

Jan SZYMENDERSKI*

KOMPUTEROWE PRZETWARZANIE SYGNAŁÓW RADIOWYCH

Praca dotyczy cyfrowego przetwarzania sygnałów o częstotliwościach radiowych. Opracowano i zrealizowano program komputerowy pozwalający na pracę z sygnałami radiowymi, tj. filtrację, demodulację oraz wzmocnienie. Zastosowanie komputera znacząco rozszerzyło możliwości systemu oraz zapewniło możliwość podglądu widma sygnału radiowego, na podstawie którego możliwa jest ocena rodzaju i jakości odbieranego sygnału. System posiada również możliwość sterowania zewnętrznym urządzeniem odbiorczym z wykorzystaniem transmisji szeregowej.

SŁOWA KLUCZOWE: programowanie obiektowe, cyfrowe przetwarzanie sygnałów, radio definiowane programowo

1. WPROWADZENIE

Rozwój techniki cyfrowej znacząco poszerzył możliwości dotychczasowych rozwiązań w dziedzinie telekomunikacji i dał podstawy do stworzenia standardów radiowych konkurujących z ich przewodowymi odpowiednikami. Początkowo sygnał analogowy uzupełniany był jedynie o pewne cyfrowe elementy, związane głównie ze sterowaniem urządzeniami. Obecnie dąży się do przesyłania wszystkich danych w postaci zmodulowanych sygnałów cyfrowych. Świadczy o tym m. in. wprowadzenie naziemnej telewizji cyfrowej DVB-T na terenie całego kraju oraz nadawanie audycji w standardzie DAB+ uruchomione 1 października 2013 r. Również w paśmie fal długich, średnich i krótkich, gdzie stosowane są głównie analogowe modulacje amplitudy, wprowadza się międzynarodowy standard DRM oraz inne cyfrowe technologie transmisyjne. Zakres tych częstotliwości jest szczególnie interesujący ze względu na możliwości pokrycia znacznych obszarów sygnałem pochodzącym od jednego nadajnika. Zastosowanie metod cyfrowej transmisji daje w tym przypadku ciekawe rezultaty jakościowe oraz ekonomiczne. Jednak, aby istniał sens wprowadzania nowych systemów konieczne jest istnienie urządzeń odbiorczych umożliwiających dekodowanie sygnałów cyfrowych. Rozwiązania w postaci dedykowanych odbiorników są kosztowne i posiadają wiele ograniczeń technologicznych. Zastosowanie aplikacji komputerowej, elastycznej ze

* Politechnika Poznańska.

względu na możliwość rozwoju, znacząco obniża nakłady finansowe użytkownika oraz pozwala w łatwiejszy sposób rozpropagować nowoczesne standardy. Na podstawie tego założenia w pracy zrealizowano projekt oprogramowania komputerowego przetwarzającego sygnały radiowe zarówno analogowe jak i cyfrowe. Warunkiem koniecznym do działania aplikacji jest dostarczenie sygnału radiowego o odpowiedniej częstotliwości do przetwornika analogowo-cyfrowego, który jest elementem składowym karty dźwiękowej komputera.

2. PROJEKT APLIKACJI RADIA PROGRAMOWALNEGO

2.1. Założenia projektowe i wybór środowiska programistycznego

Podstawową cechą radia programowalnego jest szeroki zakres konfiguracji i pracy z różnymi typami sygnałów. Zastosowanie aplikacji komputerowej rozszerza możliwości standardowych urządzeń radiowych o funkcje dostępne dotychczas jedynie w drogich i trudno dostępnych konstrukcjach. Sprzyja to popularyzacji radia oraz rozszerza możliwości komunikacji.

Koncepcja projektu aplikacji systemu komputerowego przetwarzania sygnałów radiowych powstała na bazie następujących założeń:

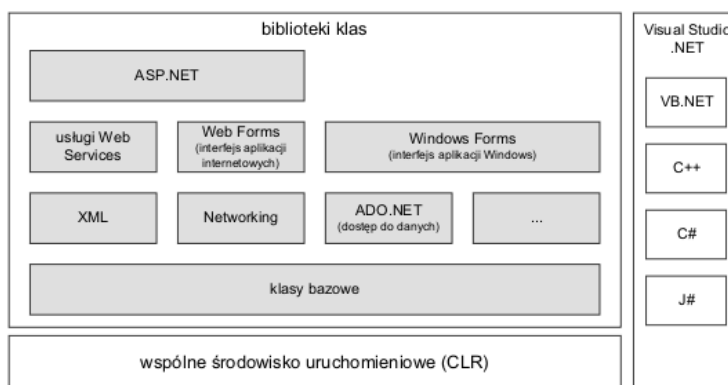
- działanie aplikacji w popularnym systemie operacyjnym,
- współpraca z urządzeniami dźwiękowymi komputera (odczyt sygnałów z wejść oraz generacja sygnałów wyjściowych),
- przetwarzanie sygnału z jak najmniejszym opóźnieniem,
- wizualizacja odbieranych sygnałów,
- przejrzysty i intuicyjny interfejs użytkownika,
- możliwość współpracy z zaprojektowanym układem fizycznym radia programowalnego.

Realizację założeń rozpoczęto od wyboru systemu operacyjnego, w którym będzie pracować aplikacja, platformy programistycznej, środowiska programistycznego oraz modelu aplikacji. Ogromna różnorodność dostępnych wariantów daje szeroki zakres wyboru. Jednak aby osiągnąć wysoką kompatybilność aplikacji wybrano następującą konfigurację:

- platforma programistyczna .NET Framework 4.0 firmy Microsoft,
- system operacyjny umożliwiający działanie platformy .NET 4.0,
- środowisko programistyczne Visual Studio 2012 Express for Windows Desktop,
- model obiektowy utworzony na podstawie języka wysokiego poziomu C#,
- interfejs graficzny programu stworzony w technologii WPF (ang. Windows Presentation Foundation).

Platforma programistyczna .NET Framework jest rozbudowanym narzędziem wykorzystywanym do tworzenia i obsługi aplikacji utworzonych na podstawie języków wysokiego poziomu. Jej podstawową cechą jest zastosowanie środowiska

uruchomieniowego CLR (ang. *Common Language Runtime*). Narzędzie to jest pewnego rodzaju wirtualną maszyną pozwalającą na uruchomienie aplikacji w każdym systemie z zainstalowaną platformą .NET. Kod napisany w jednym z obsługiwanych przez platformę języków wysokiego poziomu podczas kompilacji tłumaczony jest do postaci standaryzowanego kodu pośredniego CIL (ang. *Common Intermediate Language*). Podczas uruchomienia aplikacji na docelowej maszynie CLR, kompiluje kod CIL do postaci maszynowej w sposób optymalny dla procesora zastosowanego w urządzeniu. Dodatkowo środowisko uruchomieniowe czuwa nad prawidłową pracą takiej aplikacji, zajmowanymi zasobami pamięci oraz obsługą błędów. Platforma .NET zawiera wiele gotowych bibliotek pozwalających korzystać z możliwości sprzętowych komputera. Strukturę platformy .NET przedstawiono na rys. 1.

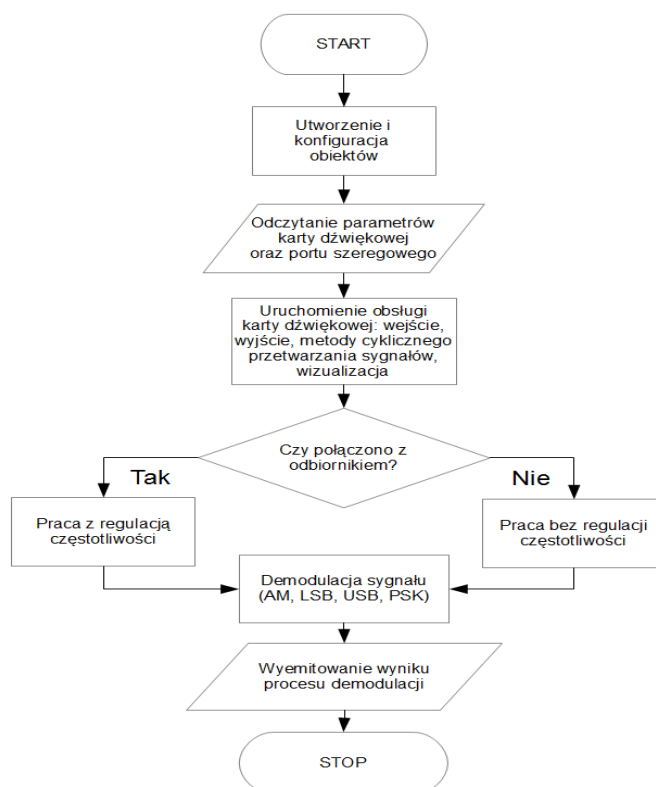


Rys. 1. Struktura platformy .NET [4]

W przedstawionym projekcie ze względu na wykorzystanie platformy .NET zastosowano środowisko programistyczne Visual Studio 2012 Express for Windows Desktop firmy Microsoft. Jest to w pełni funkcjonalny pakiet narzędzi umożliwiający tworzenie aplikacji okienkowych, konsolowych a także w formie dołączanych bibliotek z wykorzystaniem jednego z trzech dostępnych języków programowania. Do stworzenia programu wybrano język C#, który jest nowoczesnym, w pełni obiektywnym, stworzonym specjalnie dla platformy .NET językiem programowania wysokiego poziomu. System obiektowy polega na zastosowaniu klas hierarchicznych powiązanych wzajemnie między sobą, dzięki temu struktura aplikacji napisana w C# jest bardziej przejrzysta i intuicyjna. Nadrzędną klasą scalającą wszystkie pozostałe jest klasa typu *Object*. Ogromną zaletą jest również to, że pewne procesy zostały zautomatyzowane, np. zwalnianie pamięci po nieużywanych obiektach, co ułatwia pracę programisty [2].

Środowisko Visual Studio pozwala tworzyć aplikacje okienkowe w dwóch różnych systemach, Windows Forms (WF) oraz Windows Presentation Foundation (WPF). Między systemami występują różnice. W projekcie

zastosowano system WPF, który do tworzenia struktury graficznej interfejsu użytkownika wykorzystuje język XAML (ang. eXtensible Application Markup Language) oparty na standardzie XML. Dodatkowo w systemie WPF operacje obsługi interfejsu wspierane są sprzętowo poprzez wykorzystanie procesora karty graficznej [5]. Ogólna zasada działania projektowanej aplikacji radia programowalnego przedstawiona została na rys. 2.



Rys. 2. Algorytm działania aplikacji radia programowalnego

2.2. Komunikacja z wykorzystaniem portu szeregowego

Jednym z założeń projektu aplikacji jest możliwość komunikacji z układem odbiornika. W środowisku .NET biblioteką odpowiedzialną m. in. za komunikację poprzez port szeregowy jest *System.dll*. Zawiera ona wbudowaną klasę *SerialPort* zapewniającą kompletną współpracę z portem szeregowym. W projekcie zastosowano stałą szybkość transmisji o wartości 9600 baud. Możliwy jest wybór odpowiedniego portu oraz wielokrotne nawiązywanie i zamykanie połączenia podczas używania programu. Zastosowano również zabezpieczenia

uniemożliwiające połączenie w przypadku braku odpowiedniego urządzenia w komputerze, na którym uruchomiona jest aplikacja. Rysunek 3 przedstawia tworzenie obiektu *portSzeregowy* inicjującego pracę urządzenia.

```
portSzeregowy = new SerialPort(listaPortowCbx.SelectedItem.ToString(),  
9600, Parity.None, 8, StopBits.One);
```

Rys. 3. Inicjacja obiektu *portSzeregowy*

Ramka transmisyjna zastosowana w komunikacji składa się zawsze z 8 bajtów. Dane przesyłane są w formie pojedynczych znaków przypadających w ilości jednego znaku na bajt. Ponieważ wartość przesyłanej częstotliwości nie zawsze składa się z tylu cyfr, dlatego zastosowano automatyczne uzupełnianie pustych bajtów wartością zero wykorzystując metodę *PadLeft()*. Rysunek 4 przedstawia polecenie zapisujące dane rejestru do portu szeregowego.

```
portSzeregowy.Write(czestotliwosc.ToString().PadLeft(8, '0'));
```

Rys. 4. Polecenie wysyłające dane do urządzenia odbiorczego

Wartość wysyłanej częstotliwości zmieniana jest w momencie wystąpienia zdarzenia *MouseWheelEventsArgs*, które inicjowane jest ruchem obrotowym koła myszki jeżeli kursor znajduje się w obrębie kontrolki *czestotliwoscLbl* klasy *Label*. Wysyłane dane pobierane są z właściwości *Content* tej kontrolki. Jeden obrót zmienia tą wartość odpowiednio w górę lub dół o zadany krok, który reprezentowany jest przez właściwość *SelectedIndex* kontrolki *krokCbx* klasy *ComboBox*.

2.3. Komputerowy proces detekcji i demodulacji sygnałów

Podstawowym zadaniem projektowanej aplikacji jest możliwość demodulacji sygnałów radiowych odbieranych przy pomocy zewnętrznego układu. Wymaga to kontrolowania urządzenia dźwiękowego w komputerze, które wyposażone jest w odpowiednie przetworniki analogowo-cyfrowe. Standardowe składniki platformy .NET nie wspierają w pełni obsługi karty dźwiękowej, dlatego zastosowano zewnętrzną bibliotekę *Bass.NET* znacznie rozszerzającą możliwości współpracy oprogramowania w tej kwestii. Zapewnia ona kompleksową obsługę strumieni dźwiękowych, a więc ich tworzenie, przechwytywanie, przetwarzanie danych zawartych wewnątrz oraz wiele innych. Specjalny zbiór delegatów daje pełną kontrolę nad przetwarzanymi sygnałami. Cechy biblioteki umożliwiły opracowanie metod dokonujących przetwarzania sygnału w taki sposób, aby możliwe było otrzymanie pierwotnego sygnału pochodzącego od nadawcy. W procesie takim wykonywane są następujące kroki:

- przechwyt strumienia dźwięku z wejścia przetwornika analogowo-cyfrowego,
- wczytanie danych z bufora dźwięku,
- przetwarzanie danych (filtracja, przemiana częstotliwości, demodulacja),
- zapis przetworzonych danych do bufora,
- wyemitowanie otrzymanego dźwięku przez przetwornik cyfrowo-analogowy.

Zaprojektowane w aplikacji algorytmy umożliwiają demodulację sygnałów analogowych AM, SSB oraz cyfrowych BPSK. W pierwszym przypadku dane skopiowane z bufora przekształcane są na formę zespoloną, a następnie wyliczany jest moduł z jej wartości. Algorytm ten działa w taki sam sposób jak klasyczny detektor obwiedni [1, 3]. Fragment kodu realizującego tę operację przedstawia rys. 5.

```

if (demodulator == TypDemodulatora.AM)
{
    for (int i = 0; i < length / 2; i++)
    {
        sygnalZespolony = Poziom*strumienDzwieku16bit[i];
        strumienDzwieku16bit[i] = (short)(sygnalZespolony.Magnitude);
    }
}

```

Rys. 5. Kod przedstawiający proces demodulacji sygnału AM

Otrzymanie sygnału pierwotnego z modulacji jednowstęgowej jest zadaniem trudniejszym, ponieważ sygnał ten pozbawiony jest fali nośnej, którą należy wygenerować w odbiorniku. Do tego celu zaimplementowano specjalną klasę *Oscylator*, która umożliwia generowanie przebiegów sinusoidalnych o częstotliwościach akustycznych. Rysunek 6 przedstawia implementację metody demodulacji sygnału LSB [1, 3].

```

if (demodulator == TypDemodulatora.LSB)
{
    for (int i = 0; i < length / 2; i++)
    {
        vfo.GenerujSygnal();
        bfo.GenerujSygnal();
        sygnalZespolony= new Complex(strumienDzwieku16bit[i]
        *(Poziom*vfo.WyjścieCosinus), strumienDzwieku16bit[i]
        *(Poziom*vfo.WyjścieSinus));
        strumienDzwieku16bit[i]=(short)(sygnalZespolony.Real*
        bfo.WyjścieCosinus -
        sygnalZespolony.Imaginary*bfo.WyjścieSinus);
    }
}

```

Rys. 6. Kod przedstawiający proces demodulacji sygnału SSB

W pierwszym kroku generowany jest sygnał *vfo*, przy pomocy którego następuje sprowadzenie częstotliwości w dół oraz wytworzenie składowych zespolonych, które w następnym etapie mnożone są przez zespolony sygnał generatora nośnej *bfo* o częstotliwości 1,5 kHz. W zależności od typu odbieranej wstęgi bocznej urojony sygnał zespolony jest odpowiednio odejmowany (dla LSB) lub dodawany (USB) do wartości rzeczywistych. W obydwu metodach

demodulacji odbierany z wejścia liniowego sygnał jest w pierwszej kolejności filtrowany przy pomocy metod dostępnych w bibliotece *Bass.NET*.

W aplikacji zastosowano również dekodery sygnałów cyfrowych modulacji PSK31. Jego działanie polega na pomiarze fazy sygnału i określenie na tej podstawie wartości logicznej przesyłanego symbolu. Ponieważ sygnał PSK31 nadawany jest w standardzie BPSK, dlatego istnieje potrzeba określenia jedynie dwóch stanów fazy. Po przeprowadzeniu Szybkiej Transformaty Fouriera określany jest znak otrzymanej części rzeczywistej dla danego prążka, który odpowiada częstotliwości sygnału i na tej podstawie podejmowana jest decyzja o przypisaniu do zmiennej odpowiedniej wartości logicznej. Kod realizujący tę funkcję przedstawiono na rys. 7.

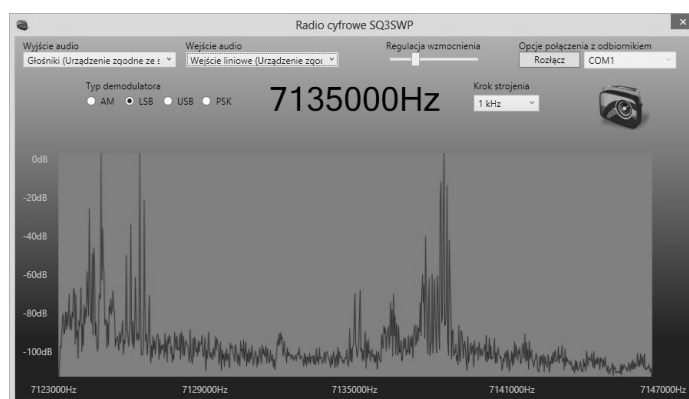
```
if (licznik < 10)
{
    if (KartaDzwiekowa.daneFFT[1228] > 0)
    {
        znak=(znak<< 1);
        znak += 1;
    }
    else
    {
        LicznikOdstepu++;
        if (LicznikOdstepu < 2)
        {
            znak = (znak << 1);
        }
        else
        {
            LicznikOdstepu = 0;
            licznik = 10;
        }
    }
    licznik++;
}
```

Rys. 7. Wykrywanie symboli PSK

2.4. Interfejs programu i wizualizacja odbieranych danych

Projekt graficzny aplikacji przewidziany został do współpracy z urządzeniem o minimalnej rozdzielczości ekranu 900 na 510 pikseli. Wszystkie opcje programu dostępne są z poziomu okna głównego, które podzielone jest na trzy części. W pierwszej (górna część formularza) znajdują się opcje dotyczące konfiguracji aplikacji, tj. wybór urządzeń dźwiękowych, regulacja wzmocnienia, wybór portu COM, przy pomocy którego aplikacja ma komunikować się z odbiornikiem oraz przycisk ustanawiający połączenie. W części środkowej formularza znajdują się ustawienia ściśle związane z funkcją radia, czyli wybór demodulatora, kroku strojenia oraz główny panel umożliwiający strojenie podłączonego

urządzenia. W sekcji trzeciej okna znajduje się wykres widma częstotliwościowego sygnału pochodzącego z wybranego wejścia audio. Funkcję tą zaimplementowano z wykorzystaniem klasy *Visuals* wbudowanej w strukturę biblioteki *Bass.NET*. Spektrum częstotliwości generowane jest cyklicznie w postaci bitmapy i wyświetlane w zarezerwowanym miejscu. Zastosowanie do tworzenia interfejsu technologii WPF umożliwia również tworzenie różnych schematów i skórek dotyczących wyglądu programu. Rysunek 8 przedstawia ekran główny opracowanej aplikacji.



Rys. 8. Główne okno aplikacji radia programowalnego

3. UWAGI KOŃCOWE

Rozwój radiokomunikacji dąży do ciągłej poprawy jakości i pewności transmitowanych sygnałów, a także ze względów ekologicznych osiągnięcia jak najwyższej sprawności transmisji. Układy cyfrowego radia programowalnego znacznie ułatwiają te działania dlatego ich tworzenie i użytkowanie jest wysoce pożądane. Możliwe, że w przyszłości każdy komputer klasy PC będzie standardowo wyposażony w moduł umożliwiający pracę z sygnałami radiowymi w bardzo szerokim zakresie częstotliwości.

Przedstawiona praca proponuje zastosowanie programu komputerowego posiadającego następujące właściwości:

- sterowanie elektronicznym urządzeniem odbiorczym,
- demodulacja sygnałów analogowych modulacji amplitudy AM, SSB,
- detekcja sygnałów cyfrowych PSK31,
- współpraca z różnego rodzaju kartami dźwiękowymi,
- możliwość rozbudowania o nowe funkcje bez potrzeby ingerencji w elektronikę układu odbiorczego,
- demodulacja sygnałów pochodzących od różnych odbiorników (warunkiem jest odpowiednia częstotliwość sygnału),

- wyświetlanie widma częstotliwościowego sygnału wejściowego,
- regulacja wzmocnienia sygnału,
- obsługa portu szeregowego.

Wydaje się, że wykorzystanie w projekcie platformy .NET i języka C# pozwala tworzyć ciekawe konstrukcje programistyczne o zaawansowanych cechach użytkowych, a także estetycznych. Tak stworzona aplikacja może być używana z powodzeniem w wielu systemach operacyjnych. W przyszłości możliwe jest po zastosowaniu pewnych modyfikacji, np. umieszczenie jej poprzez stronę zarządzaną ASP.NET w sieci i zapewnienie dostępu globalnego do odbiornika znajdującego się w konkretnej lokalizacji.

Wykorzystanie cyfrowych standardów radiowych daje również szerokie możliwości sterowania w elektroenergetyce. Uwzględniając czynnik ekonomiczny, ten sposób komunikacji może okazać się wyjątkowo korzystny.

LITERATURA

- [1] Haykin S., Systemy Telekomunikacyjne Tom I i II, Wydawnictwa Komunikacji i Łączności, Warszawa 1998.
- [2] Sharp J., Microsoft Visual C# 2010 Step by Step, Microsoft Press, USA 2010.
- [3] Szabatin J., Podstawy teorii sygnałów, Wydawnictwa Komunikacji i Łączności, Warszawa 2003.
- [4] Wprowadzenie – Platforma.NET [w:] MSDN, [Dostęp: 10.06.2013], Dostępny w World Wide Web: <<http://msdn.microsoft.com/pl-pl/library/wprowadzenie--platforma--net-framework.aspx>>
- [5] Wprowadzenie do WPF [w:] MSDN, [Dostęp: 14.04.2013], Dostępny w World Wide Web: <<http://msdn.microsoft.com/pl-pl/library/ms754130.aspx>>

COMPUTER PROCESSING OF RADIO SIGNALS

This paper concerns digital signal processing for radio frequencies. Developed and implemented a computer program which allows to work with radio signals, such as filtration, demodulation and amplification. The use of the computer significantly expanded capabilities of the system and provided the opportunity to preview radio spectrum, on the basis of which it is possible to evaluate the nature and quality of the received signal. The system also has possible to control the external receiver device using serial transmission.