

# SimpliciTI

## Protokół małej sieci radiowej

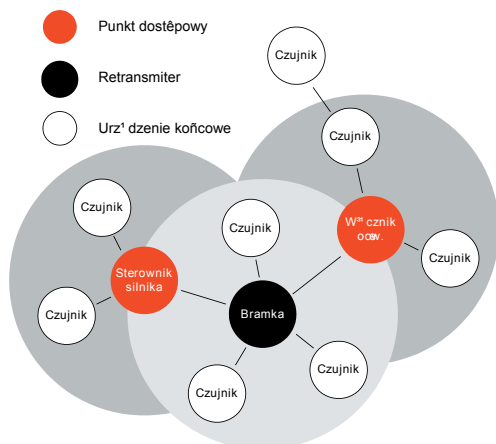
Rozwój technologiczny podzespołów do komunikacji bezprzewodowej spowodował, że obszar zastosowań łączności radiowej rozrasta się. Spora część aplikacji działa w obszarach o dużym zagęszczeniu ruchu radiowego, gdzie poszczególne węzły działają w zorganizowany sposób tworząc sieć, a to wymaga zastosowania protokołu sieciowego zdolnego zapanować nad potokiem informacji. Jednym z dostępnych rozwiązań jest protokół SimpliciTI przygotowany przez Texas Instruments dla odbiorców swoich transceiverów.

Urządzenia komunikujące się bezprzewodowo przypominają nieco załogę na cotygodniowym zebraniu. Jeśli załoga jest nieliczna a tematyka nieciekawa, to każdy chętny zdąży się wypowiedzieć. Jednak w dużej grupie krzykacze zagłuszają mniej asertywnych, niektórzy wchodzą sobie w słowo, niezainteresowani dyskutują na boku o czymś innym a istotne wypowiedzi nikną w ogólnym rozgardiaszu.

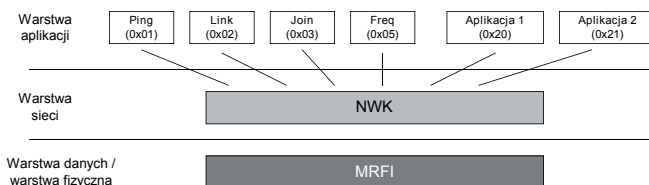
Protokół sieciowy to taki zbiór reguł, w których naturalne zalecenia, by nie przerywać wypowiedzi innych czy wypowiadać się, gdy poproszą, to tylko część obowiązujących zasad. Protokół ustala również wspólny język, co pozwala wszystkim się rozumieć i rozpoznać, gdy ktoś przeinacza fakty. Poza tym, ma jeszcze jedną istotną cechę: inaczej niż w społeczności ludzkiej, jest naprawdę przestrzegany.

Kiedy mówimy o standardach protokołów sieci bezprzewodowych nie sposób zignorować istnienia ZigBee. W powstaniu standardu ZigBee włożono wiele wysiłku a prace właściwie nie ustają. Od kilku lat ogłaszane są kolejne specyfikacje, usiłujące nadać za potrzebami. Użytkowanie standardu ma swoje zalety ale również kilka wad: nadmiarowość, duży koszt licencji ZigBee Alliance i obszerne biblioteki. Dlatego w wielu wypadkach lepszy okazuje się mały, sztywny na miarę, stos protokołu.

SimpliciTI nie jest standardem. Protokół został przygotowany przez Texas Instruments, który kontynuuje obecnie rozwój linii układów radiowych oferowanych przez norweskiego Chipcon. Zastosowa-



Rys. 1. Topologia sieci



Rys. 2. Struktura warstwowa oprogramowania

wania SimpliciTI obejmują też podobne, jak w przypadku ZigBee, obszary: to sieci czujników w środowisku domowym lub przemysłowym, urządzenia kontrolne i sterujące oraz zdalna identyfikacja i zdalny odczyt wyników pomiarów. Stos SimpliciTI dostępny jest bezpłatnie, podobnie jak Z-Stack i TIMAC, przygotowane przez TI stos protokołu ZigBee/MAC 802.15.4. Jego użytkowanie, inaczej niż oprogramowania dla ZigBee, również nie wiąże się z kosztami.

Sieć SimpliciTI może pracować w topologii gwiazdy, gdzie centralnym węzłem komunikacyjnym jest punkt dostępowy lub w topologii peer-to-peer. W takim wypadku centralny element zarządzający nie jest niezbędny a poszczególni uczestnicy sieci wymieniają 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 przypadkach może też przechowywać dane dla innych węzłów i pośredniczyć w komunikacji. Z racji stałej aktywności wymaga właściwie zasilania sieciowego.

### Retransmitter – Range extender (RE)

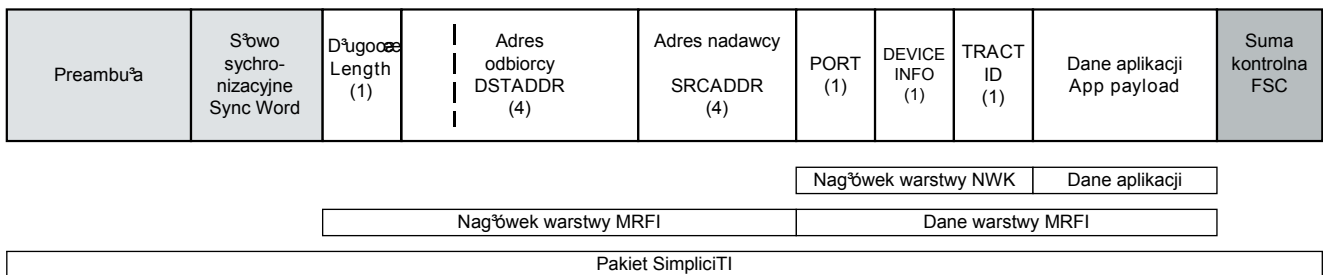
Z punktu widzenia protokołu retransmitter jest bardzo prostym urządzeniem. Jego zadaniem jest powtarzanie każdego usłyszanego prawidłowego pakietu. Liczebność tego typu urządzeń w sieci ograniczona jest do czterech. Podobnie jak punkt dostępowy wymaga zasilania sieciowego.

### Urządzenie końcowe – End Device (ED)

Urządzenie końcowe to 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).

Tab. 1. Porównanie protokołów SimpliciTI i ZigBee

Protokół	SimpliciTI	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



Rys. 3. Pakiet protokołu SimpliciTI

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.

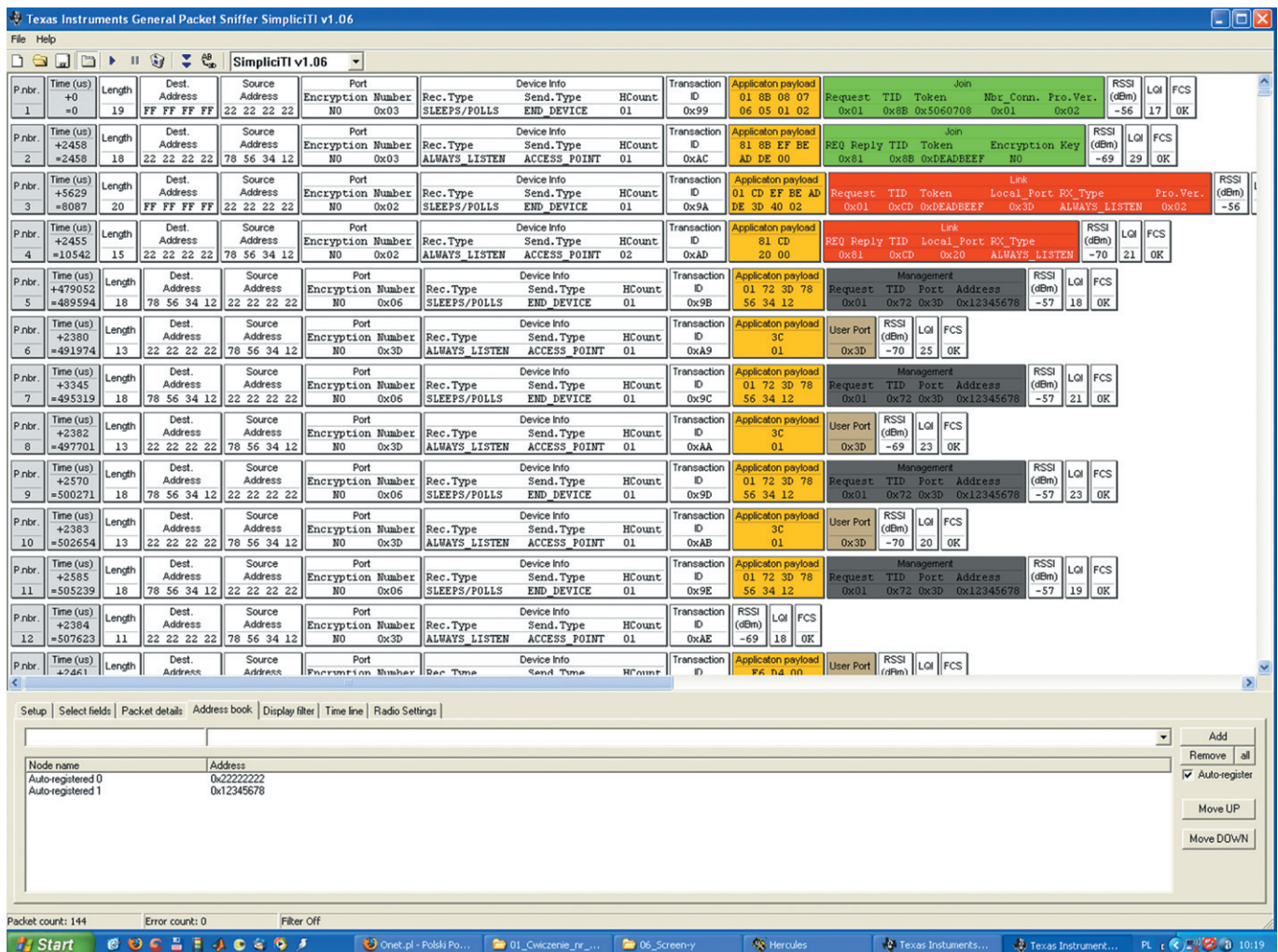
Sposób organizacji sieci, w porównaniu do ZigBee, wykazuje pewne uproszczenia. Zrezygnowano tutaj z funkcjonalności mesh. Jej pozytywną cechą jest możliwość przesyłania komunikatu dowolną drogą, wadą – znaczny przyrost objętości kodu i zużycie energii. W sieci SimpliciTI sposób przenoszenia komunikatu ze źródła do celu jest ściśle określony. Maksymalna ilość skoków, jaką może pokonać

komunikat wynosi 4 i odpowiada scenariuszowi, według którego 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 w prosty sposób sieć przed lawinowym narostem ilości powtarzanych komunikatów. Dodatkowo, retransmitery nie mogą komunikować się bezpośrednio. W wypadku połączenia peer-to-peer, gdy punkt dostępowy nie uczestniczy w transakcji, ilość skoków jest w naturalny sposób ograniczona do 2 (do retransmitera a potem do celu).

### Oprogramowanie

Oprogramowanie ma strukturę warstwową ograniczoną do niezbędnego minimum. W warstwie aplikacji znajduje się oprogramowanie użytkownika, wyniesiono również do niej aplikację sieci. Warstwa sieciowa Network zajmuje się uzupełnianiem pakietów o dane doty-

Z końcem marca opublikowano kolejną wersję protokołu – SimpliciTI 1.1.0. W nowej wersji zaimplementowano szyfrowanie. W konsekwencji niewielkiej zmiany 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.



Rys. 4. Widok okna programu Packet Sniffer z zapisem komunikacji pomiędzy śpiącym urządzeniem i punktem dostępowym buforującym dane

czące 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).

### Warstwa łącza – MRFI (*Minimal Radio Frequency Interface*) odpowiada za komunikację z układem radiowym i wykorzystanie jego sprzętowych możliwości (np. dekodowanie adresów)

Aplikacje sieciowe odpowiadają za określone zadania związane z nawiązaniem i podtrzymaniem komunikacji. Używają przypisanych im portów z zakresu 0x01- 0x1F. Aplikacje użytkownika posługują się portami 0x20 i wyżej.

#### Ping (0x01)

Polecenie używane do sprawdzenia obecności innych urządzeń.

#### Link(0x02)

Odpowiada za nawiązanie połączenia pomiędzy użytkownikami sieci. Podczas ustanawiania połączenia urządzenia wymieniają się swoimi adresami sieciowymi i przypisują je do identyfikatora (*LinkID*). Od tej chwili możliwa jest wymiana danych.

#### Join(0x03)

Odpowiada za proces przyłączenia do sieci zarządzanej przez punkt dostępowy. W efekcie procesu, inicjowanego przez urządzenie końcowe posługujące się predefiniowanym hasłem (*Join Token*), otrzymuje ono identyfikator połączenia. Urządzenia śpiące otrzymują dodatkowo bufor do przechowywania kierowanych do nich komunikatów w przestrzeni pamięci punktu dostępowego.

#### Security(0x04)

Odpowiada za przesyłanie informacji dotyczących szyfrowania. W obecnej wersji protokołu (1.06) szyfrowanie nie jest zaimplementowane.

#### Freq(0x05)

Odpowiada za zarządzanie częstotliwością pracy sieci. W protokole przewidziano zmianę częstotliwości pracy (mechanizm *frequency agility*). O wyborze jednego z predefiniowanych kanałów częstotliwościowych decyduje punkt dostępowy. Mechanizm działa tylko w sieciach zarządzanych przez AP. Urządzenia o jednokierunkowej transmisji, które nie mogą reagować na takie polecenia, z założenia nadają komunikaty na wszystkich dostępnych częstotliwościach.

#### Management(0x06)

Port Management użytkowany jest np. przez urządzenie śpiące do przesłania zapytania do punktu dostępowego o buforowane dane. Poza tym może służyć do komunikacji serwisowej.

Oprogramowanie SimpliTI udostępnia aplikacji użytkownika proste API składające się z 6 poleceń:

- inicjalizacja sieci – polecenie *SMPL\_Init*,

- proces linkowania – *SMPL\_Link* dla inicjatora i *SMPL\_LinkListen* dla drugiej strony,
- transmisja *SMPL\_Send* i *SMPL\_Receive*,
- konfiguracja – *SMPL\_Ioctl*.

Konfigurowanie urządzenia wykonuje się dwuetapowo. Na etapie przygotowania programu decyduje się o podstawowych parametrach: typie urządzenia, adresie(każde urządzenie musi mieć unikalny) i tokenach (wartość uniwersalna dla wszystkich potencjalnych użytkowników sieci) czy maksymalnej ilości skoków komunikatu. Definiuje się też parametry decydujące o zapotrzebowaniu dla pamięć RAM – liczbę możliwych połączeń a w przypadku AP również liczbę urządzeń, dla których przechowywane będą dane.

Za pomocą polecenia *SMPL\_Ioctl* konfiguruje się np. częstotliwość pracy, odczytuje ustawienia transceivera oraz dane o wersji protokołu. Wykorzystanie *Ioctl* pozwala również aplikacji rozpoznać i zmienić własny adres sieciowy.

### Jak wygląda struktura pakietu?

Pakiet rozpoczyna się preambułą i słowem synchronizacyjnym, generowanymi sprzętowo przez transceiver, po których występuje definicja długości pakietu. Następny w kolejności jest adres docelowy – *DSTADDR* i adres nadawcy komunikatu. Po nim następuje pole *PORT*, w którym umieszczona jest definicja portu aplikacji i informacja szyfrowaniu. Bajt *DEVICE INFO* zawiera informację o typie urządzenia (urządzenie śpiące lub nasłuchujące ciągle oraz *ED/AP/RE*) i licznik skoków komunikatu. Pole *TRACTID* zawiera liczbę porządkową komunikatu – pozwala na zidentyfikowanie duplikatów przesyłanych przez retransmitery. Tuż za polem danych aplikacji jest jeszcze pole zawierające sumę kontrolną.

Oprogramowanie SimpliTI jest dostępne pod adresem [www.ti.com/simpliiti](http://www.ti.com/simpliiti). Do stosu dołączone jest przykładowe oprogramowanie dla kilku konfiguracji.

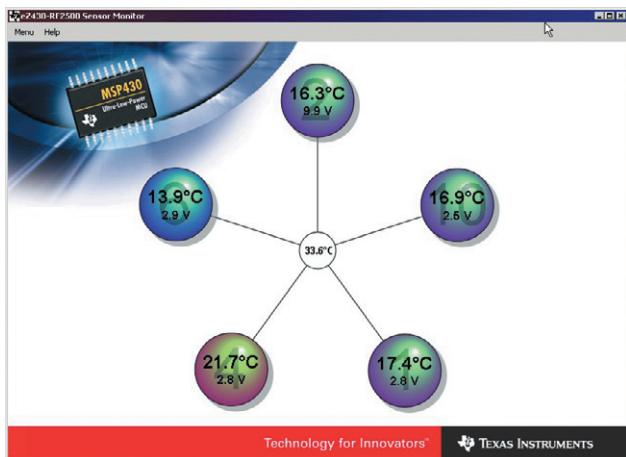
- Komunikacja bezpośrednia *peer-to-peer*: przykład komunikacji pomiędzy dwoma urządzeniami końcowymi
- Komunikacja *peer-to-peer* z udziałem punktu dostępowego: odmiana pierwszego przykładu, gdy jedno z urządzeń końcowych jest urządzeniem w trybie uśpienia. W tym wypadku punkt dostępowy buforuje przeznaczone dla niego dane.
- Sieć w topologii gwiazdy z buforującym punktem dostępowym: przykład, w którym urządzenia końcowe dołączają do sieci zarządzanej przez punkt dostępowy i komunikuje się z nim bezpośrednio.
- Kaskadowe rozgłaszanie komunikatu: przykład zastosowania protokołu w niesformalizowanej sieci czujników alarmowych, gdzie uczestnicy sieci rozsyłają komunikaty o zagrożeniu do wszystkich pozostałych używając adresu rozgłoszeniowego.

Stos napisano w języku C i przygotowano dla środowisk *IAR Embedded Workbench* oraz *CEssentials*. Działa z układami *System-On-Chip* (np. CC2430, CC2510) oraz transceiverami dedykowanymi do ZigBee (CC2520) i podstawowymi transceiverami na pasmo 2,4 GHz oraz 433/868 MHz (CC1101/CC2500).

### Uwagi praktyczne

Protokół SimpliTI jest alternatywą nie tylko dla standardowego protokołu ZigBee. Możemy go z powodzeniem stosować również w prostej komunikacji pomiędzy dwoma punktami, w miejsce samodzielnie opracowanego protokołu. Pełnię swoich możliwości oprogramowanie pokazuje jednak, gdy łatwość komunikacji trzeba połączyć z oszczędnością energii, jak w przypadku urządzeń o zasilaniu bateryjnym. W takim wypadku optymalne jest zastosowanie rozwiązania z centralnym punktem dostępowym buforującym dane i siecią śpiących urządzeń końcowych. Szybko uzyskuje się efekt, którego osiągnięcie przy samodzielnym projektowaniu protokołu wymaga wielu wysiłków.

Mariusz Kaczor, Sylwester Nowocień, Piotr Tadrzak  
support@contrans.pl



Rys. 5. Okno programu demonstrującego działanie sieci SimpliTI