

Zygmunt KUBIAK

*Instytut Informatyki
Politechnika Poznańska*

SIMPLICITI™ – PROTOKÓŁ DLA PROSTYCH SIECI RADIOWYCH

SIMPLICITI™ - A PROTOCOL FOR SIMPLE RADIO NETWORKS

The paper presents the SimpliCiTI radio protocol. It is a simple low-power RF network protocol aimed at small RF networks. Such networks typically contain battery operated devices which require long battery life, low data rate and low duty cycle and have a limited number of nodes. End devices exchange messages directly to each other or through an access point or range extenders. Access point and range extenders are not required but provide extra functionality such as store and forward messages. SimpliCiTI main features are: low memory needs (<8kB flash and 1kB ram depending on the configuration), advance network control (security, radio frequency agility, ...), sleeping modes support. Typical application areas for SimpliCiTI are alarm and security systems, home automation systems as well as smart measurements and AMR (Automatic Meter Reading). These areas are compatible with the systems of environmental protection and water management.

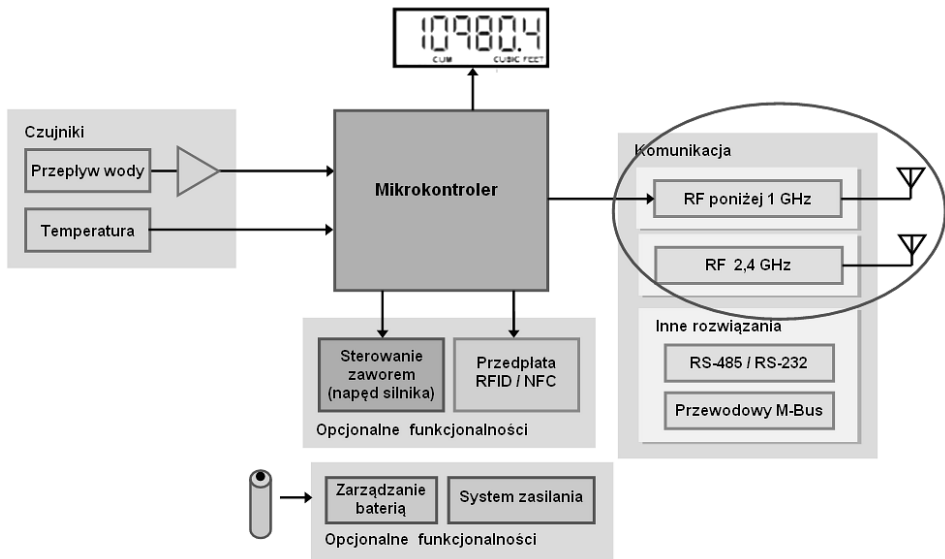
1. Wprowadzenie

Rosnąca na świecie produkcja specjalizowanych elementów elektronicznych jak mikrokontrolery, sensory, radiowe układy nadawczo-odbiorcze małej mocy itp., powoduje spadek ich cen a to stymuluje rozwój i modernizację a także tworzenie i realizację nowych pomysłów, między innymi w dziedzinie systemów pomiarowych i sterujących. Z drugiej strony zastosowania intensyfikują produkcję. Następują rewolucyjne zmiany w systemach zarządzania mediami, takimi jak energia elektryczna, woda, gaz, energia cieplna, niestety w Polsce z pewnym opóźnieniem [1]. Pojawiły się nowe zadania: inteligentne pomiary (ang. *Smart Metering*) oraz w odniesieniu głównie do energetyki, inteligentne sieci (ang. *Smart Grid*) [1,3]. Powinny one między innymi umożliwić świadome gospodarowanie mediami. Z inteligentnymi pomiarami wiąże się bezpośrednio koncepcja sieci domowej HAN (ang. *Home Area Networks*) oraz tzw. Internet rzeczy (przedmiotów) (ang. *Internet of Things*). Jednym z zadań HAN jest zapewnienie komunikacji między domowymi odbiornikami energii a elektronicznymi licznikami, monitorowanie i sterowanie w założonym zakresie, tymi urządzeniami. Internet rzeczy jest to koncepcja, w której przedmioty takie jak urządzenia gospodarstwa domowego, oświetle-

niowe, grzewcze, wodociągowe, kanalizacyjne mogą pośrednio lub bezpośrednio gromadzić, przetwarzać lub wymieniać dane za pośrednictwem sieci komputerowej.

W związku z wyżej przedstawionymi zadaniami rośnie znaczenie sieci radiowych małej mocy typu WSN (ang. *Wireless Sensor Network*). Węzły takich sieci, w uproszczeniu nazywane sensorami, charakteryzują się np. tym, że mogą pracować bez wymiany baterii nawet kilkanaście lat [2]. W niektórych rozwiązaniach sieci WSN do zasilania węzłów pozyskuje się energię z otoczenia. Jednym z prostszych rozwiązań dla bezprzewodowych sieci sensorowych jest protokół *SimpliciTI™*, firmowany przez Texas Instruments [4,5,6].

SimpliciTI™ jest protokołem transmisji radiowej niskiej mocy przeznaczonym dla realizacji małych, stacjonarnych sensorowych sieci bezprzewodowych. Przykładowe zastosowania to automatyka domowa, systemy alarmowe, w tym realizacja węzłów, np. z czujnikami obecności, czujnikami dymu, czujnikami światła, czujnikami zbitcia szkła. Z powodzeniem może być stosowany w systemach inteligentnych pomiarów, np. w licznikach energii elektrycznej, gazomierzach, miernikach ciepła i wodomierzach. Stanowi również bardzo dobre rozwiązanie dla rozproszonych punktów pomiarowych w systemach ochrony środowiska, oczyszczalniach ścieków i gospodarki wodnej. Przykład miejsca węzłów sieci *SimpliciTI™* pokazano na rys. 1.



Rys. 1. Transmisja radiowa w systemie pomiaru przepływu wody, na podstawie propozycji Texas Instruments [8]

Fig. 1. Radio transmission in the measurement system of water flow, on the basis of the proposals Texas Instruments

2. Protokół SimpliciTI™

Protokół ten dedykowany jest wyłącznie dla układów radiowych (RF) małej mocy oferowanych przez firmę Texas Instruments, tzn. zintegrowanych z mikrokontrolerem radiowych układów nadawczo-odbiorczych SoC (*System-on-chip*) oraz mikrokontrolerów serii MSP430 w połączeniu z układami RF typu CC1xxx/CC25xx. Protokół zapewnia sieci wiele interesujących cech, np. zmiana częstotliwości (ang. *Frequency Agility*) pracy w przypadku zajętości kanału. Zmiany wprowadzane są przy użyciu tabel częstotliwości. Od wersji protokołu SimpliciTI™ 1.1.0 wprowadzono szyfrowanie wspierane programowo lub sprzętowo (jeżeli układ RF posiada takie cechy).

Sieć składa się z trzech rodzajów urządzeń:

- Punkt dostępowy - AP (ang. *Access Point*)
Jest węzłem zarządzającym siecią. Może też przechowywać dane dla innych węzłów i pośredniczyć w komunikacji również z urządzeniami wyższego poziomu jak np. komputer. Ponieważ wymagana jest w zasadzie stała aktywność AP, w związku z tym potrzebne jest ciągłe zasilanie.
- Węzeł retransmisji – RE (ang. *Range extender*)
Zadaniem węzła jest powtarzanie każdego poprawnie odebranego pakietu. W jednej sieci mogą być stosowane maksymalnie cztery urządzenia RE. Skoki retransmisji są zliczane i zapisywane w pakiecie co zabezpiecza przed dużą liczbą powtarzanych pakietów krążących w sieci. Podobnie jak dla punktu dostępowego, wymagane jest ciągłe zasilanie, z reguły sieciowe.
- Urządzenie końcowe – ED (ang. *End Device*)
Urządzenie końcowe jest podstawowym węzłem występującym w sieci. Węzły ED mogą realizować funkcje sensorowe lub/i wykonawcze. Pracują okresowo, przez większość czasu pozostają w stanie uśpienia. Nie biorą udziału w retransmisji. Na czas uśpienia komunikaty przeznaczone dla węzła ED przechowywane są w punkcie dostępowym. Węzeł zasilany jest z reguły bateryjnie. Dopuszcza się transmisję jednokierunkową (nadawanie).

Sieć SimpliciTI™ może pracować w topologii gwiazdy, gdzie koordynatorem jest punkt dostępowy lub w topologii „peer-to-peer”(rys.2). W takim przypadku punkt dostępowy nie jest potrzebny a poszczególni uczestnicy sieci wymieniają dane bezpośrednio.

Oprogramowanie SimpliciTI™ ma strukturę warstwową, ograniczoną do trzech warstw (rys. 3). Programista musi w zasadzie rozwinąć jedynie warstwę aplikacji. Obsługa komunikacji między modułami realizowana jest za pomocą prostych funkcji API, używanych do inicjalizacji i konfiguracji sieci oraz do odbierania i wysyłania wiadomości [6].

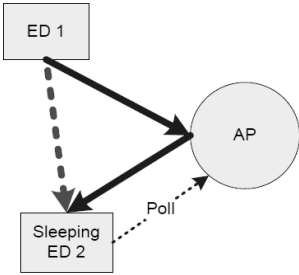
Warstwa łącza danych/fizyczna (ang. *Data Link/PHY*)

MRFI (ang. *Minimal Radio Frequency Interface*) odpowiada za komunikację z układem radiowym. Zapewnia odczyt / zapis ramki danych między aplikacją interfejsu a układem RF. Różne układy radiowe wspierane przez SimpliciTI™ wymagają różnych implementacji, ale podstawowy interfejs oferowany w warstwie sieciowej jest taki sam dla wszystkich RF. MRFI tworzy w rezultacie standardowy interfejs dla różnych rozwiązań układów nadawczo-odbiorczych. Z warstwą tą związane jest również oprogramowanie BSP (ang. *Board Support Package*), które stanowi pewne wsparcie dla różnych docelowych mikrokontrolerów. Uwzględnia obsługę interfejsu SPI (ang. *Serial Peripheral Interface*) do komunikacji z układem RF, realizuje zarządzanie przerwaniem poprzez połączenia GPIO

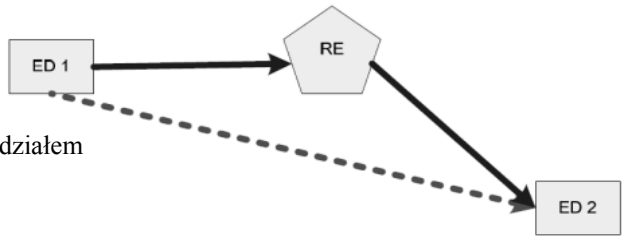
(ang. *General Purpose Input/Output*) aby wspierać asynchroniczne komunikaty od RF (np. pusty bufor nadawczy, odebrano pakiet), ponadto wspiera sterowanie diodami LED oraz obsługę przycisków.



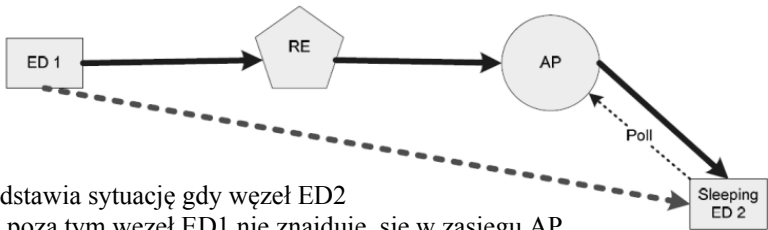
Bezpośrednie połączenie peer-to-peer



Przykład konfiguracji, w której AP pośredniczy w transmisji. Przechowuje wiadomość od ED1 dla ED2 do czasu wybudzenia ED2. Funkcja ta nazywa się *Store-and-forward*.



Komunikacja pośrednia z udziałem węzła RE



Rysunek przedstawia sytuację gdy węzeł ED2 jest uśpiony a poza tym węzeł ED1 nie znajduje się w zasięgu AP.

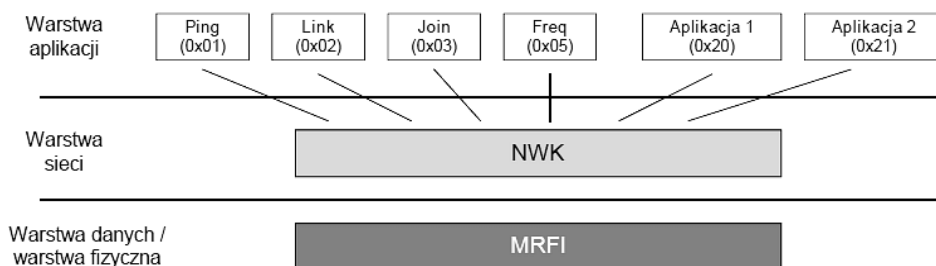
- ścieżka logiczna
- ścieżka fizyczna
- zarządzanie przez warstwę NWK

Rys. 2. Topologie komunikacyjne wspierane przez *SimpliciTI*™ [5]

Fig. 2. Topologies supported by *SimpliciTI*™

Warstwa sieciowa (ang. *Network Layer*)

NWK odpowiada za określone zadania związane z nawiązaniem i podtrzymaniem komunikacji oraz realizuje zadania zlecone w warstwie aplikacyjnej. Warstwa sieciowa zajmuje się uzupełnianiem pakietów o dane dotyczące typu urządzenia i zliczaniem skoków a także kolejkuje dane przychodzące z warstwy aplikacji i buforuje je, jeśli odbiorca nie jest aktywny (przechowywanie danych przez punkt dostępowy dla śpiącego urządzenia końcowego). Ustawienia sieciowe są przygotowywane na etapie kompilacji programu i obejmują takie parametry jak: kanał transmisji, liczba przechowywanych pakietów w AP dla urządzeń końcowych w uśpieniu, sieciowe klucze szyfrujące, dopuszczalna liczba skoków pakietów itp.



Rys. 3. Architektura SimpliTI™ [4,5]

Fig. 3. SimpliTI™ Architecture

Warstwa aplikacji (ang. *Application Layer*)

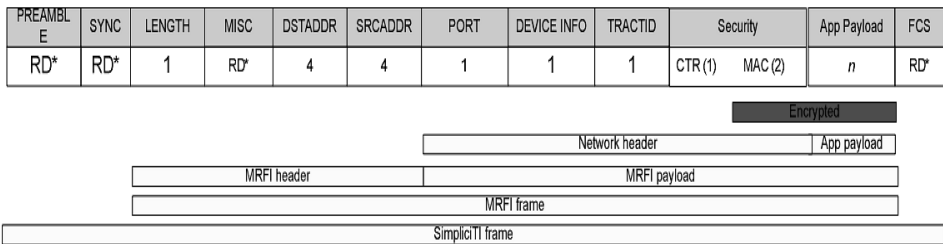
Aplikacje użytkownika korzystają z portów 0x20 i wyższych. Oprogramowanie SimpliTI™ udostępnia aplikacji użytkownika prosty interfejs API, zawierający 6 poleceń: inicjalizacja sieci – polecenie *SMPL_Init*; konfiguracja – *SMPL_Ioctl*; proces łączenia węzłów – dla inicjatora *SMPL_Link* oraz dla drugiej strony *SMPL_LinkListen*; transmisja – nadawanie *SMPL_Send* oraz odbiór *SMPL_Receive*. Ważnym etapem jest inicjalizacja sieci. Obejmuje ona trzy fazy: inicjalizacja mikrokontrolera, inicjalizacja układu RF oraz inicjalizacja stosu.

Porty o niższych numerach zapewniają poprawne funkcjonowanie sieci SimpliTI™. Polecenie **Ping** (0x01) używane jest do sprawdzenia obecności węzłów. **Link** (0x01) odpowiada za nawiązanie połączenia między węzłami. Podczas tworzenia połączenia węzły wymieniają się swoimi adresami sieciowymi i przypisują je do identyfikatora *LinkID*. Od tego momentu możliwa jest wymiana danych. **Join** (0x03) umożliwia przyłączenie węzła ED do sieci zarządzanej przez punkt dostępowy. Proces jest inicjowany przez urządzenie końcowe, które posługuje się predefiniowanym hasłem *Join Token* a po przyłączeniu otrzymuje od AP identyfikator połączenia. Węzły pracujące ze stanem uśpienia otrzymują dodatkowo bufor w AP do przechowywania na czas uśpienia, kierowanych do nich pakietów. Polecenie **Security** (0x04) dotyczy szyfrowania. **Freq** (0x05) odpowiada za zarządzanie częstotliwością pracy sieci. W przypadku dużego obciążenia kanału istnieje możliwość wyboru innego kanału z grupy częstotliwości predefiniowanych – mechanizm *Frequency Agility*. O wyborze decyduje węzeł AP, stąd też mechanizm ten może być stosowany w sieciach zawierających AP. W przypadku transmisji jednokierunkowej węzły końcowe

(nadajniki) nie mogą reagować na zmiany kanałów i w związku z tym nadają pakiety we wszystkich predefiniowanych kanałach. Polecenie **Management** (0x06) przez węzły pracujące z trybem uśpionia stosowane jest w celu zapytania o buforowane dane. Wykorzystywane jest też w komunikacji serwisowej.

W przypadku transmisji radiowej istotne jest zapewnienie jej poufności. Mechanizmem zapewniającym w tym zakresie bezpieczeństwo transmisji jest szyfrowanie. W SimpliTI™ począwszy od wersji 1.1.0 wprowadzono algorytm szyfrujący XTEA (ang. *Extended Tiny Encryption Algorithm*). Jest to szyfrowanie blokowe 64-bitowe z kluczem 128-bitowym.

Strukturę ramki przedstawiono na rys. 4. Ładunek z warstwy aplikacyjnej jest uzupełniany w warstwie NWK o nagłówek. W warstwie MRFI dochodzi kolejny nagłówek tworząc ramkę MRFI. W omawianym protokole zakłada się, że zadania warstwy fizycznej realizują układy RF. Układy radiowe sprzętowo uzupełniają pakiet o pole PREAMBL (zapewnia poprawną synchronizację bitową), pole SYNC (zapewnia synchronizację bajtową) oraz pole FCS (wprowadza nadmiarowość umożliwiającą kontrolę integralności pakietu).

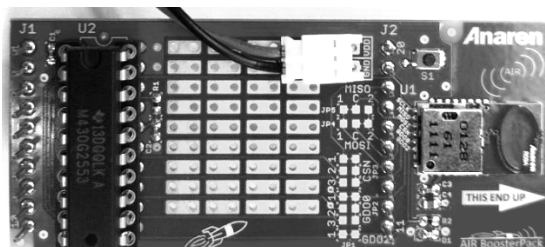


Rys. 4. Struktura ramki danych SimpliTI™ (z włączoną funkcją szyfrowania) [5]

Fig. 4. SimpliTI™ frame structure (with Security enabled)

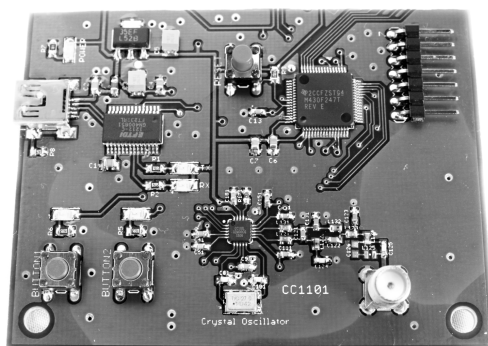
Programista budując aplikację użytkownika korzysta ze wspomnianego już interfejsu API [6]. Takie podejście znacznie upraszcza oprogramowanie sieci a realizowane na tym etapie zadania dotyczą głównie funkcjonalności tworzonego systemu. Funkcje API zostały już skrótowo przedstawione przy opisie warstwy aplikacji. Polecenie API może zwracać jedno ze słów statusowych, informujące o wyniku jego wykonania. Przykładowo słowo `SMPL_SUCCESS` oznacza poprawnie przeprowadzoną operację, `SMPL_TIMEOUT` oznacza przekroczenie limitu czasu wykonania a `SMPL_NO_LINK` – brak odpowiedzi na pakiet typu Link. Wcześniej należy przygotować funkcje inicjalizacji wybranego mikrokontrolera, zastosowanego układu nadawczo-odbiorczego oraz stosu (`BSP_Init()` i `SMPL_Init()`). Tworzenie połączeń sieci i ich utrzymanie realizowane jest przy pomocy funkcji połączeniowych (`SMPL_Link()` i `SMPL_LinkListen()`). Funkcje `SMPL_Send()`, `SMPL_SendOpt()` oraz `SMPL_Receive` realizują zadania wymiany danych między węzłami sieci, według ustalonego przez użytkownika scenariusza. Najbardziej rozbudowanym interfejsem, umożliwiającym zarządzanie pracą węzłów jest `SMPL_Iocf()`. Pozwala między innymi na wymuszanie stanu układu RF.

Na rys. 5 przedstawiono budowę uruchomieniowego węzła sieci SimpliCITI™ firmy Anaren, który zawiera moduł RF z układem CC110L oraz mikrokontroler MSP430G2553 firmy Texas Instruments. To rozwiązanie pozwala na realizację wszystkich typów węzłów, tzn. węzła końcowego, retransmisyjnego i punktu dostępowego. Pełna konfiguracja węzła końcowego wymaga uzupełnienia o niezbędne układy wejścia/wyjścia, np. sensory, natomiast punktu dostępowego np. o interfejs z komputerem. Na rys. 6 pokazano przykładowe rozwiązanie węzła AP.



Rys. 5. Przykład rozwojowego węzła sieci SimpliCITI™ firmy Anaren [7]

Rys. 5. Example of development node of the SimpliCITI™ network from Anaren



Rys. 6. Przykładowa budowa węzła AP

Fig. 6. An example of the construction of the AP node

3. Podsumowanie

W sieciach sensorowych wymaga się aby pobór mocy węzła (tu: końcowego) ze względu na zasilanie bateryjne był możliwie najmniejszy. W przypadku omawianej dziedziny zastosowań, węzeł ED może być sprzężony np. z licznikiem wody, przepływomierzem ścieków, czujnikiem poziomu, mętności lub czujnikami parametrów chemicznych. Aby

spełnić wymagania energooszczędności dla sieci WSN, węzły te przez większość czasu znajdują się w stanie uśpienia, wybudzają się na krótko w celu wykonania pomiarów i transmisji danych. Najwięcej energii zużywa transmisja. Jeżeli proces technologiczny wymaga częstszych pomiarów to wtedy ich wyniki są buforowane i przesyłane blokowo. Sieć SimpliTI™ ze względu na stosunkowo krótki kod programu (małe wymagania pamięciowe, krótszy czas wykonania programu) oraz prostą organizację wymiany danych jest w wyżej wspomnianych przypadkach zastosowań (sieci z węzłami stacjonarnymi) znacznie bardziej efektywna np. od bardziej uniwersalnej sieci ZigBee. W tab.1 przedstawiono porównanie wybranych cech protokołów SimpliTI™ i ZigBee.

Tab. 1. Porównanie protokołów SimpliTI™ i ZigBee

Tab.1. Comparison of SimpliTI™ and ZigBee protocols

Protokół	SimpliTI	ZigBee
Liczba węzłów	2...kilkadziesiąt	2...tysiące
Mesh	Nie	Tak
Rozmiar kodu	6 kB	50...60kB
Szyfrowanie	Nie ma w 1.06, możliwe sprzętowe AES-128 lub programowe	AES-128

Bibliografia

- [1] Kubiak, Z., Urbaniak A. Inteligentny system pomiarowy – rozwój standardu, przykładowe rozwiązania. Rynek Energii, nr 1 (104) 2013, s. 43-49
- [2] Kubiak, Z., Urbaniak A. Sieci bezprzewodowe dla systemów zdalnych pomiarów. Rynek Energii, nr 2 2010, s. 72-79
- [3] Kubiak, Z., Urbaniak A. Systemy monitorowania zużycia mediów w budynkach. Rynek Energii, nr 5 (84) 2009, s. 22-31
- [4] Friedman L. SimpliTI: Simple Modular RF Network, Developers Notes. Texas Instruments, San Diego, 2007-2009
- [5] Friedman L. SimpliTI: Simple Modular RF Network, Specification. Texas Instruments, San Diego, 2007-2009
- [6] SimpliTI, Application Programming Interface. SWRA221. Texas Instruments, San Diego, 2008-2009
- [7] <http://www.anaren.com/air/cc1101-air-module-boosterpack-embedded-antenna-module-anaren#overlay-context=air/air-boosterpack>
- [8] http://www.ti.com/solution/water_meter