# Technologia radia programowalnego w transporcie szynowym

Andrzej MARCZAK[1]

**Streszczenie**
Technologia radia programowalnego umożliwia wykonanie urządzeń pracujących w różnego rodzaju analogowych i cyfrowych systemach łączności radiowej. W artykule zaprezentowano zagadnienia dotyczące koncepcji przeprowadzenia radia programowalnego. Zaprezentowano oprogramowanie GNU Radio, służące do realizacji operacji przetwarzania sygnałów w torach nadawczo-odbiorczych urządzeń zrealizowanych w technologii SDR. Przedstawiono również środowisko graficzne GNU Radio Companion, które ułatwia projektowanie i oprogramowanie radia programowalnego. Przykład aplikacji nadajnika i odbiornika z wykorzystaniem oprogramowania GNU Radio, pokazuje łatwość, z jaką można tworzyć urządzenia nadawczo-odbiorcze w technologii SDR. Ta aplikacja została uruchomiona i pracowała na przykładowej platformie sprzętowej SDR, realizując transmisję z urządzeniami wykonanymi w dotychczas stosowanej technologii. Schemat i budowę tej platformy sprzętowej SDR opisano w artykule.
Technologia radia programowalnego umożliwia szybką zmianę właściwości sprzętu i dostosowanie go do aktualnych zastosowań. Technologia SDR umożliwia wykorzystanie tego samego sprzętu, z odpowiednim oprogramowaniem, w różnych, często odmiennych zastosowaniach np. w transporcie szynowym, drogowym czy zastosowaniach specjalnych. Dodatkowo możliwość łatwego, programowego upgrade'u właściwości sprzętu pozwala wydłużyć czas wykorzystania urządzeń.

**Słowa kluczowe**: radio programowalne, SDR, GNU Radio, GNU Radio Companion

## 1. Wstęp

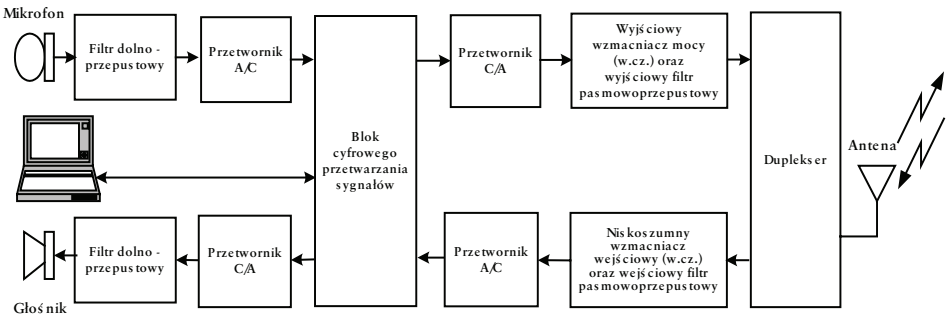
Szybki rozwój systemów cyfrowej radiokomunikacji ruchomej wywołał potrzebę nieustannego opracowywania nowych rozwiązań urządzeń, które mogłyby sprostać zapotrzebowaniu użytkowników na nowe usługi transmisji danych o dużych przepływnościach. Ponadto różnorodność standardów systemów łączności radiowej w transporcie, szczególnie w transporcie szynowym oraz duża mobilność

[1] Dr inż.; Politechnika Gdańska; e-mail: amarczak@eti.pg.gda.pl.

ich użytkowników, powodują, że pożądane jest opracowanie wielosystemowego terminala ruchomego, zdolnego do współpracy z systemami radiokomunikacyjnymi działającymi w różnych standardach i zapewniającego bezpieczeństwo kryptograficzne transmisji. Stało się to powodem podjęcia prac nad koncepcją radia programowalnego (ang. SDR – *Software Defined Radio*), której celem jest zastąpienie członów nadawczo-odbiorczych, realizowanych sprzętowo, w jednym standardzie, przez możliwie uniwersalny *hardware*, w którym występują człony wielkiej częstotliwości nadajnika i odbiornika, szerokopasmowe przetworniki C/A i A/C oraz procesor sygnałowy i inne układy programowalne [8, 9,]. Wówczas funkcje nadawczo-odbiorcze mogą być realizowane programowo przez procesor sygnałowy [3, 7].

## 2. Architektura urządzeń zrealizowanych w technice radia programowalnego

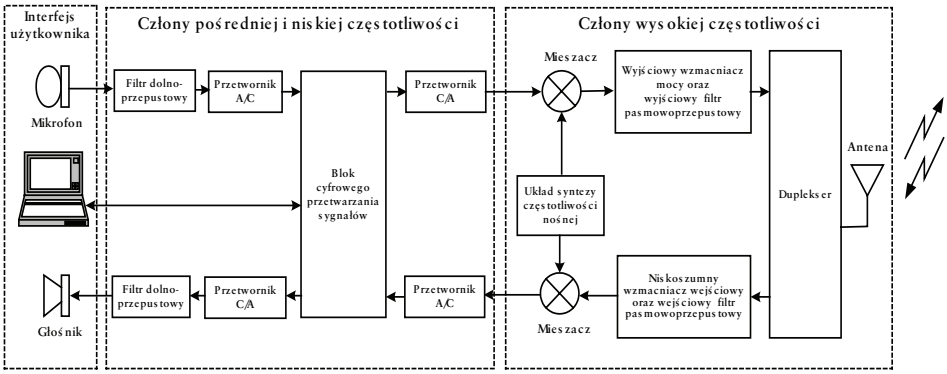
Architekturę programowalnego urządzenia radiowego przedstawiono na rysunku 1. Symbol komputera reprezentuje źródło i / lub obiekt przeznaczenia dowolnych sygnałów cyfrowych z pominięciem sygnałów mowy [7].




Rys. 1. Ogólna architektura programowalnego transceivera radiowego [7]

Analogowy wzmacniacz mocy i filtr pasmowoprzepustowy w nadajniku poprzedza przetwornik C/A, do którego są dostarczane sygnały cyfrowe z bloku cyfrowego przetwarzania sygnałów, w którym są wykonywane m.in. funkcje kodowania i modulacji, a niskoszumny wzmacniacz wejściowy i filtr pasmowoprzepustowy w odbiorniku przekazują analogowe sygnały odebrane poprzez przetwornik A/C do tego samego bloku cyfrowego przetwarzania sygnałów, m.in. w celu detekcji i dekodowania. Przy współczesnym poziomie technologicznym, taka realizacja programowalnego urządzenia radiowego jest na razie niewykonalna [8, 9]. Ograniczenia te wynikają przede wszystkim z braku przetworników A/C i C/A

o wymaganej szybkości i dynamice przetwarzania oraz ograniczonej szybkości przetwarzania dostępnych procesorów sygnałowych. W tej sytuacji obiecująca wydaje się architektura, w której przetwarzanie A/C i C/A odbywa się w paśmie pośredniej częstotliwości, co przedstawiono na rysunku 2 według [4].




Rys. 2. Architektura programowalnego transceivera z przetwarzaniem A/C i C/A w członie pośredniej częstotliwości [4]

Blok cyfrowego przetwarzania sygnałów w urządzeniu SDR powinien wykonywać następujące funkcje toru nadawczo-odbiorczego [2]:

- obsługę interfejsu użytkownika,
- kodowanie i dekodowanie źródłowe,
- kodowanie i dekodowanie kanałowe,
- szyfrację i deszyfrację,
- przeplot i rozplot bitowo-blokowy,
- cyfrową filtrację sygnału,
- modulację i demodulację,
- synchronizację.

W przypadku zastosowania w interfejsie radiowym techniki bezpośredniego rozpraszania widma, blok cyfrowego przetwarzania sygnałów powinien dodatkowo wykonywać następujące funkcje:

- ortogonalizację i deortogonalizację sygnałów,
- rozpraszanie i skupianie widma sygnałów,
- dynamiczne sterowanie mocą sygnałów wyjściowych,
- odbiór wielodrogowy i wspólny sygnałów wielu użytkowników (ang. *multi-user detection*).

## 3. Oprogramowanie GNU Radio

GNU Radio to zestaw bezpłatnych narzędzi do tworzenia oprogramowania wykonującego funkcje nadawczo-odbiorcze zgodnie z koncepcją radia programowalnego. Może on być wykorzystywany do przetwarzania sygnału warstwy fizycznej dzięki współpracy z łatwo dostępnym zewnętrznym sprzętem (platformą SDR) lub bez urządzeń w trybie symulacji. Jest on powszechnie stosowany do poznawania, budowania i wdrażania radia programowalnego zarówno w zastosowaniach biznesowych, jak i akademickich. GNU Radio jest przede wszystkim opracowany z wykorzystaniem systemu operacyjnego GNU / Linux, ale obecnie inne systemy operacyjne (Mac OS i Windows) są również obsługiwane. W GNU Radio, system radiowy jest reprezentowany jako graf przepływu sygnałów, w którym wierzchołki wykresu są znane jako bloki przetwarzania sygnału, a strzałki wskazują połączenia między dwoma blokami. Przepływ danych jest możliwy w jednym kierunku od źródła sygnału dla jednego lub większej liczby węzłów sygnałów. Aplikacje GNU Radio są najczęściej napisane w języku programowania Python. Gdy wydajność przetwarzania sygnału jest krytyczna, stosuje się język programowania C++, np. za pomocą procesora z operacjami zmiennoprzecinkowymi. Deweloper jest więc w stanie urzeczywistnić system radiowy o dużej wydajności, wykonujący transmisję w czasie rzeczywistym za pomocą prostego w obsłudze środowiska rozwoju aplikacji [2].

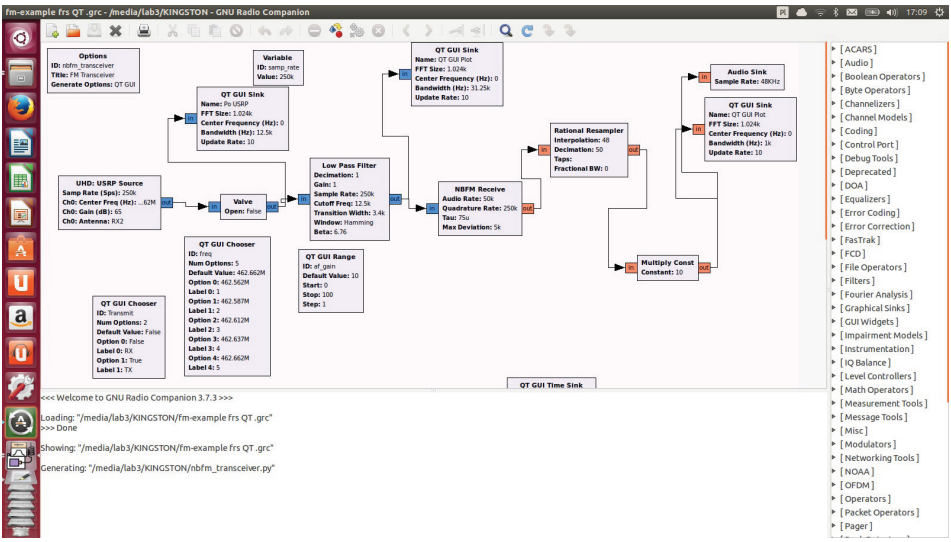
GNU Radio umożliwia rozwój algorytmów przetwarzania sygnału z wykorzystaniem wcześniej zapisanych lub wygenerowanych danych, unikając potrzeby korzystania z rzeczywistego sprzętu radiowego [2]. GNU Radio jest pakietem oprogramowania do przetwarzania sygnału, który jest rozpowszechniany na zasadach licencji GNU *General Public License* (GPL). Cały kod jest chroniony prawem autorskim *Free Software Foundation*. Celem jest udostępnienie użytkownikom oprogramowania umożliwiającego wykorzystanie widma fal radiowych. Jak w przypadku wszystkich urządzeń pracujących w technice radia programowalnego SDR, kluczowym elementem jest możliwość rekonfiguracji właściwości i parametrów. Zamiast kupować wiele kosztownych urządzeń nadawczo-odbiorczych, można wykorzystać pojedyncze urządzenie, którego pracą zarządza oprogramowanie przetwarzające sygnał. Obecnie można wykorzystać gotowe oprogramowanie do przetwarzania sygnałów niektórych standardów systemów radiowych (np. systemu komórkowego GSM) lub samodzielnie napisać niezbędne oprogramowanie [2].

Oprogramowanie GNU Radio Companion jest interfejsem graficznym do tworzenia aplikacji w GNU Radio. Jest to nakładka do bibliotek GNU Radio, która służy do przetwarzania sygnałów. Pozwala na łatwe tworzenie wykresów przepływu GNU Radio, umożliwiając naukę podstaw środowiska GNU Radio. GNU

Radio Companion pozwala na konfigurację parametrów wejściowych bloków, które są wykorzystywane przez kod źródłowy każdego bloku, w celu wygenerowania przepływu sygnału. Wykorzystując bloki graficznej wizualizacji sygnału można kontrolować pracę urządzenia na każdym etapie przetwarzania sygnałów. Można wyróżnić cztery główne rodzaje bloków [2]:

1. **Bloki źródeł sygnałów (ang. *Source Blocks*)**
   Ich główną funkcją jest generowanie sygnału wyjściowego za pomocą kilku konfigurowalnych parametrów wejściowych. Z tego powodu bloki te nie mają portu wejściowego. Istnieje wiele rodzajów źródeł, w zależności od liczby portów wyjściowych, typu danych, długości wektora itp.
2. **Bloki zakończenia przepływu sygnałów (ang. *Sink Blocks*)**
   W tym przypadku, bloki nie mają portu sygnału wyjściowego. Bloki te odbierają sygnał wejściowy o określonym typie danych i długości. Następnie z wykorzystaniem pewnych parametrów wejściowych, sygnał wejściowy jest przechowywany w wektorach lub w pliku.
3. **Bloki przetwarzania (ang. *Operation Blocks*)**
   W blokach tych używa się konfigurowalnej liczby sygnałów wejściowych o konfigurowalnych typach danych, w celu wytworzenia pewnej liczby sygnałów wyjściowych o określonych typach danych. W blokach, przy użyciu określonych parametrów wejściowych, wykonywane są operacje na próbkach sygnału. Operacjami tymi mogą być, np. modulacja lub demodulacja, kodowanie, filtracja sygnałów itd.
4. **Bloki wizualizacji (ang. *Visualization Blocks*)**
   Bloki te mogą być sklasyfikowane jako typ bloku wyjściowego, w którym jest prezentowana graficzna postać sygnałów wejściowych. W tej grupie bloków, można wymienić, np. bloki prezentujące dane w funkcji czasu lub częstotliwości. Podstawowe funkcje każdego bloku są określone przez kod źródłowy napisany w językach programowania Python lub C++.

Na rysunku 3 zaprezentowano przykład okna uruchomionej aplikacji GNU Radio Companion z fragmentem grafu przepływu sygnałów toru odbiorczego. Po prawej stronie okna roboczego widać przykłady rodzajów gotowych bloków, które można wykorzystać realizując graf przepływu sygnałów. Nad oknem roboczym znajdują się ikony realizujące funkcje zapisu i odczytu danych oraz funkcje generowania kodu źródłowego w języku Python z grafu przepływu sygnałów (ang. *Generate the flow graph*) i uruchamiania tego wygenerowanego kodu źródłowego (ang. *Execute the flow graph*).



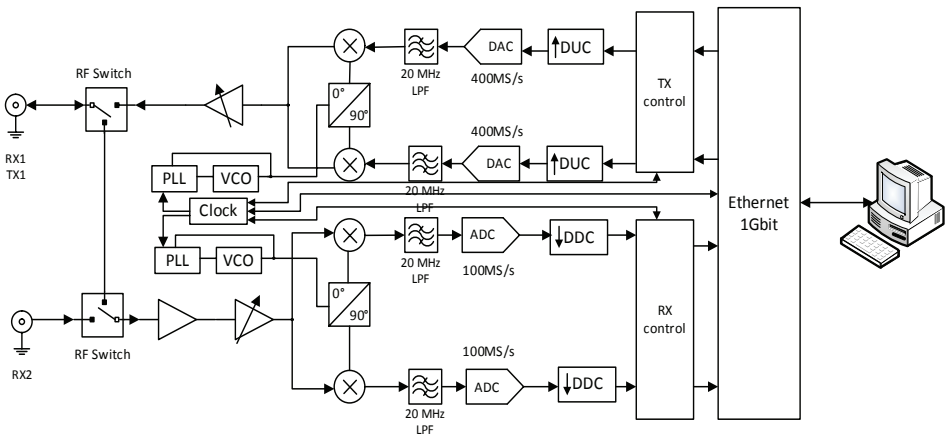

Rys. 3. Przykład okna z uruchomionym oprogramowaniem GNU Radio Companion

## 4. Platforma sprzętowa USRP

Platformę sprzętową NI USRP (ang. *National Instruments Universal Software Radio Peripheral*) zaprojektowano jako urządzenie o stosunkowo niskim koszcie, służące do uruchamiania aplikacji przygotowanych w środowisku GNU Radio. Platformę NI USRP przedstawiono na rysunku 4.


Rys. 4. Widok platformy sprzętowej radia programowalnego NI USRP [5]

Platforma NI USRP 2920 charakteryzuje się tym, że nie może pracować samodzielnie, bez połączenia z komputerem PC. W tym rozwiązaniu większość operacji związanych z przetwarzaniem sygnałów odbywa się w komputerze PC, z którego próbki sygnału docierają przez interfejs 1 Gbit Ethernet do urządzenia NI USRP (w trybie nadawania). Jeżeli platforma sprzętowa pracuje w trybie odbioru, próbki sygnału tym samym interfejsem są transmitowane do komputera, w którym odbywają się pozostałe operacje. Platforma NI USRP może realizować transmisję w pełnym dupleksie.



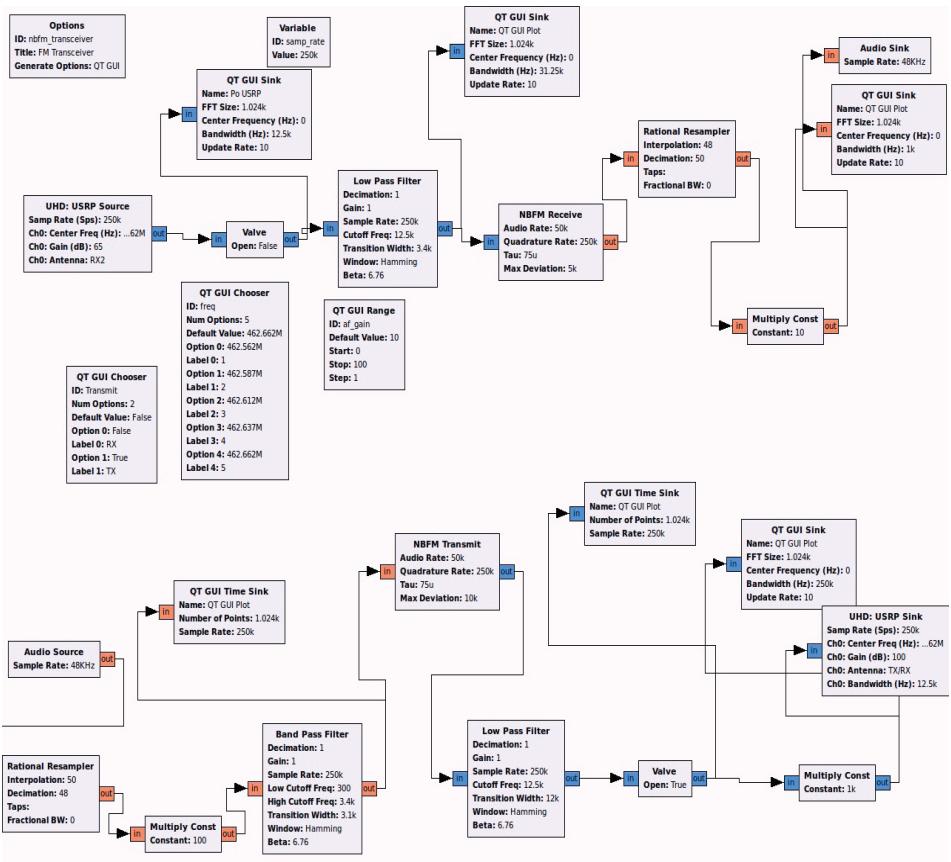

Rys. 5. Schemat blokowy platformy sprzętowej NI USRP [5]

Schemat blokowy platformy sprzętowej NI USRP zaprezentowano na rys.5. Platforma może pracować w paśmie częstotliwości 50 MHz÷2.2 GHz z szerokością kanału do 20 MHz. W urządzeniu zastosowano matrycę FPGA Spartan 3A DSP, 16 bitowy, dwukanałowy przetwornik cyfrowo-analogowy (400 MS/s), 14 bitowy, dwukanałowy przetwornik analogowo-cyfrowy (100 MS/s) oraz 1 MB szybkiej pamięci SRAM [6]. Istnieje możliwość połączenia dwóch identycznych urządzeń specjalnym interfejsem MIMO expansion do realizacji transmisji w trybie MIMO (ang. *Multiple Input Multiple Output*). Niektóre urządzenia NI USRP 29xx mogą być dodatkowo wyposażone w odbiornik GPS w celu poprawy synchronizacji [5].

Zaletą urządzeń NI USRP jest możliwość współpracy zarówno z oprogramowaniem komercyjnym (np. Matlab-Simulink, LabView), jak i z oprogramowaniem dostępnym na licencji GPL (np. GNU Radio). Urządzenie może być wykorzystane do realizacji interfejsów radiowych stosowanych w nowoczesnych systemach radiowych. Interfejsy te wymagają dużej mocy obliczeniowej, dlatego konieczne jest wykorzystanie komputera PC wyposażonego w szybki procesor wielordzeniowy, dużą pojemność pamięci RAM i szybki dysk HDD.

# 5. Przykład aplikacji wykonującej funkcje transceivera z modulacją częstotliwości w technice radia programowalnego

Wykorzystując oprogramowanie GNU Radio i platformę sprzętową radia programowalnego USRP, można wykonać u*kłady transceivera* w większości obecnie stosowanych systemów łączności radiowej. Ograniczeniem są parametry radiowe platformy USRP i moc obliczeniowa zastosowanego komputera, w którym są przetwarzane sygnały. Na rysunku 6 przedstawiono graf przepływu sygnałów dla aplikacji realizującej *transceiver* pracujący z modulacją częstotliwości w paśmie, w którym pracuje standard łączności FRS (ang. *Family Radio Service*) [1].

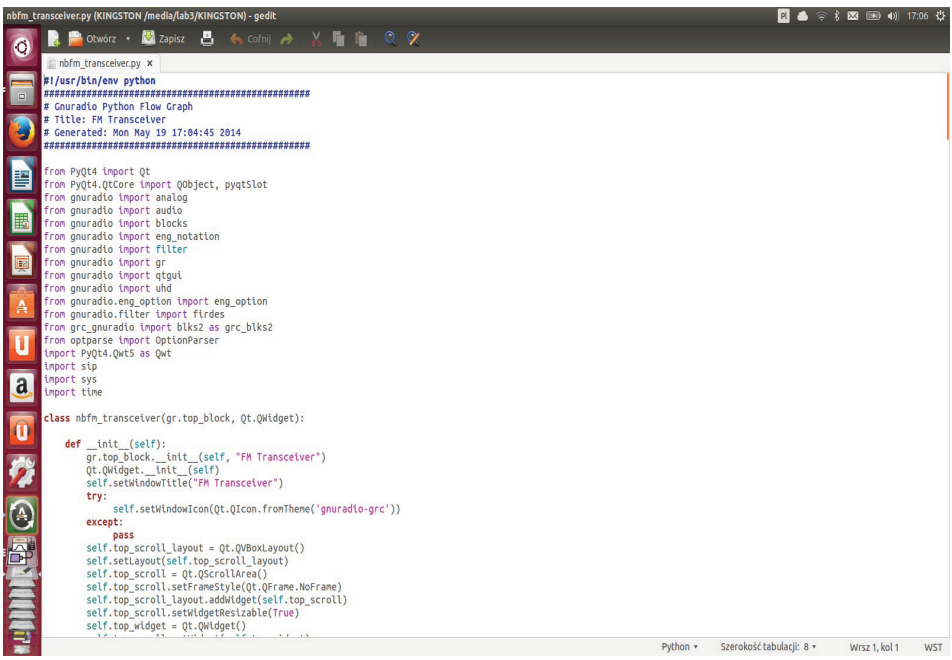



Rys. 6. Graf przepływu sygnałów dla aplikacji transceivera FM

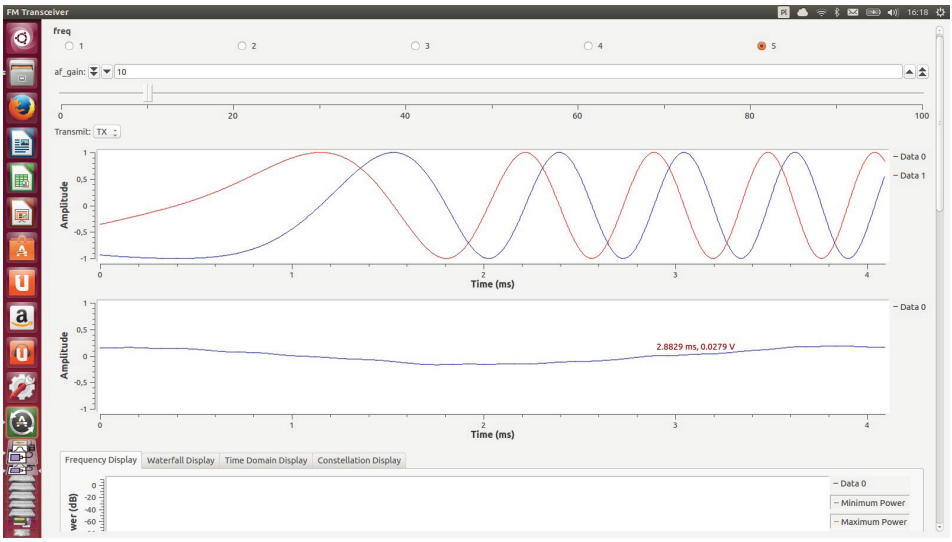
Jest to standard łączności w trybie *walkie talkie* działający w Stanach Zjednoczonych i korzystający z pasma częstotliwości (462,5625 MHz–467,7125 MHz).

W górnej części rysunku przedstawiono graf przepływu dla odbiornika, dolna część obrazuje nadajnik. W realizacji transceivera wykorzystano bloki modulatora i demodulatora częstotliwości oraz filtrów dolnoprzepustowych i pasmowo-przepustowych. Oprócz bloków źródeł sygnału (*Audio source* i *USRP Source*) wykorzystano węzły kończące transfer sygnału w grafie (Audio Sink i USRP Sink).

Dodatkowo w grafie wykorzystano bloki wizualizacji częstotliwości i czasu. Blok o nazwie QT GUI *Chooser* umożliwia wybór częstotliwości, na której będzie realizowana transmisja, a blok QT GUI *Range* pozwala wybrać poziom głośności odbieranego dźwięku. Po połączeniu wszystkich bloków i ich skonfigurowaniu, jest możliwe wygenerowanie kodu źródłowego w języku Python. W tym celu należy uruchomić funkcję *Generate the flow graph*. Wynikiem będzie kod źródłowy w języku Python, którego fragment przedstawia rysunek 7. Następnie należy użyć funkcji *Execute the flow graph*, która spowoduje połączenie z urządzeniem USRP i uruchomienie zrealizowanej aplikacji. Rysunki 8 i 9 przedstawiają okno z uruchomioną aplikacją transceivera z modulacją częstotliwości.

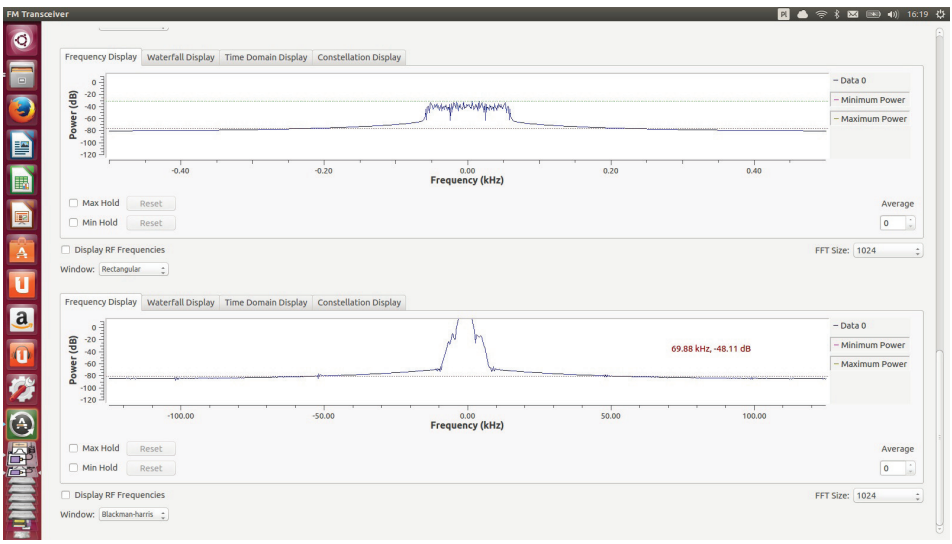



Rys. 7. Fragment kodu źródłowego aplikacji SDR w języku programowania Python




Rys. 8. Widok uruchomionej aplikacji SDR

Na rysunku 8 widać możliwość wyboru częstotliwości transmisji (*freq*), regulacji głośności (*af_gain*), kierunku transmisji (TX lub RX). Widać również wykresy czasu transmitowanych sygnałów. Rysunek 9 przedstawia wykresy sygnałów częstotliwości.




Rys. 9. Widok uruchomionej aplikacji SDR [cd.]

## 6. Podsumowanie

Technologia radia programowalnego umożliwia szybką zmianę właściwości sprzętu i dostosowanie go do aktualnych zastosowań. Technologia SDR umożliwia wykorzystanie tego samego sprzętu, z odpowiednim oprogramowaniem, w różnych, często odmiennych zastosowaniach, np. w transporcie szynowym, drogowym czy zastosowaniach specjalnych. Dodatkowo, możliwość łatwego, programowego *upgrade'u* właściwości sprzętu pozwala wydłużyć czas wykorzystania urządzeń.

Oprogramowanie GNU Radio wraz z interfejsem graficznym GNU Radio Companion jest doskonałym narzędziem do tworzenia aplikacji w technologii radia programowalnego. Umożliwia korzystanie z gotowych bloków przetwarzania sygnałów, np. modulatorów i demodulatorów, jak również samodzielne tworzenie bloków, które są wymagane do wykonania konkretnego interfejsu radiowego. Oprogramowanie to doskonale współpracuje z platformą sprzętową NI USRP. Możliwa jest również współpraca tego oprogramowania z innymi urządzeniami, które mogą pracować, np. tylko jako programowalne odbiorniki (niektóre rodzaje tanich odbiorników telewizji cyfrowej DTV podłączanych do komputera poprzez port USB i opartych na układzie scalonym RTL2832U). Duża popularność środowiska GNU Radio powoduje, że coraz więcej producentów platform sprzętowych radia programowalnego udostępnia sterowniki umożliwiające współpracę sprzętu z oprogramowaniem GNU Radio.

Opisana platforma NI USRP pozwala opracowywać urządzenia w technologii SDR. Platforma ta nie może pracować jako samodzielne urządzenie. Jest to zarówno wada jak i zaleta. Podczas badań interfejsów radiowych, oprócz urządzenia USRP musi być zawsze podłączony do niego komputer PC lub laptop, co może być problemem, jeśli badania interfejsów radiowych są wykonywane w terenie. Jednak dzięki konieczności połączenia z komputerem, można w tym rozwiązaniu w łatwy sposób zwiększyć dostępną moc obliczeniową przez rozbudowę lub łatwą wymianę zastosowanego komputera PC. Oprogramowanie testowane i uruchomione na takiej platformie sprzętowej, w zaprezentowanej wersji laboratoryjnej, może być przeniesione i uruchomione na urządzeniu wykonanym w wersji komercyjnej.

Przedstawiony w artykule przykład aplikacji transceivera z modulacją częstotliwości pokazuje, jak w łatwy sposób można wykonać tor nadawczo-odbiorczy z wykorzystaniem oprogramowania GNU Radio. Jest to prosty przykład układu radiowego, jednak w podobny sposób można wykonać układy transceiverów realizujące transmisję cyfrową w technice TDMA, CDMA lub OFDM wykorzystującej modulację cyfrową i nowoczesnych metod kodowania kanałowego i zabezpieczenia kryptograficznego. Możliwa jest więc programowa implementacja

istniejących interfejsów systemów łączności radiowej zarówno analogowych jak i cyfrowych, jak również projektowanie nowych rozwiązań do przyszłego zastosowania w radiokomunikacji.

## Literatura

1. *Family Radio Service* (FRS), dostępny na WWW: http://www.fcc.gov/encyclopedia/family-radio-service-frs.
2. *GNURADIO*, dostępny na WWW: http://gnuradio.org.
3. Harada H., Prasad R.: *Simulation and Software Radio for Mobile Communication*, Artech House, London 2002.
4. Marczak A., Katulski R. J., Stefański J.: *Technika radia programowalnego*, Przegląd Telekomunikacyjny, Nr 10/2004.
5. *NI USRP-29xx Datasheet Universal Software Radio Peripherals*, National Instruments.
6. *NI USRP™-2920 Device specification 50 MHz to 2.2 GHz Tunable RF Transceiver*, National Instruments.
7. Stefański J., Gajewski S., Marczak A.: *Radio rekonfigurowalne programowo w systemie UMTS*, Elektronik nr 11/2001.
8. Wesołowski K., Krenz R.: *Software Radio – Technologia przyszłych systemów radiokomunikacji ruchomej*: Materiały Konferencyjne Krajowej Konferencji Radiokomunikacji Radiofonii i Telewizji KKRRiT'2000, Poznań 2000.
9. Wesołowski K.: *Koncepcja Software Radio i jej znaczenie dla rozwoju radiokomunikacji ruchomej*, Materiały Konferencyjne Krajowej Konferencji Radiodyfuzji i Radiokomunikacji KKRR'1998, Poznań 1998.

# Software Defined Radio Technology in Rail Transport

**Summary**

Software Defined Radio technology enables the realization of devices in various types of analog and digital radiocommunication systems. This paper presents the issues concerning the implementation of the SDR. GNU Radio software The paper presents GNU Radio software used to implement the signal processing implemented in SDR transceiver devices. It also presents a GNU Radio Companion graphical environment, which facilitates the design and implementation of software for the implementation of the SDR. Example of the transmitter and receiver applications implemented using the GNU Radio software, shows the ease with which you can create transceivers in SDR technology. This application has been launched and worked on the SDR hardware platform realizing connection with devices made in the existing technology. Scheme and construction of the SDR hardware platform are also described in the paper.

SDR technology allows you to quickly change the properties of hardware equipment and adapting it to the current application. This technology allows you to use the same hardware equipment, with appropriate software, in different, often diverse applications e.g. in rail or road transport and special applications. In addition, the ability to easily, software upgrade equipment properties allows increase the time of use the hardware equipment.

**Keywords**: SDR, GNU Radio, GNU Radio Companion

# Технология программируемого радио в железнодорожном транспорте

**Резюме**

Технология программируемого радио дает возможность создания устройств работающих в разного рода аналоговых и цыфровых системах радиосвязи. В данном труде представлена тематика касающаяся концепции создания программируемого радио. Представлено программное обеспечение GNU Radio, которое служит для осуществления операции преобразования сигналов в передаваемо-приемных цепях устройств разработанных на основе технологии SDR. Представлено также графическую среду GNU Radio Companion, которая помогает реализовывать и проектировать программное обеспечение для программируемого радио. Пример апликации передатчика и приемника разработанных с использованием программного обеспечения GNU Radio, показывает с какой простотой можна создавать передаваемо-приемные устройства в технологии SDR. Эта Апликация применялась и работала на примерной платформе устройств SDR осуществляя трансмиссию с устройствами, которые были разработаны на основе до настоящего времени применямой технологии. Схема и структура этой платформы устройства SDR также описана в реферате.

Технология программируемого радио дает возможность быстро изменять свойства оборудования и его приспособления к применению сегодняшнего дня. Технология SDR дает возможность использования того же оборудования, с соответственным программным обеспечением, в разных, очень часто отличающихся применениях, например в железнодорожном транспорте, автодорожном или в специальных применениях. Дополнительно возможность простого, программируемого upgrade'а свойств оборудования, дает возможность увеличить время использования оборудования.

**Ключевые слова**: программируемое радио, SDR, Радио GNU, Радио GNU Companion