

Andrzej Marczak
Politechnika Gdańska

TECHNOLOGIA RADIA PROGRAMOWALNEGO W ZASTOSOWANIACH WOJSKOWYCH

STRESZCZENIE

Technologia radia programowalnego (*Software Defined Radio*) jest nowoczesnym rozwiązaniem umożliwiającym realizację urządzeń pracujących w różnego rodzaju systemach łączności radiowej, zarówno cywilnych, jak i wojskowych. W artykule zaprezentowano zagadnienia dotyczące koncepcji realizacji radia programowalnego. Opisano w sposób funkcjonalny platformę sprzętową i programową takiego rozwiązania. Zaprezentowano budowę przykładowej platformy sprzętowej do realizacji radia programowalnego. Przedstawiono również architekturę oprogramowania używaną w systemie dla zastosowań wojskowych o nazwie *Joint Tactical Radio System* (JTRS).

Słowa kluczowe:

radio programowalne, SDR, JTRS.

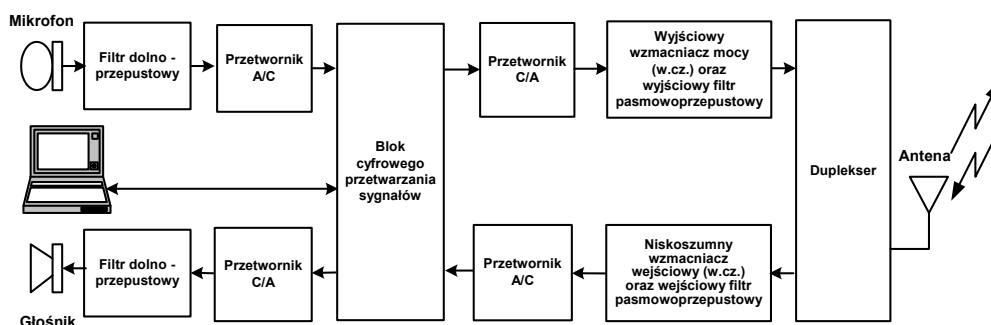
WSTĘP

Wraz z rozwojem systemów cyfrowej radiokomunikacji ruchomej istnieje potrzeba nieustannego opracowywania nowych rozwiązań terminali ruchomych, które mogłyby sprostać zapotrzebowaniu użytkowników na nowe usługi transmisji danych o dużych przepływnościach. Ponadto różnorodność standardów systemów cywilnych i wojskowych przy dużej mobilności ich użytkowników powoduje, 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 realizacji radia programowalnego (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, szeroko-pasmowe przetworniki C/A i A/C i procesor sygnałowy oraz inne układy programowalne [9, 10]. Wówczas funkcje nadawczo-odbiorcze mogą być głównie realizowane programowo przez procesor sygnałowy [1, 8].

ARCHITEKTURA PROGRAMOWALNEGO TERMINAŁA RUCHOMEGO

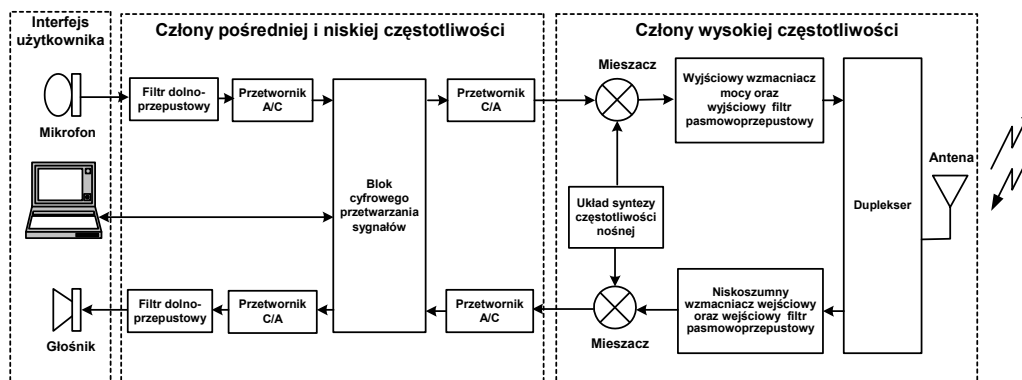
Architekturę programowalnego terminala ruchomego można ogólnie przedstawić jak na rysunku 1., przy czym symbol komputera reprezentuje źródło i/lub obiekt przeznaczenia dowolnych sygnałów cyfrowych z pominięciem sygnałów mowy [8].



Rys. 1. Ogólna architektura programowalnego terminala ruchomego

Źródło: J. Stefański, S. Gajewski, A. Marczak, *Radio rekonfigurowalne programowo w systemie UMCS*, „Elektronik”, 2001, nr 11.

Jak widać na rysunku, analogowy wzmacniacz mocy i filtr pasmowo-przepustowy w nadajniku poprzedza przetwornik C/A, do którego są dostarczane sygnały cyfrowe z bloku cyfrowego przetwarzania sygnałów, w którym są realizowane między innymi funkcje kodowania i modulacji, a niskoszumny wzmacniacz wejściowy i filtr pasmowo-przepustowy w odbiorniku przekazują analogowe sygnały odebrane poprzez przetwornik A/C do tego samego bloku cyfrowego przetwarzania sygnałów, między innymi w celu detekcji i dekodowania. Taka realizacja terminala programowalnego przy współczesnym poziomie rozwoju technologicznego jest na razie niewykonalna [9, 10]. 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 zostało przedstawione na rysunku 2. [2].



Rys. 2. Architektura terminala programowalnego z przetwarzaniem A/C i C/A w członie pośredniej częstotliwości

Źródło: A. Marczak, R. J. Katulski, J. Stefański, *Technika radia programowalnego*, „Przegląd Telekomunikacyjny”, 2004, nr 10.

Blok cyfrowego przetwarzania sygnałów w terminalu programowalnym powinien realizować następujące funkcje toru nadawczo-odbiorczego [9]:

- obsługi 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 bezpośredniego rozpraszania widma blok cyfrowego przetwarzania sygnałów powinien dodatkowo realizować następujące funkcje:

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

ARCHITEKTURA OPROGRAMOWANIA

Na tle ogólnej architektury programowalnego terminala ruchomego wydaje się celowe wydzielenie bardziej szczegółowej architektury oprogramowania takiego terminala. Niestety do tej pory nie jest znana żadna architektura tego rodzaju oprogramowania dla zastosowań cywilnych. Istnieje natomiast architektura oprogramowania terminala ruchomego dla zastosowań wojskowych. Nazywa się ona programową architekturą komunikacyjną SCA (*Software Communication Architecture*) [6] i została przygotowana oraz opublikowana przez biuro JPO (*Joint Program Office*) armii Stanów Zjednoczonych w ramach prac nad wspólnym taktycznym systemem radiowym JTRS (*Joint Tactical Radio System*). Biuro JPO zostało bowiem powołane w celu koordynowania prac nad rozwojem przyszłych wojskowych systemów telekomunikacyjnych, z uwagi na postęp technologiczny, który miał miejsce w ostatnich latach. Rozwój tych systemów ma na celu poprawę współpracy różnych nowoczesnych systemów łączności oraz redukcję kosztów ich modernizacji i rozwoju. Do podstawowych celów programu JTRS należy zwiększenie elastyczności i poprawy współdziałania systemów projektowanych przez różnych producentów oraz redukcja późniejszych kosztów utrzymania posiadanych rozwiązań.

Architektura SCA ma zapewniać przenośność aplikacji pomiędzy implementacjami SCA różnych producentów oraz umożliwiać redukcję kosztów i czasu projektowania systemów poprzez możliwości wielokrotnego wykorzystania zaprojektowanych wcześniej modułów oprogramowania, a także ułatwiać późniejsze, ewolucyjne zmiany struktury oprogramowania [4, 6]. Architektura SCA jest z założenia opracowana w celu zaspokojenia wymagań spodziewanych w odniesieniu do aplikacji wojskowych, jednak oczekuje się, że zostanie ona także uznana za standard komercyjny i będzie również wykorzystywana w cywilnych systemach radia programowalnego. Powodem tego jest fakt, że liczne, wiodące w świecie firmy zostały zaproszone do wspólnego opracowania standardu architektury SCA, który nie jest specyfikacją systemu, ale zbiorem zasad i reguł wytyczających projektowanie systemu w celu osiągnięcia podanych wyżej celów. Dokumentacja SCA zawiera więc podstawową specyfikację architektury oprogramowania, suplement dotyczący bezpieczeństwa, zasady tworzenia interfejsów aplikacji API (*Application Program Interface*) oraz dokumenty uzasadniające [6].

Struktura oprogramowania SCA definiuje środowisko programowe i specyfikuje usługi oraz interfejsy, których używają aplikacje. Środowisko programowe składa się przy tym z systemu operacyjnego czasu rzeczywistego, struktury rdzeniowej

(*Core Framework*) i oprogramowania pośredniczącego CORBA (*Common Object Request Broker Architecture*) [3, 6], służąc do komunikacji obiektów rozproszonych. Podstawowym celem oprogramowania CORBA jest umożliwienie komunikacji między odległymi i niekompatybilnymi systemami pracującymi na różnych platformach sprzętowych i programowych. Architektura oprogramowania CORBA pozwala uprościć proces tworzenia aplikacji rozproszonych w Internecie oraz w sieciach korzystających z wielu różnych protokołów [5]. Oprogramowanie CORBA wykonuje funkcje realizujące połączenia między obiektami dostarczającymi usługi a obiektami korzystającymi z nich. Elastyczność tej technologii umożliwia stosowanie dowolnych protokołów komunikacyjnych, korzystanie z dowolnej platformy systemowej oraz posługiwanie się praktycznie każdym językiem programowania [6].

Środowisko programowe narzuca ograniczenia projektowe na aplikacje dla zapewnienia większej przenośności z platform programowych zgodnych z architekturą SCA do innych platform. Polegają one na wykorzystaniu specyficznych interfejsów pomiędzy strukturą szkieletową i aplikacjami oraz ograniczeniu wykorzystania systemu operacyjnego. Architektura SCA określa ponadto moduły funkcjonalne zdefiniowane w suplemencie API. Definiują one interfejsy programowe pomiędzy różnymi zbiorami funkcji określającymi aplikacje. Takie moduły ułatwiają wielokrotne wykorzystywanie tych zbiorów funkcji i sprzyjają elastyczności projektowania [6].

Struktura szkieletowa architektury jest koncepcją wyznaczającą rdzeń złożony z otwartych programowych interfejsów i profili, które realizują operacje rozmieszczenia, zarządzania i komunikacji pomiędzy zbiorami funkcji wyznaczających aplikacje w systemie łączności opartym na przetwarzaniu rozproszonym. Ponadto część interfejsów może być wykorzystana przez aplikacje nienależące do struktury szkieletowej oraz przez producentów sprzętu. Struktura szkieletowa tworzy więc bazę danych na podstawie zbioru profili znanych jako domena profili (*Domain Profile*) i dostarcza ją do użytkownika w systemie [6].

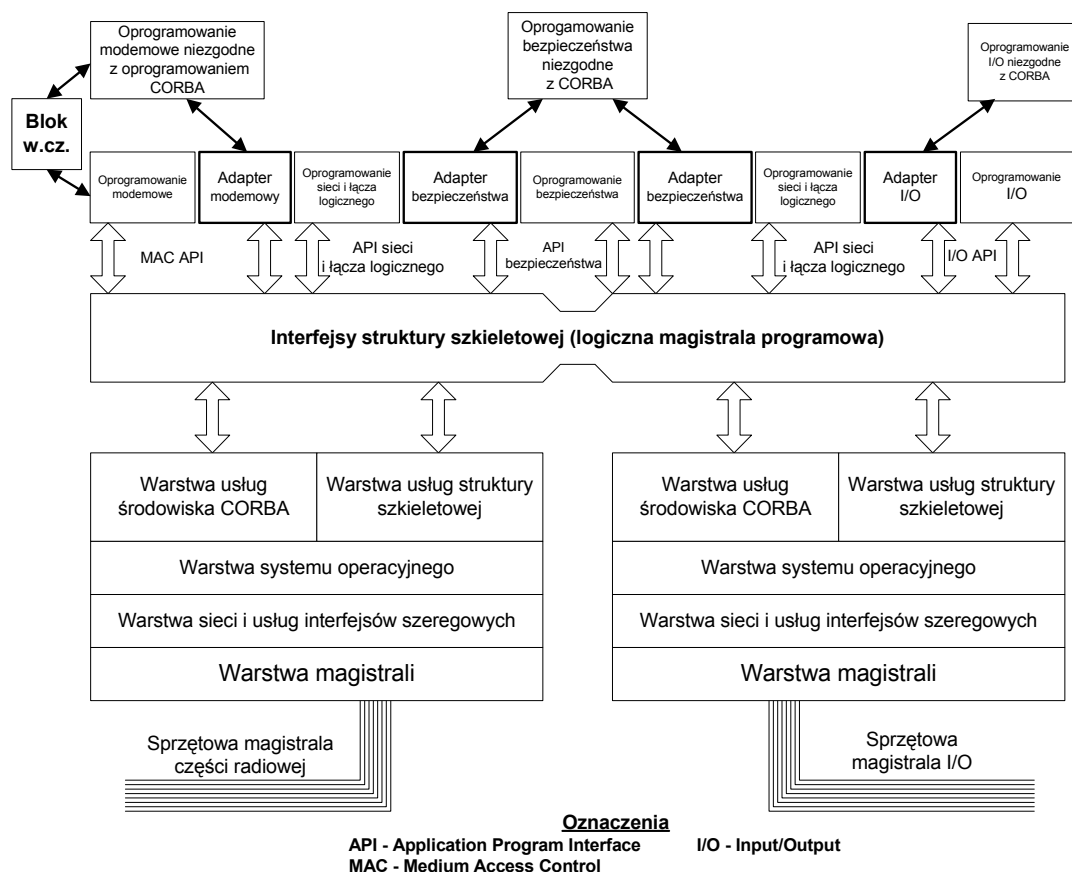
Nowością w tym rozwiązaniu jest użycie koncepcji zorientowanej obiektowo, również w opisie struktury sprzętowej. Koncepcja ta, wykorzystywana dotychczas w projektowaniu oprogramowania, została zastosowana do zdefiniowania bloków sprzętowych realizowanego systemu. Pierwotnym celem takiego podejścia do struktury sprzętowej była potrzeba wszechstronnego określenia i opisanie interfejsów oraz atrybutów poszczególnych sprzętowych elementów systemu. Zgodnie z tymi opisami producenci sprzętu mogą dostarczać dodatkowe moduły, a projektanci oprogramowania identyfikować moduły sprzętowe o konkretnych właściwościach dla określonych aplikacji [6].

Architektura SCA definiuje część programową i sprzętową na różnych poziomach hierarchii i precyzuje szerokie możliwości wielokrotnego wykorzystania i przenośności oprogramowania. Część programowa opiera się na modelowaniu obiektowym głównie w strukturze szkieletowej jako integralnej części środowiska operacyjnego. Ograniczenia projektanta oprogramowania nakładane przez architekturę wynikają z użycia interfejsów i struktury oprogramowania, a nie ze sposobu implementacji realizowanych funkcji. Dzięki temu innowacyjny projekt, lub jego część, może być wielokrotnie wykorzystany w różnych implementacjach. Taka architektura wyznacza zasady funkcjonowania systemu otwartego. Specyficzne wymagania implementacyjne mogą rozszerzać ten zbiór zasad, zwiększając możliwości wielokrotnego wykorzystania pewnych części oprogramowania wewnątrz i pomiędzy domenami. Interfejsy i zasady, które definiują zgodność z architekturą SCA, są integralną częścią specyfikacji. Wybrano je w celu zwiększenia możliwości przenoszenia, współpracy i konfiguracji oprogramowania oraz sprzętu, pozwalając nabywcy na elastyczne adresowanie wymagań i ograniczeń domeny [6].

Do graficznej reprezentacji interfejsów, układów, użytych przypadków i diagramów współpracy architektury SCA jest wykorzystywany zunifikowany język modelowania UML (*Unified Modelling Language*), określony przez zespół OMG (*Object Management Group*). Do definiowania interfejsów SCA jest używany język definicji interfejsu IDL (*Interface Definition Language*), również określony przez OMG. Jest to niezależny język programowania i może być kompilowany na przykład w językach C++ i Java. Oprócz tego wykorzystuje się język XML (*Extensible Markup Language*). Zastosowano go w profilu domen do identyfikacji właściwości oraz lokalizacji urządzeń i komponentów oprogramowania.

Architekturę oprogramowania terminala ruchomego przedstawia rysunek 3. [6]. Do głównych korzyści tej architektury należy wykorzystanie komercyjnych protokołów, oddzielenie aplikacji szkieletowych od innych aplikacji poprzez wiele warstw otwartej, komercyjnej infrastruktury programowej oraz wykorzystanie architektury CORBA w celu zapewnienia możliwości wielokrotnego wykorzystania, skalowalności i przenoszenia aplikacji.

Jak widać na rysunku 3., architektura oprogramowania ma strukturę warstwową [2, 6], najniższą jest warstwa magistrali. Architektura programowa jest zdolna funkcjonować w oparciu o różne komercyjne architektury magistrali, środowisko operacyjne obsługuje bowiem mechanizm transportowy, który może zawierać mechanizmy detekcji i korekcji błędów na poziomie obsługi magistrali. Przykładowymi magistralami możliwymi do zastosowania są magistrale: PCI, CompactPCI, Firewire i Ethernet. Środowisko operacyjne nie wyklucza wykorzystania innych magistrali [6].



Rys. 3. Architektura oprogramowania terminala ruchomego w technologii SDR

Źródło: SCA V3.0, Software Communications Architecture Specification, Joint Tactical Radio System (JTRS) Joint Program Office, August 2004.

Kolejną warstwą jest warstwa sieci i usług interfejsów szeregowych. Architektura programowa wykorzystuje również komercyjne programy do obsługi wielu interfejsów szeregowych i sieciowych. Możliwymi interfejsami sieciowymi i szeregowymi zastosowanymi w architekturze SCA mogą być: RS232, RS422, RS423, RS485, Ethernet i IEEE 802.x [6].

Dalej zostaną przedstawione pozostałe warstwy oprogramowania.

Warstwa systemu operacyjnego

Architektura programowa zawiera wbudowane funkcje systemu operacyjnego czasu rzeczywistego w celu zapewnienia wielowątkowej obsługi aplikacji. Architektura

ta wymaga standardowego interfejsu systemu operacyjnego dla usług systemowych w celu ułatwienia przenoszenia aplikacji. Przewiduje wykorzystanie systemu operacyjnego POSIX (*Portable Operating System Interface*) [2, 6], który jest akceptowanym standardem przemysłowym. System operacyjny POSIX i jego rozszerzenia czasu rzeczywistego są kompatybilne z wymaganiami obsługi architektury CORBA [2, 6].

Warstwa struktury szkieletowej

Warstwa struktury szkieletowej składa się z bazowych interfejsów aplikacji, które mogą być wykorzystane przez wszystkie aplikacje. Zawiera też szkieletowe interfejsy sterujące, które zapewniają sterowanie w systemie. Ponadto zawiera interfejsy usług szkieletowych, które obsługują zarówno aplikacje struktury szkieletowej, jak i pozostałe aplikacje. Składnikiem warstwy struktury szkieletowej jest również domena profili opisująca własności urządzeń i oprogramowania w systemie [6].

Oprogramowanie CORBA

Oprogramowanie CORBA jest strukturą wieloplatformową, która może być wykorzystana do standardowych operacji typu klient/serwer, gdy używamy przetwarzania rozproszonego [6].

Warstwa aplikacji

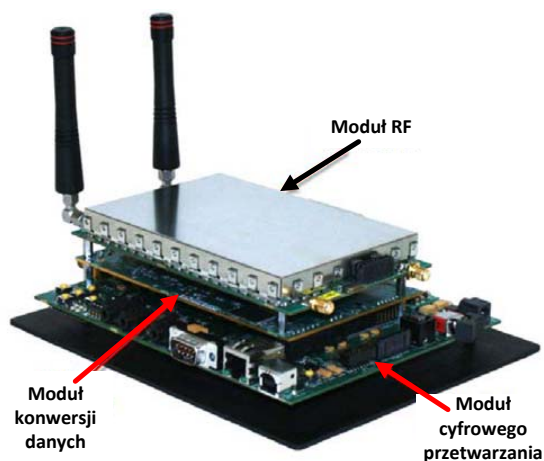
Aplikacje wykonują funkcje komunikacji z użytkownikiem i zawierają przetwarzanie sygnałów na poziomach warstw modemu, łącza oraz sieci. Realizują również międzysieciowy dobór drogi i zewnętrzny dostęp I/O. Aplikacje korzystają z interfejsów i usług struktury szkieletowej. Bezpośredni dostęp aplikacji do systemu operacyjnego jest ograniczony przez usługi opisane w specyfikacji profilu POSIX. Funkcje sieciowe, które mogą być implementowane poniżej warstwy aplikacji, takie jak komercyjna warstwa IP, nie są ograniczone profilem POSIX, jeśli są umieszczone w przestrzeni jądra systemu operacyjnego [6].

Adaptery

Adaptery są urządzeniami i zasobami programowymi wykorzystywanymi do obsługi składników struktury oprogramowania niezgodnych z oprogramowaniem CORBA. Adaptory są używane do realizacji translacji informacji pomiędzy zasobami programowymi lub sprzętowymi zgodnymi ze standardem CORBA i zasobami niepracującymi według tego standardu [6].

PLATFORMA SPRZĘTOWA DO REALIZACJI RADIA PROGRAMOWALNEGO

Przykładem platformy sprzętowej do realizacji radia programowalnego jest Small Form Factor SDR firmy Lyrtech RD [7]. Składa się ona z trzech głównych bloków zrealizowanych na trzech połączonych ze sobą płytkach. Blokami tymi są: moduł radiowy (RF), moduł konwersji danych oraz moduł cyfrowego przetwarzania. Moduł radiowy został wyposażony w dwie anteny, dzięki którym można nadawać i odbierać sygnał radiowy. Widok całego urządzenia przedstawia rysunek 4.

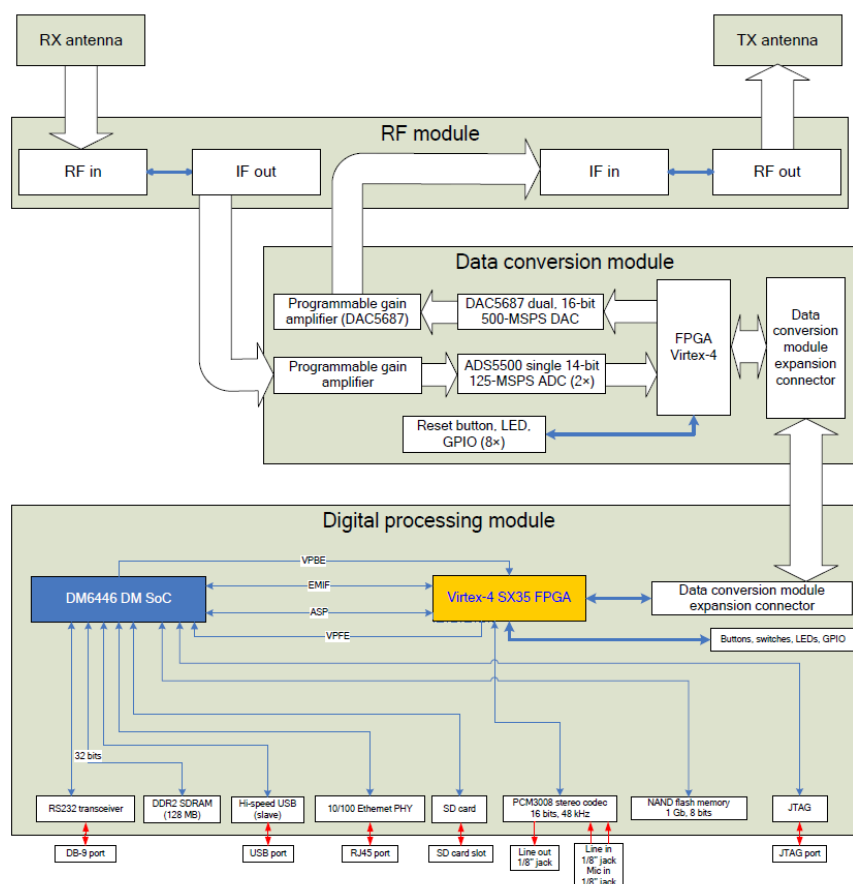


Rys. 4. Widok przykładowej platformy sprzętowej radia programowalnego [7]

Źródło: *Small Form Factor SDR Evaluation Module/Development Platform User's Guide*, Lyrtech 2010.

Platforma jest urządzeniem o niewielkich rozmiarach zawierającym wszystkie komponenty sprzętowe niezbędne od realizacji urządzenia nadawczo-odbiorczego w technologii radia programowalnego. Schemat blokowy platformy sprzętowej został przedstawiony na rysunku 5. (*Digital processing module*). Składa się on z dwóch głównych elementów: matrycy FPGA Virtex-4 firmy Xilinx i procesora TMS320DM6446 SoC firmy Texas Instruments. Procesor zawiera w jednej obudowie scalonego procesor sygnałowy DSP (*Digital Signal Processor*) i procesor ogólnego przeznaczenia GPP (*General Purpose Processor*). Procesor może korzystać z pamięci SDRAM DDR2 o pojemności 128 MB i pamięci flash o pojemności 1 GB. Moduł cyfrowego przetwarzania zawiera również interfejsy: RS232, USB i Ethernet. Zawiera też kodek stereo, dzięki któremu możliwe jest podłączenie słuchawek, mikrofonu i zewnętrznego źródła dźwięku. Poza tym zawiera przyciski, których funkcje można programować oraz diody LED informujące o stanie pracy elementów modułu. Moduł konwersji

danych będący drugim modulem platformy sprzętowej połączony jest z modulem cyfrowego przetwarzania specjalnym złączem (*Data conversion module expansion connector*). Zawiera matrycę FPGA Virtex-4, dwukanałowy 16-bitowy przetwornik cyfrowo-analogowy DAC5687, dwa 14-bitowe przetworniki analogowo-cyfrowe ADC5500 oraz wzmacniacze o programowalnej wartości wzmocnienia. Moduł radiowy w wersji low-band może pracować w paśmie 200 MHz-1 GHz z szerokością kanału 5 MHz lub 20 MHz. Wersja high-band modułu radiowego może pracować z takimi samymi szerokościami kanału w paśmie 1.6–2.2 GHz. Szerokość kanału i częstotliwości — zarówno dla kierunku nadawania, jak i odbioru — mogą być zmieniane w sposób programowy. Obie wersje modułów radiowych zawierają bloki up-konwerterów przenoszących sygnał z pośredniej częstotliwości (30 MHz) na właściwą częstotliwość radiową, bloki down-konwerterów realizujących operację odwrotną i odpowiednie filtry.



Rys. 5. Schemat blokowy przykładowej platformy sprzętowej radia programowalnego

Źródło: *Small Form Factor SDR Evaluation...*, wyd. cyt.

Przedstawiona platforma sprzętowa umożliwia uruchamianie i testowanie oprogramowania realizującego nadajnik i odbiornik w technologii radia programowalnego. Oprogramowanie napisane i skompilowane na komputerze klasy PC może być wprowadzone do procesora i matrycy FPGA poprzez interfejs sieci Ethernet i uruchomione na sprzęcie, realizując nadajnik i odbiornik dla danego systemu łączności radiowej. Możliwa jest programowa implementacja całego toru nadawczo-odbiorczego z operacjami modulacji/demodulacji, kodowania/dekodowania kanałowego i źródłowego, szyfracji/deszyfracji. Dzięki dużej szerokości kanału (5 MHz lub 20 MHz) oraz dużej mocy obliczeniowej możliwa jest realizacja interfejsów radiowych DS-CDMA, FH-CDMA i OFDM stosowanych w nowoczesnych systemach łączności wojskowych i cywilnych.

PODSUMOWANIE

Technologia radia programowalnego umożliwia szybką zmianę właściwości sprzętu i jego adaptację do aktualnych zastosowań. Szeroki zakres rozwiązań systemowych w zastosowaniach wojskowych i cywilnych powodował konieczność wykorzystania dużej liczby różnorodnego sprzętu umożliwiającego łączność radiową. Technologia SDR umożliwia użycie tego samego sprzętu, z odpowiednim oprogramowaniem, w różnych, często odmiennych zastosowaniach, zarówno wojskowych, jak i cywilnych. Dodatkowo możliwość łatwego, programowego upgrade'u właściwości sprzętu pozwala dłużej wykorzystywać urządzenia. Programowa architektura oprogramowania (SCA), opisana w artykule, wykorzystując sprawdzone i uniwersalne rozwiązania, umożliwia łatwą przenośność oprogramowania, które może być instalowane i uruchamiane na sprzęcie zgodnym z tą architekturą, wyprodukowanym przez dowolnego producenta.

Oprogramowanie testowane i uruchomione na platformie SFF SDR w zaprezentowanej wersji laboratoryjnej może być przeniesione i uruchomione na podobnym urządzeniu w wersji wojskowej w postaci radiostacji doreęcznej lub przewoźnej do zamontowania w pojeździe.

BIBLIOGRAFIA

- [1] Harada H., Prasad R., *Simulation nad Software Radio for Mobile Communication*, Artech House, London 2002.
- [2] Marczak A., Katulski R. J., Stefański J., *Technika radia programowalnego*, „Przegląd Telekomunikacyjny”, 2004, nr 10.

- [3] Mitola III J., *Software Radio Architecture Evolution: Foundations Technology Tradeoffs, and Architecture Implications*, IEICE Trans. Commun., June 2000.
- [4] Mitola III J., *Software Radio Architecture*, Wiley & Sons, 2000.
- [5] Sawerwain M., *CORBA. Programowanie w praktyce*, Wydawnictwo MIKOM, 2002.
- [6] SCA V3.0, *Software Communications Architecture Specification*, Joint Tactical Radio System (JTRS) Joint Program Office, August 2004.
- [7] Small Form Factor SDR Evaluation Module/Development Platform User's Guide, Lyrtech 2010.
- [8] Stefański J., Gajewski S., Marczak A., *Radio rekonfigurowalne programowo w systemie UMCS*, „Elektronik”, 2001, nr 11.
- [9] Wesołowski K., *Koncepcja Software Radio i jej znaczenie dla rozwoju radiokomunikacji ruchomej*, Krajowa Konferencja Radiodyfuzji i Radiokomunikacji KKRR '98, Poznań 1998.
- [10] Wesołowski K., Krenz R., *Software Radio — technologia przyszłych systemów radiokomunikacji ruchomej*, Krajowa Konferencja Radiokomunikacji, Radiofonii i Telewizji KKRRiT '00, Poznań 2000.

SOFTWARE DEFINED RADIO TECHNOLOGY IN MILITARY APPLICATIONS

ABSTRACT

Software Defined Radio (SDR) is a modern solution used to develop of devices implemented in various types of radio systems, both civilian and military. The paper presents some issues concerned with the implementation of the concept of the SDR. It describes the software and hardware platform for such a solution. The paper presents the structure of the hardware platform used to develop a programmable radio. It also includes software communication architecture for military applications called the Joint Tactical Radio System (JTRS).

Keywords:

software defined radio, SDR, JTRS.