64**

Variable **samp_rate** **2000000**

Variable **length_tag_key** **"frame_len"**

Variable **packet_length_tag_key** **"packet_len"**

Variable **packet_len** **96**

Variable **header_mod** **digital.constellation_bpsk()**

Variable **payload_mod** **digital.constellation_qpsk()**

Variable **occupied_carriers** **(range(-26, -21) + range(-20, -7) + range(-6, 0) + range(1, 7) + range(8, 21) + range(22, 27),)**

Variable **pilot_carriers** **((-21, -7, 7, 21,),)**

Variable **pilot_symbols** **((1, 1, 1, -1,),)**

Variable **header_formatter** **digital.packet_header_ofdm(occupied_carriers, n_syms=1, len_tag_key=packet_length_tag_key, frame_len_tag_key=length_tag_key, bits_per_header_sym=header_mod.bits_per_symbol(), bits_per_payload_sym=payload_mod.bits_per_symbol(), scramble_header=False)**

Variable **sync_word1** **[0., 0., 0., 0., 0., 0., 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., -1.41421356, 0., -1.41421356, 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., -1.41421356, 0., -1.41421356, 0., -1.41421356, 0., -1.41421356, 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., 1.41421356, 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., 1.41421356, 0., 1.41421356, 0., -1.41421356, 0., 1.41421356, 0., 1.41421356, 0., 1.41421356, 0., 0., 0., 0., 0., 0.]**

Variable **sync_word2** **[0j, 0j, 0j, 0j, 0j, 0j, (-1+0j), (-1+0j), (-1+0j), (-1+0j), (1+0j), (1+0j), (-1+0j), (-1+0j), (-1+0j), (1+0j), (-1+0j), (1+0j), (1+0j), (1 +0j), (1+0j), (1+0j), (-1+0j), (-1+0j), (-1+0j), (-1+0j), (-1+0j), (1+0j), (-1+0j), (-1+0j), (1+0j), (-1+0j), 0j, (1+0j), (-1+0j), (1+0j), (1+0j), (1+0j), (-1+0j), (1+0j), (1+0j), (1+0j), (-1+0j), (1+0j), (1+0j), (1+0j), (1+0j), (-1+0j), (1+0j), (-1+0j), (-1+0j), (-1+0j), (1+0j), (-1+0j), (1+0j), (-1+0j), (-1+0j), (-1+0j), (-1+0j), 0j, 0j, 0j, 0j, 0j]**

Variable **header_equalizer** **digital.ofdm_equalizer_simpledfe(fft_len, header_mod.base(), occupied_carriers, pilot_carriers, pilot_symbols)**

Variable **payload_equalizer** **digital.ofdm_equalizer_simpledfe(fft_len, payload_mod.base(), occupied_carriers, pilot_carriers, pilot_symbols, 1)**

Throttle block:

Type: **Complex**

Sample Rate **samp_rate**

Vec Length **1**

Schmidl & Cox OFDM synch block:

FFT length: **fft_len**

Cyclic Prefix length: **fft_len/4**

Preamble Carrriers: **Odd**

Delay block:

Delay: **fft_len+fft_len/4**

Frequency Mod block**:**

Sensitivity: **-2.0/fft_len**

Header/Payload Demux block:

Header Length (Symbols): **3**

Items per symbol: **fft_len**

Guard Interval (items): **fft_len/4**

Length tag key: **length_tag_key**

Output Format: **Symbols**

Timing Tag key: **“rx_time”**

Sampling Rate: **samp_rate**

Special Tag Keys: **()**

FFT blocks**:**

Input Type: **Complex**

FFT Size: **fft_len**

Forward/Reverse: **Forward**

Window: **()**

Shift: **Yes**

Num. Threads: **1**

OFDM Channel Estimation block:

Synch. symbol 1: **sync_word1**

Synch. symbol 2: **sync_word2**

Number of data symbols	**1**
Maximum carrier offset	**3**
Force One Synchr.	**No**

OFDM Frame Equalizer block (Header):

FFT length:	**fft_len**
CP length:	**fft_len/4**
Equalizer:	**header_equalizer.base()**
Length Tag Key:	**length_tag_key**
Propagate Channel State:	**Yes**
Fixed frame length:	**1**

OFDM Frame Equalizer block (Payload):

FFT length:	**fft_len**
CP length:	**fft_len/4**
Equalizer:	**payload_equalizer.base()**
Length Tag Key:	**length_tag_key**
Propagate Channel State:	**Yes**
Fixed frame length:	**0**

OFDM Serializer block (Header):

FFT length:	**fft_len**
Occupied Carriers:	**occupied_carriers**
Length Tag Key:	**length_tag_key**
Symbols skipped:	**0**
Input is shifted	**True**

OFDM Serializer block (Header):

FFT length:	**fft_len**
Occupied Carriers:	**occupied_carriers**
Length Tag Key:	**length_tag_key**
Packet Length Tag Key:	**packet_length_tag_key**
Symbols skipped:	**1**
Input is shifted	**True**

Constelation Decoder (Header):

Constelation Object: **header_mod.base()**

Constelation Decoder (Payload):

Constelation Object: **payload_mod.base()**

Packet Header Parser block:

Formatter Object: **header_formatter.base()**

Repack Bits block:

Bits per input byte: **payload_mod.bits_per_symbol()**

Bits per output byte: **8**

Length Tag Key: **packet_length_tag_key**

Packet Alignment: **Output**

Endianness: **LSB**

Stream CRC32 block:

Mode **Check CRC**

Length tag name **packet_length_tag_key**

Packet **Yes**

Type: **Byte**

Vector Length: **1**

Packet Length: **packet_len**

Lenght Tag Key **length_tag_key**

Packet Header Generator block:

Formatter Object **header_formatter.formatter()**

Repack Bits block:

Bits per input byte **8**

Bits per output byte **payload_mod.bits_per_symbol()**

Length Tag Key **length_tag_key**

Packet Alignment **Input**

Chunks to Symbol for Header bits block:

Input Type **Byte**

Output Type **Complex**

Symbol Table **header_mod.points()**

Dimension **1**

Num Ports **1**

Chunks to Symbol for Payload bits block:

Input Type **Byte**

Output Type **Complex**

Symbol Table **payload_mod.points()**

Dimension **1**

Num Ports **1**

Tagged Stream Mux block:

IO Type **Complex**

Number of inputs **2**

Length tag names **length_tag_key**

Vector Length **1**

Tags: Preserve head position **0**

OFDM Carrier Allocator block:

FFT length **fft_len**

Occupied Carriers **occupied_carriers**

Pilot Carriers **pilot_carriers**

Pilot Symbols	**pilot_symbols**
Sync Words	**(sync_word1, sync_word2)**
Length Tag Key	**length_tag_key**

FFT block:

Input Type	**Complex**
FFT Size	**fft_len**
Forward/Reverse	**Reverse**
Window	**()**
Shift	**Yes**
Num. Threads	**1**

OFDM Cyclic Prefixer block:

FFT Length	**fft_len**
CP Length	**fft_len/4**
Rolloff	**rolloff**
Length Tag Key	**length_tag_key**

Multiply Const block:

IO Type	**Complex**
Constant	**0.05**
Vec Length	**1**

Tag Gate block:

Item Type	**Complex**
Vec Length	**1**
Propagate_tags	**No**

Zapisz oraz uruchom projekt. Zaobserwuj zmiany w wykresie widma i przebiegu czasowego odbieranego sygnału dla pliku **ofdmtx.dat** oraz obejrzyj wynik symulacji transmisji pliku tekstowego (plik wyjściowy przeslane.txt).

Zrealizuj symulację transmisji pliku tekstowego pomiędzy dwoma komputerami. Uruchom swój nadajnik OFDM zrealizowany w ramach ćwiczenia 4, wprowadzając jako plik wejściowy swój plik tekstowy z dowolną zawartością. Następnie wyjściowy plik z danymi swojego nadajnika wprowadź do odbiornika i obserwuj plik z danymi wyjściowymi z odbiornika. Porównaj jego zawartość z plikiem wejściowym nadajnika.

4. Literatura

1. http://gnuradio.org/redmine/projects/gnuradio/wiki.

2. Notatki z wykładów.