

Bezprzewodowy system automatyki bazujący na protokole SimpliciTI

Piotr Szymczyk
Magdalena Szymczyk

W artykule przedstawiono w zarysie protokół SimpliciTI wraz z przykładem zastosowania go do budowy bezprzewodowych sieci systemów automatyki. Zamieszczono także projekt węzła końcowego sieci przy wykorzystaniu modułu radiowego firmy Texas Instruments i mikrokontrolera firmy Atmel.

W bezprzewodowych systemach automatyki wykorzystywane są różne sposoby przekazywania danych, z zastosowaniem różnych częstotliwości pasma a także różnych protokołów. Wśród nich można wyróżnić przekaz radiowy na częstotliwościach 434 MHz, 868 MHz oraz 2,4 GHz z grupy nielicencjonowanych częstotliwości pasma ISM (ang. *Industrial Scientific Medical*) [1]. Wykorzystanie urządzeń krótkiego zasięgu w tych częstotliwościach nie wymaga zezwolenia, są zatem chętnie i dość często stosowane. Urządzenia pracujące na częstotliwości 434 MHz mają największy zasięg – nawet do kilkunastu kilometrów [1]. W chwili obecnej różni producenci oferują bardzo tanie układy scalone przeznaczone do budowy torów komunikacji bezprzewodowej. Mogą to być np. układy ADF70xx firmy Analog Devices, ATAxxxx firmy Atmel, CCxxxx firmy Texas Instruments. Moduły te są wyposażone w kompletny układ nadawczo-odbiorczy, który należy uzupełnić o kilka elementów zewnętrznych. Niektóre z nich mają wbudowany mikrokontroler (na przykład 8051). Projektując system bezprzewodowy należy założyć, że będzie pracował on w środowisku o dużym zagęszczeniu ruchu radiowego [6]. Dlatego też warto rozważyć dobór odpowiedniego protokołu transmisji, aby projektowane rozwiązanie było niezawodne oraz ekonomiczne. Bardzo popularnym protokołem w takich zastosowaniach jest protokół ZigBee [5], zaprojektowany specjalnie z myślą o budowie sieci bezprzewodowych czujników, urządzeń sterujących i monitorujących. Protokół ma wiele zalet i stał się w pewnym sensie standardem. Do jego wad należy zaliczyć nadmiarowość oraz duży koszt licencji [6]. Można oczywiście stworzyć własny protokół, ale wymaga to znacznego nakładu pracy i czasu, a efekt niekoniecznie będzie zadowalający. W takiej sytuacji ciekawą alternatywą staje się protokół SimpliciTI przygotowany przez Texas Instruments [4]. Jest on szczególnie atrakcyjny, gdy projektowane są: małe sieci, a wymagane są łatwość komunikacji oraz oszczędność energii (ze względu na zasilanie bateryjne niektórych urządzeń).

dr inż. Piotr Szymczyk, dr inż. Magdalena Szymczyk,
Akademia Górniczo-Hutnicza w Krakowie,
Katedra Automatyki

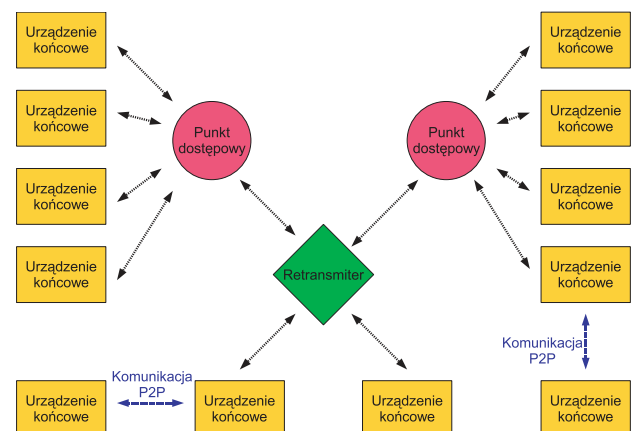
Protokół SimpliciTI

SimpliciTI został zaprojektowany przez Texas Instrument (TI) z myślą o aplikacjach bezprzewodowego sterowania, nadzoru dostępu, ochrony przed włamaniem, systemów alarmowych, układów wykrywania zagrożeń (dym, pożar, gaz), pomiarów (zużycia gazu, wody, prądu). Liczba węzłów sieci zawiera się w przedziale od 2 do 256. Charakteryzuje się dużą prostotą, niezawodnością, małym kosztem wdrożenia. Umożliwia komunikację w topologii gwiazdy lub punkt-punkt. Ma bardzo prosty interfejs API (ang. *Application Programming Interface*). Umożliwia oszczędzanie energii, ponieważ wspiera komunikację z urządzeniami, które okresowo przechodzą w stan „uśpienia” w celu oszczędzania baterii. Jego wymagania dotyczące zasobów są bardzo skromne i zależą od zastosowanego układu – od 4 KB do 8 KB pamięci flash oraz od 256 B do 512 B pamięci RAM [3]. Sieć zapewnia transmisję z szybkością do 250 Kbps i pracuje na częstotliwościach 434 MHz, 480 MHz, 868 MHz, 915 MHz, 955 MHz oraz 2,4 GHz.

W skład przykładowej sieci SimpliciTI pokazanej na rys. 1 wchodzi następujące rodzaje urządzeń:

- punkt dostępowy (ang. *Access Point* – AP)
- retransmiter (ang. *Range Extender* – RE)
- urządzenia końcowe (ang. *End Device* – ED).

Możliwe są połączenia przez punkt dostępowy, a także bezpośrednie połączenia pomiędzy urządzeniami końcowymi typu punkt-punkt (ang. *peer-to-peer* – P2P).



Rys. 1. Topologia sieci SimpliciTI

Punkt dostępowy zarządza działaniem sieci, pośredniczy w komunikacji pomiędzy urządzeniami, buforuje komunikację do „uśpionych” urządzeń. Z racji swojej funkcji powinien działać ciągle.

Retransmitter odbiera pakiety i następnie wysyła je dalej, umożliwiając zwiększenie zasięgu terytorialnego danej sieci. Liczba retransmitterów na drodze pakietu ograniczona jest do czterech ze względu na ochronę sieci przed lawiną powtarzanych pakietów, ograniczona jest również do czterech liczba samych retransmitterów w danej sieci.

Urządzenie końcowe odbiera pakiety do niego kierowane oraz wysyła pakiety do innych urządzeń w sieci. Urządzenie takie może okresowo przechodzić w stan „uśpienia” (ang. *sleeping device*). W tym czasie nie odbiera i nie nadaje żadnych komunikatów. Po powrocie do normalnego stanu pracy urządzenie odbiera, adresowane do niego, buforowane przez punkt dostępowy pakiety. Może również tylko wysyłać dane bez możliwości odbierania pakietów z sieci. Występuje to w przypadku prostych elementów, które dane z czujników wysyłają i nie mają potrzeby komunikacji dwukierunkowej.

Możliwe jest zbudowanie sieci, w której wszystkie urządzenia są zasilane bateryjnie i wszystkie mogą przechodzić w stan „uśpienia”. Sieć taka nie ma punktu dostępowego, a urządzenia muszą mieć tak dobrane czasy retransmisji danych, aby istniał przedział czasu, w którym obydwie są w stanie aktywnym i możliwe jest nawiązanie komunikacji między nimi. Węzeł sam decyduje jak długo pozostaje w stanie uśpienia, a jak długo jest aktywny – protokół nie narzuca tu żadnych ograniczeń. Jeśli sprzęt umożliwia szyfrowanie, to użytkownik może z niego skorzystać, w przeciwnym przypadku można użyć standardowego szyfrowania z symetrycznym kluczem, jakie zapewnia oprogramowanie [3]. Rys. 2 pokazuje format pakietu SimpliCI, który składa się z następujących elementów:

- PREAMBLE – preambula dodawana przez sprzęt nadawczo-odbiorczy
- SYNC – słowo synchronizacyjne dodawane przez sprzęt nadawczo-odbiorczy



Rys. 2. Format pakietu SimpliCI

Tab. 1. Aplikacje sieciowe

Aplikacja	Port	Opis
ping	0x01	Sprawdza, czy urządzenie o określonym adresie działa
link	0x02	Nawiązuje połączenie między dwoma urządzeniami
join	0x03	Podłącza urządzenie do sieci zarządzanej przez AP
security	0x04	Wymienia informacje na temat szyfrowania(klucz i kontekst szyfrowania)
freq	0x05	Zarządza zmianą częstotliwości pracy sieci
mgnt	0x06	Zarządza urządzeniem oraz pozwala na odbiór danych przez urządzenie, które wyszło ze stanu uśpienia

- LENGTH – pole jednobajtowe określające w bajtach długość pakietu (zależy od ilości danych w nim przesyłanych)
- MISC – pole zawierające różne informacje zależne od zastosowanego sprzętu nadawczo-odbiorczego, może nie występować
- DSTADDR – czterobajtowy adres odbiorcy
- SRCADDR – czterobajtowy adres nadawcy
- PORT – pole jednobajtowe, w którym bit 7 jest ustalany przez punkt dostępowy i określa czy pakiet był przekazywany przez retransmitter, bit 6 specyfikuje czy występuje szyfrowanie, a bity 5–0 definiują port aplikacji
- DEVICE INFO – pole jednobajtowe określające typ urządzenia oraz ile skoków wykonał pakiet w trakcie transmisji w sieci
- TRACKID – jednobajtowa liczba porządkowa komunikatu
- APP PAYLOAD – dane aplikacji o długości od 0 B do 111 B (dla transmisji zgodnej ze standardem IEEE 802.15.4) lub od 0 B do 50 B w przeciwnym przypadku
- FCS – suma kontrolna pakietu.

Oprogramowanie protokołu SimpliCI ma strukturę warstwową (warstwa danych/warstwa fizyczna, warstwa sieci i warstwa aplikacji).

W warstwie danych tego protokołu jest zaimplementowane m.in. sprawdzanie zajętości kanału (ang. *Clear Channel Assessment – CCA*), które umożliwia przydział wolnego kanału potrzebnego do transmisji danych. CCA wykrywa zajętość kanału na podstawie co najmniej jednego z trzech kryteriów:

- poziom sygnału na danym kanale
- wykrycie fali nośnej na danym kanale
- wykrycie fali nośnej z energią powyżej wartości progowej na danym kanale.

SimpliCI umożliwia zmianę częstotliwości pracy (ang. *frequency agility*), natomiast nie wspiera mechanizmu przeskakiwania po kanałach (ang. *frequency hopping*), gdyż o zmianie częstotliwości pracy decyduje punkt dostępowy. Mechanizm ten działa więc jedynie w sieci z punktem dostępowym. Urządzenia,

które transmitują tylko jednokierunkowo (do punktu dostępowego) i nie odbierają żadnych transmisji z sieci, nie reagują na takie polecenia punktu dostępowego, ale nadają komunikaty na wszystkich dostępnych częstotliwościach pracy sieci.

W warstwie aplikacji umieszczono zarówno aplikacje użytkownika, jak i aplikacje sieci. Aplikacje użytkownika mogą używać portów od 0x20 do 0x3F, porty od 0x00 do 0x1F są zarezerwowane i używane przez aplikacje sieciowe [2].

Aplikacje użytkownika mają do dyspozycji następujące funkcje API:

- inicjalizujące pracę sprzętu i węzła sieci:
 - BSP_Init() - inicjalizacja pracy sprzętu
 - SMPL_Init() - inicjalizacja pracy urządzenia w sieci
- zestawiające i rozłączające połączenie:
 - SMPL_Link() - próba natychmiastowego zestawienia połączenia pomiędzy dwoma urządzeniami bez możliwości oczekiwania na takie zestawienie
 - SMPL_LinkListen() - próba zestawienia połączenia między dwoma urządzeniami z określonym czasem oczekiwania na takie zestawienie
 - SMPL_Unlink() - rozłączenie połączenia
- wysyłające i odbierające dane:
 - SMPL_Send() - wysłanie danych
 - SMPL_SendOpt() - wysłanie danych z określonymi opcjami transmisji
 - SMPL_Receive() - odebranie danych
- konfiguracyjne i inne:
 - SMPL_Ioctl() - uniwersalna funkcja ustawiająca i odczytująca różne parametry konfiguracyjne
 - SMPL_Ping() - sprawdza, czy urządzenie o określonym adresie działa
 - SMPL_Commission() - zaawansowana funkcja do tworzenia wpisów w tabeli połączeń; użycie wymaga szczegółowej wiedzy na temat sieci.

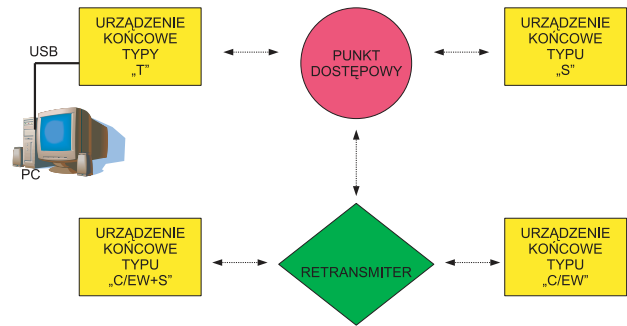
Funkcje te są proste w użyciu i zapewniają pełną obsługę sieci z poziomu aplikacji użytkownika, szczegółowy opis funkcji oraz struktur danych można znaleźć w dostępnej dokumentacji protokołu SimpliTI [3, 7].

Bezprzewodowa sieć systemu automatyki

Na bazie sieci SimpliTI można zbudować bezprzewodową sieć systemu automatyki. Jednym z elementów takiej sieci są urządzenia końcowe, które w zależności od potrzeby mogą odczytywać dane z czujników, przekazywać dane do elementów wykonawczych oraz wyznaczać sterowanie.

Na rys. 3 pokazano przykładową sieć bezprzewodowego systemu automatyki, w której znajduje się punkt dostępowy zarządzający pracą sieci oraz pośredniczący w wymianie danych między węzłami sieci będącymi urządzeniami końcowymi. Znajduje się tu również retransmitter umożliwiający zwiększenie zasięgu działania sieci oraz węzeł końcowy z podłączonym komputerem PC, z którego można monitorować i zarządzać pracą całego systemu automatyki. Węzły końcowe mogą być następujących typów:

- T - terminalowy, umożliwiający przyłączenie komputera PC do sieci
- S - sterujący, odbierający dane z innych węzłów, wyznaczający sterowanie oraz wysyłający dane do innych węzłów w celu realizacji obliczonego sterowania
- C - czujnik, odczytujący wartości z czujników i wysyłający je do węzła obliczającego sterowanie



Rys. 3. Przykładowa sieć bezprzewodowa systemu automatyki

EW - element wykonawczy, odbierający dane od węzła wyznaczającego sterowanie i przekazujący je do elementów wykonawczych

C/EW - węzeł pełniący funkcje zarówno węzła typu C, jak i węzła typu EW

C+S - węzeł pełniący funkcje zarówno węzła typu C, jak i węzła typu S

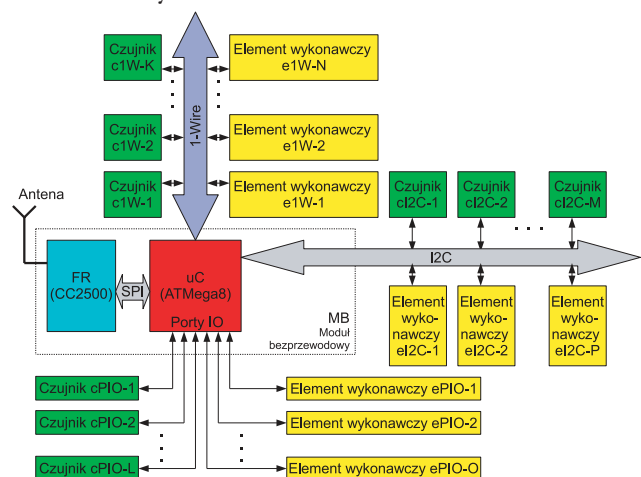
EW+S - węzeł pełniący funkcje zarówno węzła typu EW, jak i węzła typu S

C/EW+S - węzeł pełniący funkcje zarówno węzła typu C/EW, jak i węzła typu S.

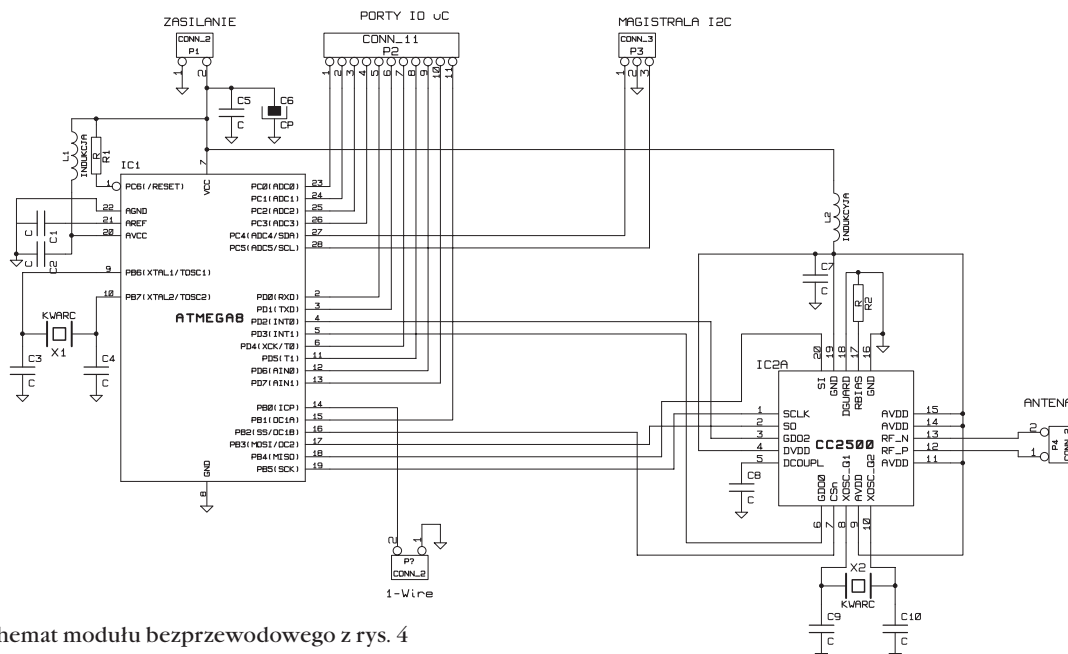
W sieci może znajdować się wiele węzłów końcowych danego typu, możliwe jest też wykonywanie obliczeń związanych ze sterowaniem na komputerze PC przyłączonym do tej sieci za pomocą węzła typu T.

Uniwersalny węzeł końcowy pokazany na rys. 4 stanowi bazę do budowy wszystkich typów węzłów końcowych z wyjątkiem węzła typu T. Składa się z modułu bezprzewodowego z blokiem radiowym CC2500 oraz mikrokontrolerem ATmega8. Do mikrokontrolera mogą być dołączone, poprzez magistralę I2C lub magistralę 1-Wire, urządzenia zewnętrzne takie jak: czujniki, elementy wykonawcze itp. Możliwe jest także bezpośrednie podłączenie tego typu elementów do portów wejścia-wyjścia mikrokontrolera.

Na rys. 5 pokazany jest schemat elektroniczny realizacji modułu bezprzewodowego będącego zasadniczą częścią uniwersalnego węzła końcowego. Jak widać moduł taki jest bardzo prosty i zawiera bardzo mało elementów, co w efekcie powoduje, że jest bardzo tani i niezawodny.



Rys. 4. Uniwersalny węzeł końcowy sieci z modulem bezprzewodowym (MB)



Rys. 5. Schemat modułu bezprzewodowego z rys. 4

Aby mógł funkcjonować, należy oczywiście go zaprogramować. Programowanie modułu jest bardzo łatwe, a fragment kodu programu w języku C dla uniwersalnego węzła końcowego wygląda następująco:

```
//...
int endDevice()
{
    linkID_t link_to_Master;
    // INITIALIZATION
    BSP_Init();
    if (SMPL_Init(NULL) != SMPL_SUCCESS)
        return (ERROR_INIT);
    if (SMPL_Link(&link_to_Master) != SMPL_SUCCESS)
        return (ERROR_LINK);
    // MAIN LOOP
    while (TRUE)
    {
        //...
        readDataFromMaster();
        readDataFromSensors();
        calculateControl();
        sendDataToActuators();
        sendDataToMaster();
    }
}
```

Przedstawiona powyżej funkcja endDevice() składa się z dwóch części. Pierwsza inicjalizuje pracę węzła i zestawia połączenie, a druga to pętla nieskończona, w której sekwencyjnie następuje odbiór danych z węzła nadrzędnego, odczyt danych z czujników, obliczenie sterowania, wysłanie danych do elementów wykonawczych i do węzła nadrzędnego.

Poszczególne węzły końcowe mogą być zasilane z zasilaczy sieciowych lub z baterii. Dla węzłów zasilanych z baterii należy zastosować specjalne metody oszczędzania energii, polegające na „usypianiu” węzła, szczegółowo opisane w literaturze [8]. Dzięki takiemu rozwiązaniu węzeł zasilany bateryjnie może pracować przez długi okres (rzędu miesięcy lub lat), a w przypadku wyczerpywania się baterii sam powiadomić o tym fakcie nadzór nad siecią.

Podsumowanie

Bezprzewodowe systemy automatyki można projektować opierając się na protokole SimpliciTI. Protokół ten jest prosty w użyciu, posiada wsparcie dla urządzeń zasilanych z baterii, okresowo przechodzących w stan „usypienia”. Do budowy takich sieci można stosować tanie urządzenia wykorzystujące moduły radiowe firmy Texas Instruments oraz mikrokontrolery firmy Atmel. Szacunkowy koszt węzła to około 100 zł. Rozwiązanie takie zapewnia dużą elastyczność, a tak zbudowany węzeł sieci może obsługiwać czujniki i elementy wykonawcze podłączone bezpośrednio do portów mikrokontrolera lub za pośrednictwem magistrali 1-Wire lub magistrali I2C.

Bibliografia

1. Bogusz J.: *Partnerstwo bezprzewodowe (1). Układy scalone do transmisji danych w paśmie ISM*, Elektronika Praktyczna, maj 2009.
2. Fieldman L.: *SimpliciTI: Simple Modular RF Network Developers Notes*, Texas Instruments Inc., San Diego 2009.
3. Fieldman L.: *SimpliciTI: Simple Modular RF Network Specification*, Texas Instruments Inc., San Diego 2009.
4. <http://focus.ti.com/docs/toolsw/folders/print/simpliciti.html>
5. <http://www.zigbee.org/>
6. Kaczor M., Nowocień S., Tadrzak P.: *SimpliciTI Protokół małej sieci radiowej*, Elektronika Praktyczna, maj 2009.
7. SimpliciTI Application Programming Interface, Texas Instruments Inc., San Diego 2009.
8. Szymczyk P., Szymczyk M.: *Energooszczędne oprogramowanie systemów czasu rzeczywistego mikrokontrolerów*, XVI Konferencja SCR'09, Pułtusk, 14–17 września 2009. ■