



# Popularne protokoły komunikacyjne w sieciach radiowych WPAN

*Urządzenia pracujące w sieci bezprzewodowej przypominają ludzi, którzy zabrali się, aby coś przedyskutować. Pół biedy, jeśli nie ma ich zbyt wielu, a tematyka spotkania nie budzi specjalnych emocji. Wtedy każdy ma szansę wypowiedzieć się i być wysłuchanym. Jednak w miarę jak przybywa osób, a „temperatura” spotkania rośnie, krzykacze (niekoniecznie mający coś ważnego do powiedzenia) zagłuszają cichych, a niektórzy mówcy wchodzą sobie w słowo, przekrzykując się nawzajem. Do tego, niektórzy o czymś szepczą pomiędzy sobą, a ważne informacje giną w ogólnym „kotle”. Czy można jakoś uporządkować tę sytuację?*

Przed podobnym problemem stanęły już parlamenty w starożytnych, greckich państwach – miastach. Aby każdy zainteresowany mógł dojść do głosu, ustalono zbiór pewnych reguł, który dawał szansę wypowiedzenia się każdemu zainteresowanemu. Ten „zbiór reguł” to coś, co w sieciach nazywamy protokołem komunikacyjnym. Oczywiście, zaraz rozwinęły się szkoły mówców, które uczyły sposobów przekonywania słuchaczy do swoich racji (i tak do dziś, ale o tym szła!). Na szczęście sieci radiowe służące do przesyłania danych, są wolne od poglądów politycznych, bo nie wiadomo, czy jakkolwiek transmisja doszłaby do skutku.

Protokół sieci radiowej to taki zbiór reguł funkcjonowania jej komponentów, w których zawarte są naturalne zalecenia, by nie przerywać wypowiedzi innych i wypowiadać się tylko wtedy, gdy przydzielono nam na to czas lub gdy o to poproszą. Ale to tylko część zasad, ponieważ istotny jest również język wypowiedzi. I nie mam tu na myśli jego poprawnego stylu, ale raczej to, aby wszystkie urządzenia mówiły w sposób pozwalający im „rozumieć się” i rozpoznawać sytuację, gdy któreś z nich przeinacza fakty. „Język” powinien też pozwalać na skorygowanie błędu np. poprzez powtórzenie wypowiedzi. Na

szczęście urządzenia komunikujące się drogą radiową – inaczej niż ludzie – są wolne od emocji i przestrzegają ustalonych (zaprogramowanych) zasad, nie próbują w trakcie funkcjonowania sieci wprowadzać jakiś innowacji i własnych reguł postępowania.

Za każdym razem, gdy konstruktor systemu staje przed koniecznością zbudowania takiego „parlamentu” bezprzewodowego musi rozwiązać problem protokołu komunikacyjnego. Można to zrobić samodzielnie, ale można również skorzystać z gotowych rozwiązań. Tym bardziej, że wykorzystując pewne standardy komunikacyjne możemy komunikować się z urządzeniami innych producentów i wymieniać z nimi dane. Mamy też pewność, że nie naruszamy reguł współużytkowania pasm radiowych. Niemniej jednak, w praktyce można spotkać oba rozwiązania problemu.

Jak zawsze, użytkowanie standardu ma swoje zalety, między innymi te wymienione wyżej, ale ma też kilka wad. Do podstawowych należy zaliczyć nadmiarowość protokołu (nie zawsze uda nam się użyć wszystkich funkcji protokołu w aplikacji), a co za tym idzie – oprogramowania (obszerne biblioteki),

duży koszt licencji, na które stać tylko bogate firmy. Zaprezentujemy najbardziej popularne standardy lokalnych sieci radiowych małego zasięgu: ZigBee i Bluetooth. Standard WiFi będzie przedmiotem kolejnego, wrześniowego numeru Elektroniki Praktycznej.

## ZigBee

W opracowanie standardu ZigBee włożono wiele wysiłku i co ważne – jest on stale rozwijany. Od kilku lat ogłaszane są kolejne specyfikacje, usiłujące nadążać za potrzebami. Sieci ZigBee przeważnie buduje się z użyciem gotowych modułów.

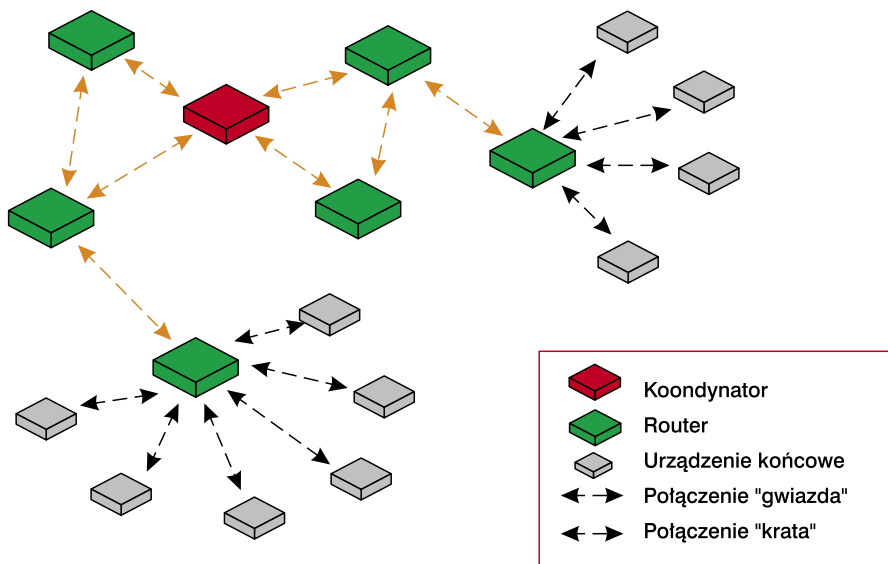
ZigBee jest opisem protokołu komunikacyjnego wysokiego poziomu, używającym modułów radiowych o niewielkiej mocy, bazujących na standardzie 802.15.4-2003 opisującym działanie lokalnych, bezprzewodowych sieci cyfrowych małego zasięgu (WPAN tj. *Wireless Personal Area Network*). Twórcy standardu użyli słowa „osobistych”, aby podkreślić przeznaczenie protokołu. W zamierzeniach moduły ZigBee miały być prostsze w budowie i tańsze, niż inne typy sieci WPAN, jak np. Bluetooth.

Standard ZigBee powstał głównie z myślą o zastosowaniach przy budowie lokalnych sieci czujników i urządzeń monitorujących, do aplikacji związanych z automatyką budynków, kontrolą urządzeń pracujących w obrębie pomieszczenia, systemami alarmowymi i logistyką. Główny nacisk położono nie tyle na zapewnienie jak największej przepływności, ile na niezawodność transmisji danych, łatwość budowy (i rozbudowy) sieci oraz elastyczność jej konfigurowania. Jako, że przy budowie sieci czujników tego typu zawsze istnieje problem dostępności zasilania, to takie urządzenia często są zasilane z baterii. Dlatego też twórcy protokołu położyli szczególny nacisk na minimalizację poboru energii zasilania. Maksymalnie uproszczono też stos komunikacyjny, co w założeniu miało zapewnić niską cenę modułów radiowych.

Na **rysunku 1** pokazano przykładową sieć ZigBee. Może w niej pracować praktycznie dowolnie duża liczba urządzeń, ponieważ ograniczeniem w pojedynczej sieci jest abstrakcyjna wręcz liczba  $2^{64}$  węzłów, a łącznie może ze sobą współpracować  $2^{16}$  sieci!

Protokół komunikacyjny umożliwia realizację sieci o praktycznie dowolnej strukturze: gwiazdzysta (*star*), równy z równym (*peer-to-peer*), kratowa (*mesh*). Dodatkowo, urządzeniom o wymaganym krótkim czasie reakcji można nadać odpowiednio wysoki priorytet.

Moduły radiowe ZigBee pracują z wykorzystaniem pasm ISM, zgodnie z regulacjami prawnymi obowiązującymi na danych terytoriach. Prędkość transmisji w paśmie 2,4 GHz (dostępnym praktycznie na całym świecie) jest równa 250 kb/s (16 kanałów, każdy o szerokości 5 MHz). Mimo popularności tego pa-



Rysunek 1. Struktura sieci ZigBee

sma, spotyka się również urządzenia ZigBee wykorzystujące pasma 915 MHz (USA i Australia) i mające prędkość transmisji 40 kb/s oraz 868 MHz (Europa) i prędkość 20 kb/s. Zasięg modułów waha się w granicach od 10 do 70 m, aczkolwiek dostępne są również specjalne moduły ZigBee Pro o zasięgu do 1500 m. Typowo, moc nadajnika modułu ZigBee jest równa 0 dBm (1 mW).

Ze względu na fakt, że implementacja pełnego stosu transmisji ZigBee nie jest taka tania, jak przewidywano (powodem jest wymagany, spory rozmiar pamięci programu oraz wymagania odnośnie do pamięci operacyjnej), to w opisie protokołu wyróżniono dwa rodzaje urządzeń:

- **FFD** (*Full Function Device*) tj. urządzenia mogące pracować w sieciach o dowolnej topologii, jako koordynator sieci lub po prostu komunikując się z każdym urządzeniem w sieci.
- **RFD** (*Reduced Function Device*) tj. urządzenie mogące pracować wyłącznie w sieciach o topologii gwiazdy, mogące komunikować się wyłącznie z koordynatorem sieci i niemogące spełniać funkcji koordynatora.

Oprócz opisanego wyżej podziału, wprowadzono również inny, uwzględniający ich funkcje, zgodnie z protokołem ZigBee:

- Koordynator oznaczany literami **ZC** (*ZigBee Coordinator*) lub **CO** (*Coordinator*). Jest to urządzenie o największych możliwościach. Koordynator formuje korzeń drzewa sieci i może łączyć się z inną siecią. W każdej sieci ZigBee wymagany jest jeden taki moduł, ponieważ bez niego sieć nie może pracować. Moduł ten inicjuje pracę sieci, zarządza tablicą jej połączeń, zapamiętuje informacje związane z siecią, jak również może pracować jako centrum uwierzytelniania urządzeń sieciowych i przechowywać repozytoria kluczy szyfrowania.

- Router oznaczany literami **ZR** (*ZigBee Router*) lub **RT** (*Router*). Jest to urządzenie w sieci ZigBee, które pośredniczy w przekazywaniu danych. Często router realizuje również zadania związane bezpośrednio z aplikacją użytkownika.
- Urządzenie końcowe oznaczane jako **ZED** (*ZigBee End Device*) lub **RFED** (*Reduced Function End Device*). Urządzenie to ma wystarczającą funkcjonalność, aby komunikować się z routerem lub koordynatorem. Nie może jednak porozumiewać się z innymi urządzeniami końcowymi. Taki sposób funkcjonowania umożliwi urządzeniu końcowemu na przebywanie przez większość czasu w stanie czuwania, w którym minimalizowany jest pobór energii.

W porównaniu z innymi rozwiązaniami sieci WPAN, ZigBee ma unikatowe cechy, niedostępne w innych rozwiązaniach sieci bezprzewodowych. Stosunkowo nieskomplikowany protokół ZigBee nie ma zbyt dużych wymagań co do zasobów sprzętowych i bez problemu daje się uruchomić na mikrokontrolerze o 8-bitowym rdzeniu. Dlatego też moduły mogą być tanie. Ponadto protokół komunikacyjny umożliwia omijanie zatorów komunikacyjnych, a dzięki kratowej strukturze sieci, łatwo jest zwiększać obszar pokrycia bez podnoszenia mocy nadajników, zmiany anten itp. Urządzenia ZigBee przez większość czasu przebywają w trybie obniżonego poboru mocy i wybudzają się z niego na czas liczony w milisekundach. Jak wspomniano, nadajnik ma moc zaledwie 1 mW (0 dBm), co prowadzi do radykalnych oszczędności baterii. W praktyce, okres wymiany baterii może sięgać nawet lat.

## Bluetooth

W 1994 r. rozpoczęto prace nad skonstruowaniem standardu, którego celem było zdefiniowanie nowego, bezprzewodowego

sposobu łączenia telefonów komórkowych z innymi urządzeniami, cechującego się niewielkim zasięgiem, małym poziomem mocy nadawczej, niewielkim poborem prądu oraz niską ceną. Wynikiem końcowym było uzyskanie w 1999 r. standardu transmisji pracującego w paśmie ISM 2,4 GHz. Opisano go w specyfikacji IEEE 802.15.1. Nowemu standardowi nadano nazwę Bluetooth. Jego liczne zalety w połączeniu z niską ceną spowodowały ogromną popularność tego rozwiązania. Aplikacje, w których jest on stosowany, dawno już wykroczyły poza pierwotne pomysły.

„Dziwna” nazwa interfejsu pochodzi od przydomka króla duńskiego, Haralda Sinozębnego (Blåtand), który około 970 roku zjednoczył rywalizujące plemiona duńskie i norweskie. Podobnie Bluetooth, został zaprojektowany, aby połączyć ze sobą odmienne technologie: telefony komórkowe, komputery, drukarki, kamery, aparaty cyfrowe, bezprzewodowe tory transmisji sygnału audio itp. Logo Bluetooth łączy w sobie znaki alfabetu runicznego będące odpowiednikami łacińskich liter H i B.

Podstawową jednostką organizacyjną w sieci urządzeń Bluetooth jest *piconet* (pikosieć). Zawiera ona węzeł typu *master* i co najwyżej 7 węzłów typu *slave*. W jednym obszarze może koegzystować wiele takich jednostek, a nawet mogą one być ze sobą połączone za pomocą węzła typu *bridge*. Piko-sieci połączone ze sobą w taki sposób określa się mianem *scatternet* (sieć rozproszona).

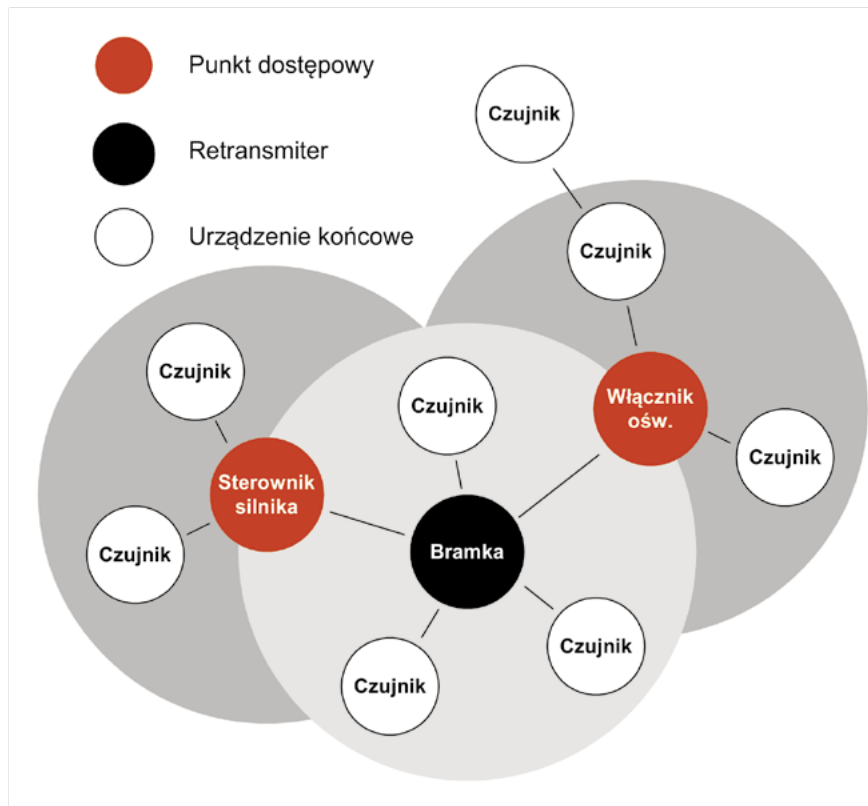
Oprócz 7 węzłów typu *slave*, w jednej pikosieci może pracować do 255 węzłów pozostających w stanie synchronizacji z urządzeniem *master*, będących w trybie oczekiwania i obniżonego poboru mocy. Urządzenia te nie uczestniczą w wymianie danych, a jedynie mogą otrzymać sygnał aktywacyjny lub nawigacyjny od węzła typu *master*. Istnieją dwa przejściowe stany: *hold* oraz *sniff*.

Konsekwencją wprowadzenia takiego ścisłego podziału na pikosieci oraz węzły *master* oraz *slave* jest całkowite podporządkowanie tych drugich węzłom *master*. To one w obrębie danej pikosieci kontrolują zegar i określają, które urządzenie i w którym slotcie czasowym może się z nim komunikować, a wymiana danych może nastąpić tylko pomiędzy węzłem *master* i *slave*.

Mimo pewnych niedogodności standard bardzo upowszechnił się i konstruktorzy coraz chętniej (albo z przymusu) sięgają po moduły Bluetooth w celu umożliwienia komunikowania się urządzeń z resztą „cyfrowego świata”. Na pierwszy rzut oka może to wydawać się dosyć trudne do wykonania, jednak współcześnie są dostępne gotowe rozwiązania w formie modułów Bluetooth, które wystarczy zamontować w urządzeniu. Rozwiązują one problem zgodności ze specyfikacją IEEE 802.15.1 oraz wykonania części

Profile Bluetooth (w kolejności alfabetycznej):

- Advanced Audio Distribution Profile (A2DP) – jednokierunkowa transmisja sygnału audio o wysokiej jakości.
- Attribute Profile (ATT) – dla potrzeb Bluetooth Low Energy.
- Audio/Video Remote Control Profile (AVRCP) – kontrola urządzeń RTV.
- Basic Imaging Profile (BIP) – przesyłanie obrazów.
- Basic Printing Profile (BPP) – przesyłanie tekstu dla drukarki.
- Common ISDN Access Profile (CIP) – obsługa usług ISDN.
- Cordless Telephony Profile (CTP) – obsługa bezprzewodowych słuchawek telefonicznych.
- Device ID Profile (DIP) – zaawansowaną identyfikacja urządzenia.
- Dial-up Networking Profile (DUN) – umożliwienie korzystania za pośrednictwem Bluetooth z połączeń wdzwanianych.
- Fax Profile (FAX) – umożliwia korzystania z faksu.
- File Transfer Profile (FTP) – wykonywanie operacji na plikach.
- Generic Audio/Video Distribution Profile (GAVDP) – baza dla A2DP i VDP.
- Generic Access Profile (GAP) – umożliwienie wykrywania się urządzeń Bluetooth i nawiązywania połączeń.
- Generic Attribute Profile (GATT) – przekazywanie informacji o urządzeniu.
- Generic Object Exchange Profile (GOEP) – określa sposób wymiany informacji pomiędzy urządzeniami.
- Hard Copy Cable Replacement Profile (HCRP) – wirtualny kabel pomiędzy urządzeniem a drukarką.
- Health Device Profile (HDP) – umożliwia wymianę danych z urządzeniami medycznymi.
- Hands-Free Profile (HFP) – dla samochodowych zestawów głośnomówiących; wzbogacone funkcje w porównaniu do Headset Profile.
- Human Interface Device Profile (HID) – do obsługi urządzeń interfejsu użytkownika, np.: klawiatury, myszki, kontrolerów gier, itp.
- Intercom Profile (ICP) – do bezpośredniej, dwukierunkowej komunikacji audio pomiędzy dwoma urządzeniami Bluetooth.
- LAN Access Profile (LAP) – mostek do sieci LAN.
- Message Access Profile (MAP) – wymiana krótkich wiadomości tekstowych.
- Object Push Profile (OPP) – przesyłanie obiektów, takich jak np. pliki graficzne, pomiędzy urządzeniami.
- Personal Area Networking Profile (PAN) – enkapsulacja Bluetooth w warstwie trzeciej (wg OSI).
- Phone Book Access Profile (PBAP, PBA) – dostęp do książki adresowej.
- Serial Port Profile (SPP) – emulacja portu szeregowego UART.
- Service Discovery Application Profile (SDAP) – sposób wykrywania usług dostępnych na zdalnym urządzeniu Bluetooth.
- SIM Access Profile (SAP, SIM, rSAP) – korzystanie z danych zgromadzonych na kartach SIM innego urządzenia Bluetooth.
- Synchronization Profile (SYNCH) – synchronizacja informacji.
- Video Distribution Profile (VDP) – przesyłanie strumienia wideo przez łącze Bluetooth.
- Wireless Application Protocol Bearer (WAPB) – przekazywanie pakietów WAP (Wireless Application Protocol) przez łącze Bluetooth.



Rysunek 2. Topologia sieci SimpliCI



radiowej. Dla naszego systemu nadrzędnego będzie to jeszcze jeden element, z którym komunikuje się on za pomocą interfejsu np. UART.

Różnice pomiędzy poszczególnymi modułami wynikają z ich aplikacji. Cechą wyróżniającą moduł Bluetooth są profile zaimplementowane w jego stosie komunikacyjnym. Determinują one sposób użycia modułu w aplikacji oraz rodzaje interfejsów łączących moduł z urządzeniami zewnętrznymi. Na przykład, jeśli moduł ma profil SPP (*Serial Port Profile*), to prawdopodobnie ma też interfejs UART. Nierzadko profile mogą być zmieniane dzięki podmianie oprogramowania *firmware* modułu, ale trzeba pamiętać też o tym, że są produkowane również moduły o specjalnym przeznaczeniu, np. do przesyłania sygnału audio z wysoką jakością.

Innym parametrem umożliwiającym rozpoznanie przeznaczenia modułu jest jego klasa, która określa moc nadajnika, a tym samym zasięg transmisji. Specyfikacja Bluetooth rozróżnia 3 klasy mocy:

- klasa *I* (10 dBm = 100 mW),
- klasa *II* (4 dBm = 2,5 mW),
- klasa *III* (0 dBm = 1 mW).

Teoretyczny zasięg modułu Bluetooth pracującego w klasie *I* wynosi około 150 metrów, natomiast w klasie *III* – około 2 metrów. Najbardziej rozpowszechnione są urządzenia pracujące w klasie *II*, które charakteryzują się zasięgiem około 10 metrów. Niestety, w wypadku transmisji radiowej można określić tylko przybliżony zasięg transmisji, ponieważ silnie zależy on od warunków środowiskowych.

Poszczególne specyfikacje Bluetooth różnicowały też jego przepustowość i tak maksymalna prędkość transmisji poszczególnych wersji protokołu wynosi:

- Bluetooth 1.0 – 21 kb/s,
- Bluetooth 1.1 – 124 kb/s,
- Bluetooth 1.2 – 328 kb/s,
- Bluetooth 2.0 – 2,1 Mb/s, w trybie *Enhanced Data Rate* – 3,1 Mb/s,
- Bluetooth 3.0 + HS (High Speed) – 24 Mb/s (3 MB/s),
- Bluetooth 3.1 + HS (High Speed) – 40 Mb/s (5 MB/s).

Jak można zauważyć, moduł może wspierać różne wersje standardu. Jest to ważne, że kolejne standardy nie wypierają poprzednich i jest zapewniona kompatybilność wsteczna. Warto dodać, że wiele nowości jest wprowadzanych jako rozszerzenia, których implementacja nie jest konieczna. Dotyczy to przede wszystkim rozszerzenia EDR (*Enhanced Data Rate*), które wprowadzono w wersji BT 2.0. Zwiększa ono szybkość transmisji do 3,1 Mb/s. W najnowszej wersji protokołu, BT 4.0, wprowadzono rozszerzenie o nazwie *Low Energy*. Warto wiedzieć, że nie wszystkie moduły komunikacyjne mają zaimplementowaną obsługę rozszerzeń, pomimo deklarowanej zgodności z daną wersją protokołu.

Idea profili powstała w celu zapewnienia kompatybilności pomiędzy urządzeniami z interfejsem Bluetooth pochodzącymi od różnych producentów. Wśród podstawowych można wymienić:

- **SPP** (*Serial Port Profile*) profil wirtualnego portu szeregowego opisuje wymagania związane z emulacją bezprzewodowego łącza szeregowego pomiędzy dwoma urządzeniami.
- **HSF** (*Hands-Free Profile*) profil „wolne ręce” opisuje urządzenie bramy wejściowej pozwalające na odbieranie i wykonywanie połączeń poprzez to urządzenie bez używania rąk. Typową aplikacją jest zestaw głośnomówiący umożliwiający użytkowanie telefonu w samochodzie.
- **HSP** (*Headset Profile*) niezwykle popularny profil stosowany do łączenia słuchawek bezprzewodowych z telefonem komórkowym.
- **HID** (*Human Interface Device Profile*) profil interfejsu człowiek – maszyna, popularny w bezprzewodowych klawiaturach, myszkach itp.

Określony profil jest dostępny jako odpowiedni *firmware*, czasami moduły różnią się jedynie oprogramowaniem. Zwykle rodzaj dostępnych profili można zidentyfikować za pomocą symbolu modułu Bluetooth.

Dla modułów tzw. multimedialnych, przeznaczonych do budowy toru transmisyjnego sygnału audio o bardzo dobrej jako-

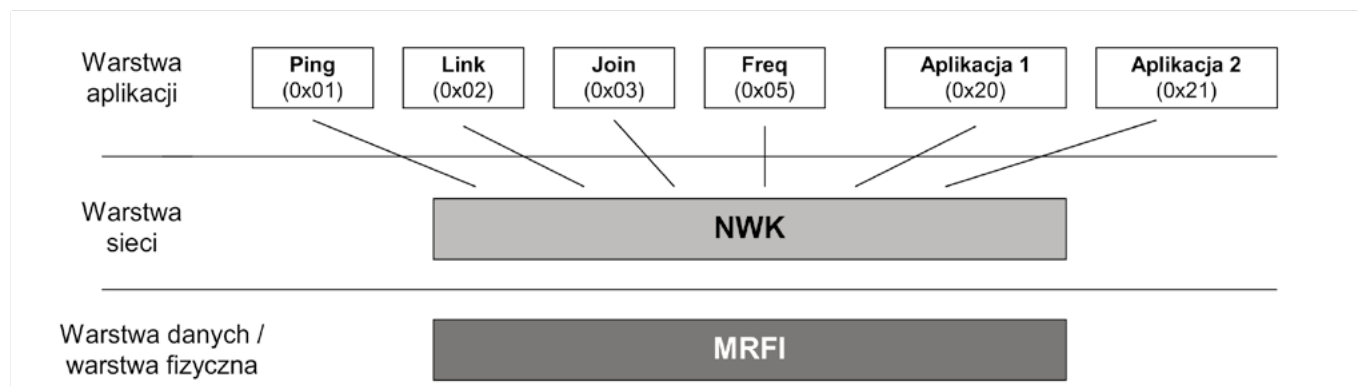
ści, szczególne znaczenie będą miały profile A2DP (*Advance Audio Distributor Profile*) oraz AVRCP (*Audio/Video Remote Control Profile*). Transmisja sygnału audio może przebiegać w trybie stereofonicznym lub monofonicznym. Dodatkowo, przesyłanie sygnału audio może być wspomagane przez układy dodatkowe, takie jak: koprocesory, pamięci o dużej pojemności, 16-bitowe kodeki audio oraz interfejs SPDIF. Pliki audio są kompresowane do formatu pozwalającego na ich przesyłanie bez straty jakości, z ograniczoną przepustowością do słuchawek lub głośników.

### Protokoły niestandardowe

W Elektronice Praktycznej wiele razy poruszyliśmy tematykę różnych niestandardowych protokołów sieciowych. Niektóre z nich powstawały na użytek pojedynczych projektów, inne są promowane przez firmy. Wśród nich warto wymienić IQRF czeskiej firmy Microrisc oraz SimpliCI międzynarodowego giganta Texas Instruments. Nie są one na tyle powszechne, aby były stosowane w każdym telefonie komórkowy, ale mimo to są bardzo interesujące dla elektronika – konstruktora i warto się z nimi zapoznać.

SimpliCI nie jest standardem, ale protokołem dla niewielkiej sieci opracowanym przez firmę Texas Instruments, który po zakupieniu firmy Chipcon kontynuuje rozwój wynalezionych i produkowanych przez nią układów radiowych. Zastosowania protokołu sieciowego SimpliCI obejmują podobne do ZigBee obszary. Wśród nich można wymienić: sieci czujników, urządzenia kontrolne i sterujące oraz służące do zdalnej identyfikacji i udostępnienia wyników pomiarów. Stos SimpliCI jest dostępny bezpłatnie. Jego użytkowanie nie wiąże się z opłatami licencyjnymi.

Strukturę warstwową oprogramowania SimpliCI pokazano na **rysunku 3**. Sieć SimpliCI może pracować w topologii gwiazdy lub peer-to-peer (**rysunek 2**). W tym pierwszym wypadku, centralnym węzłem komunikacyjnym jest punkt dostępowy. Wówczas centralny element zarządzający nie jest niezbędny a poszczególni uczestnicy sieci wy-



Rysunek 3. Struktura warstwową oprogramowania SimpliCI

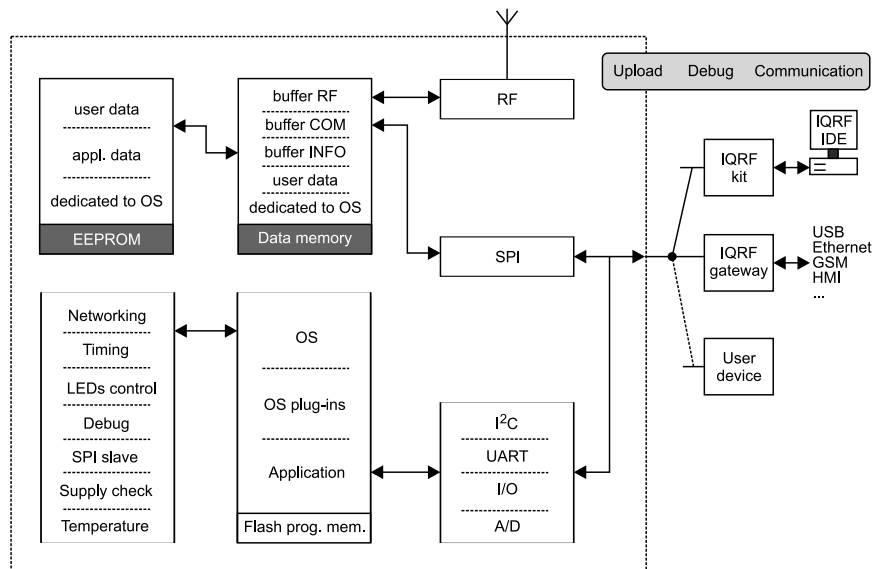
mieniają dane bezpośrednio. Sieć składa się z trzech rodzajów urządzeń:

- **Punkt dostępowy – Access Point (AP).** Zadaniem punktu dostępowego jest przydzielanie miejsca w sieci każdemu potencjalnemu uczestnikowi. Zarządza on również działaniem sieci. W określonych sytuacjach może też przechowywać dane dla innych węzłów i pośredniczyć w komunikacji. Z racji stałej aktywności wymaga zasilania sieciowego.
- **Retransmitter – Range extender (RE).** Jego zadaniem jest powtarzanie każdego odebranego prawidłowego pakietu. Liczebność tego typu urządzeń w sieci jest ograniczona do 4. Podobnie jak punkt dostępowy, wymaga zasilania sieciowego.
- **Urządzenie końcowe – End Device (ED).** Najczęściej występujący element sieci, identyfikowany z elementem wykonawczym. Nie uczestniczy w przekazywaniu komunikatów. Może utrzymywać kontakt radiowy z siecią okresowo (tzw. urządzenie śpiące – sleeping device). Ponieważ urządzenie śpiące nie nasłuchuje ciągle, dane do niego kierowane są przechowywane na okres pozostawania w uśpieniu przez punkt dostępowy. Może pracować z zasilaniem baterijnym. Dopuszcza się też możliwość istnienia urządzenia o jednokierunkowej transmisji (tylko nadawanie).

Funkcjonalność urządzenia sieciowego nie zawsze jest jednoznacznie identyfikowana z zadaniami urządzenia z punktu widzenia jego zastosowania w aplikacji. Punkt dostępowy i retransmitter mogą również posiadać cechy urządzenia wykonawczego np. czujnika czy włącznika oświetlenia.

W porównaniu z ZigBee sposób organizacji sieci wykazuje pewne uproszczenia. Zrezygnowano tutaj funkcjonalności *mesh*. Zaletą sieci SimpliCI jest możliwość przesyłania komunikatu dowolną drogą, wadą – znaczny przyrost wielkości kodu programu i zużycie energii. Sposób przenoszenia komunikatu ze źródła do celu jest ściśle określony. Maksymalna liczba skoków, która może pokonać komunikat, wynosi 4 i odpowiada scenariuszowi, zgodnie z którym komunikat pochodzący od urządzenia końcowego jest powtarzany przez retransmitter, po czym dociera do punktu dostępowego i poprzez drugi retransmitter do docelowego urządzenia końcowego. Każdy skok jest liczony i zapisywany w pakiecie. Zabezpiecza to sieć przed lawinowym narostem liczby powtarzanych komunikatów. Dodatkowo, retransmitery nie mogą komunikować się bezpośrednio. W wypadku połączenia peer-to-peer, gdy punkt dostępowy nie uczestniczy w transakcji, liczba skoków ograniczona do 2 (do retransmitera a potem do celu).

Począwszy od wersji oprogramowania 1.1.0 wprowadzono pewne modyfikacje. Sieć wspiera szyfrowanie komunikatów. W kon-



Rysunek 4. Architektura systemu IQRF OS

sekwencji niewielkiej zmianie uległa struktura pakietu danych. Wprowadzono również możliwość rozłączenia węzłów (unlink) oraz zachowania i odtworzenia tablicy połączeń z poziomu aplikacji. Poza tym, poprawiono pracę sieci w zawierającej retransmitery.

Czeska firma Microrisc produkuje moduły radiowe z wbudowanym mikrokontrolerem. Taki pomysł nie jest czymś niespotykanym, gdyby na nim poprzestano. Jednak wraz z modułami jest oferowane oprogramowanie – system operacyjny – zaprojektowany w celu budowania sieci urządzeń komunikujących się drogą radiową. Dodatkowo, firma oferuje zestawy ewaluacyjne, które pozwalają na łatwe przetestowanie aplikacji docelowych i sprawdzenie pomysłów na funkcjonowanie sieci.

W protokole komunikacyjnym zaimplementowano funkcje, które są niezbędne do transmisji danych pomiędzy modułami. Nie oznacza to jednak rezygnacji z maksymalnej dostępnej funkcjonalności ograniczonej zasobami mikrokontrolera. Oprócz możliwości wymiany danych pomiędzy dwoma modułami pracującymi w konfiguracji *peer-to-peer*, zaimplementowano również własny protokół sieciowy o nazwie IQMESH.

W pamięci mikrokontrolera modułu funkcjonuje preinstalowany system operacyjny IQRF OS, którego strukturę pokazano na **rysunku 4**. Użytkownik uruchamia swoją aplikację pod jego kontrolą i ma do dyspozycji wiele funkcji systemowych. Program użytkownika jest pisany w języku C i kompilowany za pomocą kompilatora CC5X. Trzeba jednak pamiętać, że taki moduł jest „inteligentnym” układem peryferyjnym i nie są na nim uruchamiane żadne duże, pamięciożerne zadania.

Tworzenie aplikacji umożliwia firmowy pakiet IRQF IDE. Za jego pomocą można wykonać wszystkie czynności projektowe: edytować (z użyciem zewnętrznego edytora

tekstowego) plik źródłowy, skompilować go, zaprogramować mikrokontroler modułu radiowego i wyszukiwać błędy w działającym programie. Programator modułów umożliwia zaprogramowanie pamięci Flash mikrokontrolera z poziomu IRQF IDE za pomocą pliku wynikowego.

### Podsumowanie

Mimo iż uproszczone opisy nie wyczerpują wszystkich aspektów technicznych, to pozwolą na zorientowanie się „co w trawie piszczy”. Z praktyki konstruktorskiej mogę powiedzieć, że budowa własnego modułu nadajnika, odbiornika, opracowanie własnych protokołów sieciowych, rzadko się opłaca. Tym mniej, im więcej funkcji sieciowych ma być zaimplementowanych w module. Stosowanie fabrycznych, gotowych rozwiązań ma duży sens praktyczny, umożliwia wymianę danych z innymi urządzeniami zgodnymi ze standardem np. ZigBee. Pytanie „własny czy opracowany samodzielnie?” warto sobie zadać tylko przy produkcji masowej. Wówczas to różnica w cenie może zrekompenzować nakłady poniesione na opracowanie modułu i protokołu transmisyjnego. Jednak w wypadku aplikacji produkowanych nawet w setkach sztuk, po prostu czasu. Problemów do rozwiązania będzie naprawdę dosyć i bez tego.

Jacek Bogusz, EP

### Bibliografia:

- <http://pl.wikipedia.org/wiki/ZigBee>
- <http://pl.wikipedia.org/wiki/Bluetooth>
- EP 5/2009, str. 109: J. Bogusz „Kto ma pszczoły, ten ma miód”
- EP 5/2009, str. 66: M. Kaczor, S. Nowocień, P.Tadrzak „SimpliCI. Protokół małej sieci radiowej”
- EP 9...12/2011, EP 3...5/2012 T. Jabłoński „IQRF więcej niż radio”