

Systemy dla Internetu Rzeczy (17)

Jednoczesna komunikacja radiowa z użyciem dwóch protokołów i w dwóch pasmach

Bezpieczeństwo jest obecnie najważniejszą cechą systemów Internetu Rzeczy (IoT). A dużym problemem jest bezpieczna komunikacja użytkownika sieci z węzłem sensorów oraz bezpieczne przekazanie informacji pomiędzy węzłem centralnym i węzłami sensorów. Również problemem jest bezpieczne dołączenie węzła do sieci. Rozwiązaniem może być zastosowanie przez węzeł obsługi jednoczesnej transmisji z zastosowaniem różnych protokołów. Jeszcze lepiej, jeśli jednoczesna transmisja jest wykonywana w dwóch różnych pasmach radiowych. To już stało się możliwe z użyciem nowych układów scalonych SoC (System on Chip).

Wiele układów scalonych, może pracować w paśmie 2,4 GHz z obsługą protokołu Bluetooth 5, a także z komunikacją w standardzie IEEE 802.15.4. Umożliwia to dodatkową obsługę różnych protokołów: Zigbee, Thread, Wireless M-Bus, ANT oraz protokołów autorskich (proprietary). Obszerne omówienie takich układów zostało zamieszczone w artykule „Układy scalone z obsługą Bluetooth 5” (EP5/2018) [4].

Praca z wieloma protokołami

Pracę z wieloma protokołami na jednym układzie scalonym można realizować na cztery sposoby [11]:

1. Programowany (Programmable): Układ scalony może pracować z wieloma protokołami, ale fabrycznie jest ustawiany na jeden z nich.
2. Przelączany (Switched): Aplikacja przelacza pomiędzy protokołami używając bootloadera lub w kodzie aplikacji.
3. Dynamiczny (Dynamic): Przelaczanie pomiędzy protokołami z zastosowaniem podziału czasowego (time-slices).

Współbieżny (Concurrent): Oba protokoły pracują jednocześnie wykorzystując pojedynczy moduł radiowy.

Jednoczesna komunikacja radiowa z użyciem dwóch protokołów (i w dwóch różnych sieciach) jest możliwa w dwóch ostatnich przypadkach.

Nazwa jednoczesna – simultaneous jest często w dokumentacji zastępowana słowem concurrent, co może prowadzić do mylnej sugestii pracy współbieżnej.

PRACA Z PRZELĄCZANIEM PROTOKOŁÓW. Praca z przelączaniem oznacza, że układ jest w danym momencie dołączony tylko do jednej sieci z jednym protokołem. Przelączanie może być statyczne, gdy układ scalony obsługuje wiele protokołów, ale bootloader decyduje o tym, którego protokołu obsługa zostanie uruchomiona. Można też przelaczać protokoły podczas pracy aplikacji. W tym systemie działania trzeba wykonać odłączenie od aktualnej sieci z pierwszym protokołem. Następnie trzeba od początku wystartować obsługę nowego protokołu i wykonać dołączenie do nowej sieci. Czas przelączania jest znacząco długi [13].

PRACA Z PRZELĄCZANIEM DYNAMICZNYM. Przy pracy z dynamicznym przelączaniem protokołów układ scalony

obsługuje jednocześnie dwie sieci bezprzewodowe, każda z własnym protokołem. Jednak moduł radiowy układu scalonego jest tylko jeden. Dlatego raptownie zmienia się jego konfiguracja do pracy raz z jednym, raz z drugim protokołem. Bardzo istotny jest wybór momentów czasowych do realizacji tego podziału czasowego dostępu do modułu radiowego. Typowo jest to realizowane przez układ szeregowania (scheduler). Wymagane jest zarządzanie priorytetami dostępu i przydziałem czasu pracy modułu radiowego. Implementacja dynamicznym przelączaniem protokołów umożliwia jednoczesną pracę w obu sieciach komunikacyjnych.

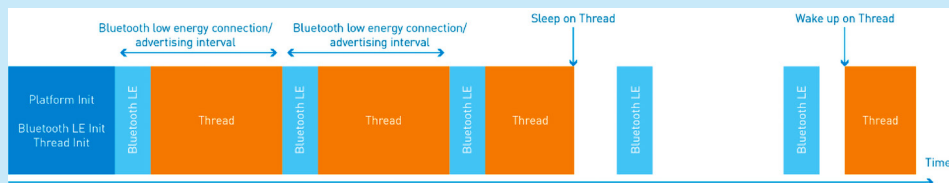
PRACA WSPÓLBIEŻNA. Układ scalony przy pracy współbieżnej protokołów obsługuje jednocześnie dwie sieci bezprzewodowe używając tego samego modułu radiowego. Wymaga to specjalnej konstrukcji sprzętowej.

JEDNOCZESNA KOMUNIKACJA RADIOWA Z UŻYCIEM WIELU MODUŁÓW

RADIOWYCH. Właściwie jest jeszcze jedno rozwiązanie współbieżnej komunikacji bezprzewodowej. Można zastosować dwa osobne moduły radiowe. Takie rozwiązanie może być przydatne (lub konieczne) w sytuacji, gdy obsługiwane mają być jednocześnie dwa różne pasma radiowe. Przykładem może być wprowadzanie do roku 2020 w Wielkiej Brytanii inteligentnych mierników domowych w 30 milionach gospodarstw. Wymagane jest zastosowanie dwóch stosów Zigbee. Jeden obsługujący Home Area Network i pracujący w paśmie 2,4 GHz. Drugi obsługujący PAN w paśmie 868 MHz.

Układy z obsługą wielu protokołów

Kilka układów scalonych, różnych producentów, umożliwia w paśmie 2,4 GHz jednoczesną transmisję w sieci Bluetooth 5 i sieci z innym protokołem. Są to np. układy serii nRF52x (Nordic Semiconductor) i układy rodziny SimpleLink CC13x2/CC26x2 (Texas Instruments). Inni producenci też zapowiadają możliwość równoczesnej transmisji z protokołem Bluetooth 5 oraz innym standardem: ST-M32WB55VG (ST Microelectronics), K32W042 (NXP) [4].



Rysunek 1. Jednoczesna praca z dynamicznym przelączaniem protokołu BLE5 i Thread (Nordic Semiconductor) [13]

Praca układów serii nRF52x z dynamicznym przełączaniem protokołów

Układy serii nRF52x (Nordic Semiconductor) umożliwiają w pasmie 2,4 GHz jednoczesną transmisję w sieciach Bluetooth 5 i Thread [2]. Zapewnia to mechanizm Dynamic Multiprotocol z obsługą stosu BLE5 oraz stosu OpenThread (IEEE 802.15.4) [13]. Mechanizm zapewnia, że praca modułu radiowego jest dzielona pomiędzy protokoły – z zapewnieniem, że transmisja obu jest obsługiwana. Współdzielenie czasowe pracy różnych protokołów jest realizowane automatycznie w tle i jest niewidoczne dla użytkownika.

Każdą przykładową aplikację z pakietu nRF5 SDK ze stosem BLE można uzupełnić o mechanizm Dynamic Multiprotocol z obsługą OpenThread. Nowy pakiet programowy S140 v6.0 SoftDevice umożliwia też jednoczesną pracę stosu BLE5 oraz stosu protokołu Zigbee 3.0. Co więcej z obsługą sieci mesh.

Mechanizm Dynamic Multiprotocol obsługuje podział czasowy (time-sliced) dostępu różnych protokołów do wspólnego modułu radiowego. Każdy protokół (np. BLE lub Thread) przed każdą operacją transmisji zgłasza żądanie przydziału czasu. To rozwiązanie pozwala na kontynuowanie pracy w sieci z jednym protokołem bez konieczności przerywania połączenia w sieci z drugim protokołem. Przykład jednoczesnej pracy z dynamicznym przełączaniem protokołu BLE5 i Thread pokazano na **rysunku 1** [13].

Pakiet programowy S140 v6.0 SoftDevice zawiera Radio Event Manager (REM) który zajmuje się arbitrażem dostępu do modułu radiowego. Drajwer stosu 802.15.4 (używany w protokole OpenThread) zawiera klienta arbitra, który używa Timeslot API do zarządzania zgłaszania żądań przydziału czasu przed każdą aktywnością komunikacyjną (**rysunek 2**).

Dynamiczne przełączanie różnych protokołów jest wykonywane automatycznie w tle aktualnie pracującej aplikacji i jest transparentne dla użytkownika. W przypadku aplikacji z użyciem dynamicznym przełączaniem protokołów, użytkownik pisze część dotyczącą transmisji z protokołem BLE tak jak zwykłą aplikację. Podobnie pisze osobno część dotyczącą transmisji z protokołem Thread. W tym przypadku transmisja z protokołem BLE ma zawsze wyższy priorytet niż transmisja z protokołem Thread. Protokół BLE stosuje podział czasowy (TDMA) dlatego wymaga aby poziom błędów transmisji pakietów wynosił 0%. W przypadku protokołu Thread sytuacja jest trochę łatwiejsza. Pakiety są przesyłane asynchronicznie. Z tego wynika, że podczas normalnej pracy występuje gubienie pakietów. Jeśli aktywność sieci Thread wystąpi podczas pracy modułu radiowego z protokołem BLE, to taka sytuacja jest traktowana jako zwykłe zgubienie pakietu. Transmisja jest wtedy ponawiana.

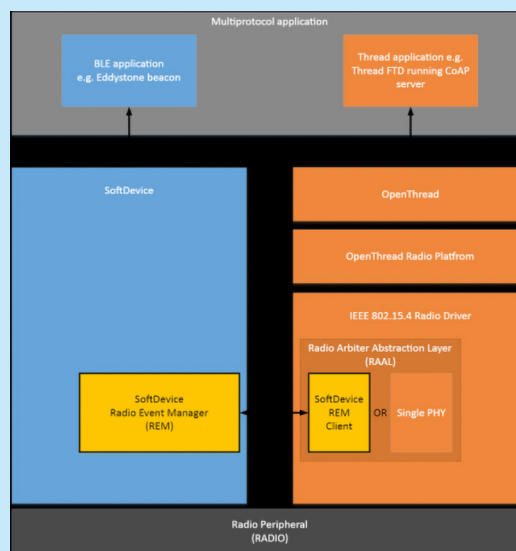
Układy z pracą dwuzakresową

Niektóre układy scalone pozwalają na pracę dwuzakresową [4]. Oprócz pracy w pasmie 2,4 GHz mogą one pracować w pasmie poniżej 1 GHz (Sub-1 GHz). Praktycznie w Europie oznacza to pasma ISM 868 MHz oraz 440 MHz. Typowo stosowana jest komunikacja w standardzie IEEE 802.15.4. Umożliwia to obsługę różnych protokołów: 6LoWPAN, KNX RF, Wi-SUN, SUN-FSK oraz rozwiązań autorskich producentów.

Dwuzakresowy procesor Mighty Gecko EFR32 (Silicon Labs) jest już w produkcji i dostępny za rozsądną cenę [3]. Drugim przykładem jest układ SimpleLink CC1352R1 (Texas Instruments) [1].

Praca dwuzakresowa układów serii EFR32 Mighty Gecko z dynamicznym przełączaniem protokołów

Układy scalone SoC serii EFR32 Mighty Gecko mają zastosowany procesor ARM Cortex-M4F (40 MHz), dużo pamięci (do 1024 KiB Flash, do 512 KiB RAM) oraz wiele modułów peryferyjnych [3]. Moduł radiowy obsługuje transmisję w pasmie 2,4 GHz z protokołem

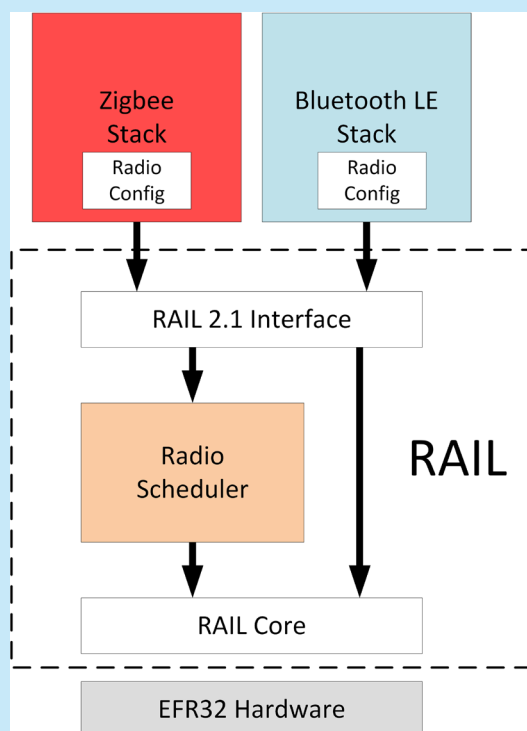


Rysunek 2. Schemat dynamicznego przełączania protokołu BLE5 i Thread (Nordic Semiconductor) [13]

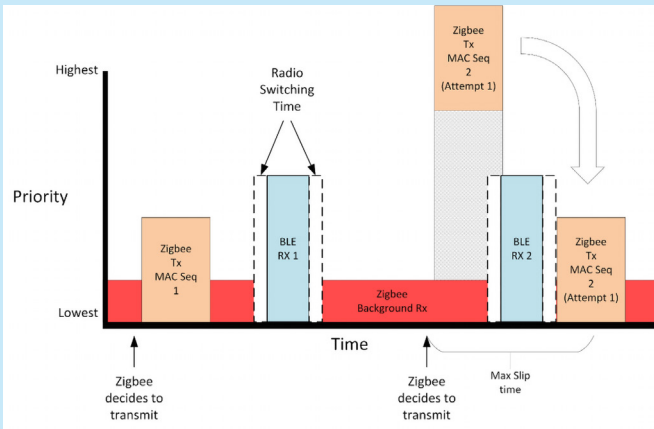
Zigbee 3.0 i Thread (IEEE 802.15.4), Bluetooth 5 oraz protokół autorski. Możliwe jest tworzenie sieci mesh [11].

Moduł radiowy obsługuje również transmisję w pasmie poniżej 1 GHz (915 MHz, 868 MHz, 433 MHz, 169 MHz). W tym pasmie obsługuje protokoły M-BUS, WAN oraz protokół autorski.

Producent udostępnia bardzo obszerne oprogramowanie oraz dokumentację z przykładowymi projektami. Oprogramowanie firmowe jest zorganizowane w postaci pakietu programowego Silicon Labs Flex SDK [11]. Zawiera ono Silicon Labs RAIL (Radio Abstraction Interface Layer) oraz Radio Scheduler. Oprogramowanie pracuje pod kontrolą systemu operacyjnego czasu rzeczywistego Micrium OS-5 [12]. RAIL 2.x udostępnia obsługę jednoczesnej pracy z wieloma protokołami pod nazwą Silicon Labs Dynamic Multiprotocol. Umożliwia to podział czasowy pracy modułu radiowego i szybkie przełączanie jego konfiguracji pracy. Stosy komunikacyjne są dostarczane jako pakiety EmberZNet SDK oraz Bluetooth SDK [12].



Rysunek 3. Ogólny schemat struktury modułów programowych (Silicon Labs) [11]



Rysunek 4. Przykład pracy szeregowania priorytetowego (Silicon Labs) [12]

Dla pracy z dynamiczną zmianą protokołów najważniejszy jest moduł szeregowania Radio Scheduler, będący częścią biblioteki RAIL. Odpowiada on za szeregowanie poleceń do modułu radiowego. Ogólny schemat struktury modułów programowych jej obsługi pokazano na **rysunku 3**. Radio Scheduler pracuje powyżej sprzętu modułu radiowego oraz poniżej API RAIL. Oprogramowanie RAIL Core konfiguruje moduł radiowy w odpowiedzi na polecenia od modułu Radio Scheduler.

Każdy stos pracuje jako osobne zadanie systemu operacyjnego Micrium OS-5, niezależnie od drugiego stosu. Większość operacji szeregowania poleceń jest wykonywana automatycznie przez RAIL. Wystarczy, że użytkownik używa stosu ze zwykłym API.

Są dwa warianty pracy z jednoczesną obsługą dwóch stosów:

- Zigbee (2,4 GHz) + Bluetooth 5 (2,4 GHz)
- RAIL (Sub-1 GHz) + Bluetooth (2,4 GHz)

Protokół komunikacji w standardzie BLE 5 ma bardzo restrykcyjne wymagania czasowe. Za to są one dokładnie ustalone i przewidywalne. W odróżnieniu, protokół Zigbee, pracujący ze standardem IEEE 802.15.4 jest bardziej elastyczny czasowo. Opóźnienie transmisji może wynosić wiele milisekund. To umożliwia wykonywanie transmisji pakietów standardu IEEE 802.15.4 w czasie wolnym, gdy nie ma transmisji w standardzie BLE 5. Komunikacja z zastosowaniem protokołu Zigbee ze stosem IEEE 802.15.4 wymaga stosunkowo niewiele czasu do realizacji transmisji oraz zapewnia mechanizm ponawiania prób transmisji. Dla tego możliwe jest wyłączenie transmitera na krótki okres czasu bez straty danych na poziomie aplikacji. Praca z protokołem BLE podczas rozgłaszania również wymaga wysyłania dosyć krótkich pakietów (zwykle do 30 B). Wymaga to około 1 ms czasu transmisji, z powtarzaniem nie częściej jak co 100 ms. Oznacza to wykorzystanie tylko 1% czasu pracy modułu radiowego. Pozostały czas można wykorzystać na obsługę transmisji innego protokołu [11].

Przykład jednoczesnej pracy stosu Bluetooth LE oraz stosu Zigbee z wykorzystaniem jednego modułu radiowego układu scalonego SoC serii EFR32 Mighty Gecko jest pokazany na **rysunku 4**.

Polecenia radiowe stosu Zigbee mają niższy priorytet oraz mogą być opóźnione o ustaloną wartość czasu. Aplikacja pracuje ze włączonym stosem Zigbee pracującym jako odbiornik. Odpowiada to pracy węzła centralnego sieci, co wymaga odebrania pakietu w dowolnym momencie. Stos BLE jest również włączony i wymaga odbioru danych dokładnie co 30 ms. Ze względu na przewidywalność tego wymagania, tworzone jest powtarzalne żądanie radiowe.

W przykładzie na rys. 4 najpierw stos Zigbee decyduje o konieczności wysłania pakietu (Tx1). Może to zrobić poprzez wysłanie polecenia na żądanie, co oznacza, że stos chce wysłać pakiet teraz, bez wcześniejszego informowania o tym modułu szeregowania. Jest to odwrotnie, niż pracuje stos BLE 5, gdzie moduł szeregowania ma informację znacząco wcześniej. Moduł szeregowania szacuje, że można wykonać

żądaną transmisję TX1 stosu Zigbee przed wykonaniem planowanej transmisji stosu BLE 5. Dlatego moduł szeregowania umożliwi przeprowadzenie komunikacji (czyli wysłanie pakietu danych, oczekiwanie na potwierdzenie i odebranie potwierdzenia). Estymowany czas komunikacji nie uwzględnia czasu na powtórzenie transmisji pakietu.

Następnie moduł szeregowania zezwala na zaplanowaną transmisję stosu BLE 5. Na rysunku uwzględniony jest także czas potrzebny na zmianę konfiguracji modułu radiowego, czyli przełączanie do obsługi stosu BLE 5 i z powrotem.

Po pewnym czasie stos Zigbee ponownie zgłasza żądanie wysłania pakietu (Tx2). Jednak tym razem moduł szeregowania szacuje, że nie można wykonać żądanej transmisji TX1 stosu Zigbee przed wykonaniem planowanej transmisji stosu BLE 5 o wyższym priorytecie. Jednak dla stosu Zigbee opóźnienie transmisji może wynosić wiele milisekund. Szacowany czas realizacji planowanej transmisji stosu BLE 5 jest krótszy niż dopuszczalny czas opóźnienia transmisji stosu Zigbee. Dlatego pierwsza próba wykonania transmisji TX2 stosu Zigbee zostanie uruchomiona bezpośrednio po komunikacji stosu BLE 5. Jeśli wszystko pójdzie dobrze to komunikacja w sieci obsługiwanej przez stos Zigbee będzie prowadzona bez błędów spowodowanych przez zastosowanie szeregowania podczas jednoczesnej pracy z dynamiczną zmianą protokołów.

Pokazany przykład jednoczesnej pracy z dynamiczną zmianą protokołów z zastosowaniem układu scalonego SoC serii EFR32 Mighty Gecko odnosi się również do jednoczesnej obsługi stosów RAIL (Sub-1GHz) oraz Bluetooth (2,4 GHz) [12].

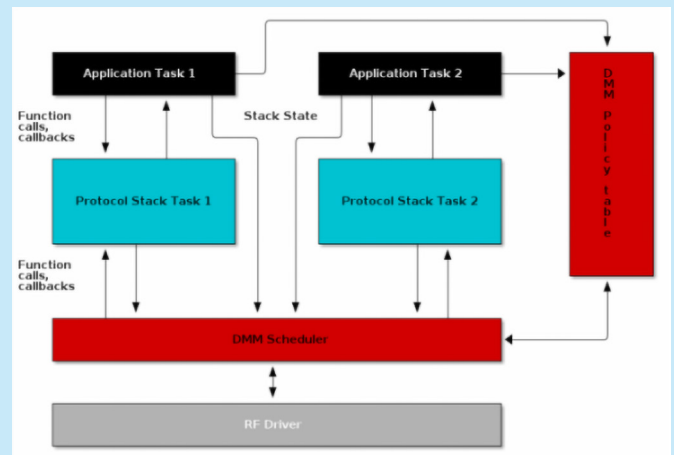
Układ SimpleLink CC1352R1F3 z jednoczesną transmisją z dwoma protokołami oraz w dwóch pasmach

Procesor SimpleLink CC1352R1 firmy Texas Instruments [1] umożliwia jednoczesną transmisję z dwoma protokołami: Bluetooth 5 i EasyLink oraz w dwóch pasmach: 2,4 GHz i w pasmie poniżej 1 GHz (Sub-1GHz).

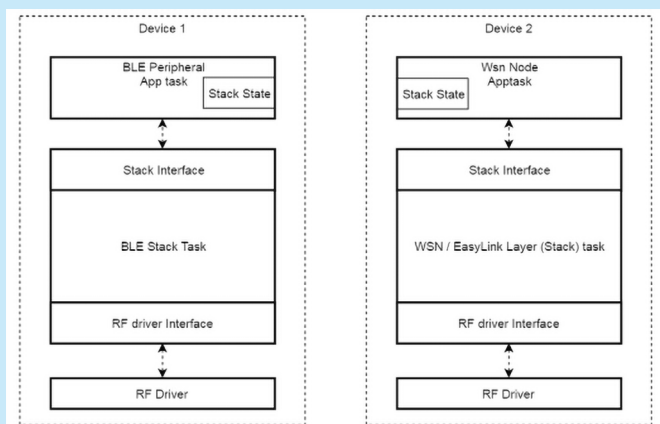
Umożliwia to technologia Dynamic Multi-protocol Manager (DMM) udostępniana jako część pakietu programowego CC13x2 SDK [6].

Układ CC1352R1F3 (typu SoC) z najnowszej rodziny SimpleLink CC13x2/CC26x2 firmy Texas Instruments ma nowy rdzeń główny ARM Cortex-M4F z bardzo rozbudowanymi mechanizmami sprzętowymi debugowania i śledzenia [1]. Drugi rdzeń Sensor Controller ma dodane dynamiczne sterowanie poborem mocy. Trzeci rdzeń ARM Cortex-M0 obsługuje komunikację radiową. Układ scalony CC1352R1 ma osobne wyprowadzenia dla obsługi transmisji w pasmie 2,4 GHz oraz w pasmie 868 MHz/915 MHz.

Układ scalony SimpleLink CC1352R1 (Texas Instruments) jest w fazie przedprodukcyjnej. Dostępne są próbki układów z tej serii



Rysunek 5. Opis użycia technologii DMM dla procesora CC1352R1F3 [9]



Rysunek 6. Schemat powiązania DMM ze stosami protokołów i aplikacją (Texas Instruments) [6]

o oznaczeniu XCC1352R1F3 (Lipiec 2018). Jest to zaawansowana wersja PG1.1. Jednak wersja z pełną kwalifikacją produkcyjną jest zapowiadana na początek roku 2019.

Dokumentacja

Dosyć zwyczajowo, trudno odszukać dokumentacji dotyczącej użycia technologii DMM z układami scalonymi rodziny CC13x2. Poszukiwania na portalu TI nie dają praktycznie żadnych rezultatów.

Dla rodziny CC13x2 opis jest ukryty w portalu TIREX [9]. Dla procesora CC1352R1F3 opis znajduje się w pakiecie SimpleLink CC13x2 Software Development Kit [6]. Trzeba nawigować do ścieżki Documents → Proprietary RF oraz Proprietary RF User's Guide. Następnie w dokumencie SimpleLink CC13x2 Proprietary RF User's Guide v:1.03.01.00 trzeba wybrać Dynamic Multi-protocol Manager (DMM) (rysunek 5).

W pakiecie CC13x2 SDK [6] udostępnione są też dwa przykładowe projekty w ścieżce Examples → Development Tools → CC1352R LaunchPad → dmm. Pierwszy projekt ma nazwę dmm_wsnnode_remote_display [10], a drugi dmm_wsnnode_ble_sp. Każdy projekt zawiera plik README.html w którym jest zamieszczony opis aplikacji projektu.

Pakiet SimpleLink Academy 2.10.02 for SimpleLink CC13x2 SDK 2.10 [6] zawiera opis bardzo ciekawego ćwiczenia laboratoryjnego Dynamic Multi-protocol Manager Fundamentals [14].

Wybrane pozostałe artykuły kursu

- S15 Systemy dla Internetu Rzeczy (15): Zestaw CC1352R1 LaunchPad, EP 5/2018
- S16 Systemy dla Internetu Rzeczy (16): Dynamiczne monitorowanie prądu zasilania układu SoC, EP 6/2018

Procesory

1. CC1352R (PREVIEW) SimpleLink Multi-Band CC1352R Wireless MCU, Product Page, <http://bit.ly/2vjkmqm>
2. nRF52840 High-end Bluetooth 5/Thread/802.15.4/ANT/2.4GHz multiprotocol SoC, Nordic Semiconductor, <http://bit.ly/2r29E8H>
3. EFR32 Mighty Gecko Mesh Networking Wireless SoCs for Zigbee and Thread, Silicon Labs, <http://bit.ly/2JYgB39>
4. Układy scalone z obsługą Bluetooth 5, Henryk A. Kowalski, „Elektronika Praktyczna”, 5/2018

Moduły sprzętowe

5. SimpleLink Multi-Band CC1352R Wireless MCU LaunchPad Development Kit, LAUNCHXL-CC1352R1, <http://bit.ly/210oP99>

Pakiety programowe

6. SimpleLink CC13x2 Software Development Kit, SIMPLELINK-CC13X2-SDK, 09-Jul-2018, Texas Instruments, <http://bit.ly/2M4Anub>

Jest też ciekawe wideo prezentacji technologii DMM „SimpleLink CC1352R Sub-1 GHz + Bluetooth low energy concurrency example” [17].

Zestaw startowy SimpleLink CC1352R1 LaunchPad

Do eksperymentów z technologią DMM potrzebny jest zestaw CC1352R1 LaunchPad [5]. Jego dokładny opis został zamieszczony w EP5/2018 [S15]. Zestaw startowy SimpleLink Multi-Band CC1352R Wireless MCU LaunchPad Development Kit (LAUNCHXL-CC1352R1) jest wyposażony w układ scalony CC1352R1F3 [1]. Zestaw jest przeznaczony do pracy w pasmach 868 MHz/915 MHz oraz 2,4 GHz. Cała elektronika zestawu CC1352R1 LaunchPad jest umieszczona na jednej wielowarstwowej płytce drukowanej.

Płytkę jest podzielona na dwie części: u góry płytki jest debugger sprzętowy standardu XDS110 oraz układ monitora zasilania, a na dole układ scalony CC1352R1F3.

Układ scalony CC1352R ma osobne wyprowadzenia dla obsługi transmisji w pasmie 2,4 GHz oraz w pasmie 868 MHz ISM (Europa) lub 915 MHz ISM (USA). W każdym torze radiowym są elementy dopasowujące doprowadzone do przełącznika sygnału radiowego (U28). Na płytce są zamieszczone dwa złącza radiowe (typu uSMA (JSC), żeńskie 50 Ω). Jedno gniazdko (P11) umożliwia dołączenie do toru radiowego anteny zewnętrznej dla pasma 2,4 GHz po zdjęciu rezystora R16 i założeniu R19. Drugie gniazdko (P12) umożliwia dołączenie do toru radiowego anteny zewnętrznej dla pasma 868/915 MHz po zdjęciu kondensatora C36 i założeniu C37. W obu przypadkach gdy używane jest gniazdko antenowe to sygnał nie jest wtedy doprowadzany do przełącznika. Takie zorganizowanie wyprowadzenia sygnału radiowego na dwie anteny sugeruje możliwość jednoczesnej pracy w obu zakresach [6].

Do wyjścia przełącznika jest, poprzez dopasowanie, dołączona antena (A1) wykonana na płytce drukowanej (umieszczona na samym dole płytki drukowanej). Antena ta pozwala na pracę zarówno w pasmie 2,4 GHz jak również w pasmie 868/915 MHz. Na płytce jest zamieszczone trzecie złącze radiowe (typu uSMA, żeńskie 50 Ω). Umożliwia ono, po zdjęciu kondensatora C482 i założeniu C483, dołączenie anteny zewnętrznej w miejsce anteny drukowanej.

Dynamic Multi-protocol Manager (DMM)

Technologia Dynamic Multi-protocol Manager (DMM) została opracowana przez firmę Texas Instruments i udostępniona jako część pakietu

7. iTunes, TI SimpleLink Starter By Texas Instruments (Version 5.2, Jan 24, 2018), <https://apple.co/2JNC8Q7>
8. Google Play, Simplelink SensorTag, Texas Instruments Inc. (Version: 5.3.3, 4 maja 2018), <http://bit.ly/2tk14Ub>

Opisy

9. TI Resource Explorer (TIREX), Texas Instruments, <http://bit.ly/2HHDqgo>
10. DMM Wsn Node + BLE Remote Display (README.html), Texas Instruments, <http://bit.ly/2K49XLQ>
11. UG103.16: Multiprotocol Fundamentals, Rev. 0.1, Silicon Labs, <http://bit.ly/2JMs49H>
12. UG305: Dynamic Multiprotocol User's Guide, Rev. 0.3, Silicon Labs, <http://bit.ly/2t7qnJV>
13. Multiprotocol support, nRF5 SDK for Thread and Zigbee v1.0.0, Nordic Semiconductor, <http://bit.ly/2JYdaN9>

SimpleLink Academy

14. Dynamic Multi-protocol Manager Fundamentals, Texas Instruments, <http://bit.ly/2MF9I8m>

programowego CC13x2 SDK [6]. DMM jest arbitrem poleceń RF pomiędzy stosami protokołów a drajwerami RF. Przechwytuje on wywołania drajwerów i potencjalnie może zmieniać kolejność wykonania poleceń. Schemat powiązania DMM ze stosami protokołów i aplikacją pokazano na **rysunku 5**. Na czerwono są zaznaczone bloki DMM. Pozostałe elementy są pokazane w typowej konfiguracji jak z jednym protokołem.

DMM używa dwóch komponentów:

- DMM Scheduler.
- DMM Policy Manager (DMM policies and states table).

W aplikacji pracuje jeden moduł szeregowania DMM Scheduler. Przechwytuje on polecenia wydawane przez stos i PHY. Tablica zawiera bieżący stan każdego stosu oraz parametry szeregowania dla specyficznych kombinacji stanów. DMM Scheduler wykonuje szeregowanie poleceń bazując na bieżącym stanie tablicy. Uwzględniane są ograniczenia i wymagania stosów oraz aplikacji.

Przykład pracy z dwoma osobnymi stosami jest pokazany na **rysunku 6**. Każdy stos pracuje na osobnym układzie scalonym. Na jednym pracuje aplikacja typu BLE Peripheral (może też być BLE 5 ProjectZero) z obsługą pełnego stosu BLE 5. Na drugim pracuje aplikacja WSN Node (może być też WSN Concentrator) z obsługą jest autorskiego stosu Easy-Link firmy Texas Instruments. Jest to prosty stos o małych rozmiarach, zbliżony funkcjonalnie do stosu TI 15.4. Dla wygody nazywany jest dalej jako stos WSN. Aplikacja WSN Node wraz z aplikacją WSN Concentrator tworzą sieć WSN (Wireless Sensor Network).

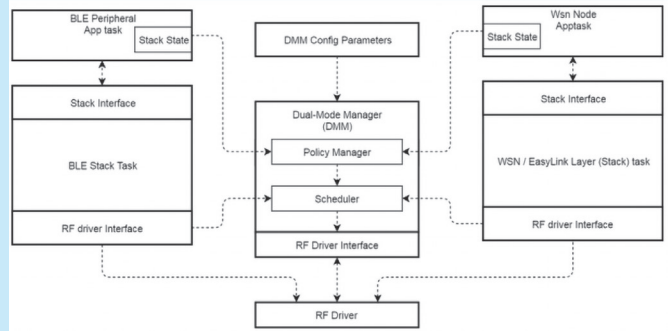
Przykład pracy DMM tych dwóch stosów na jednym układzie scalonym pokazano na **rysunku 7**. Jest to organizacja przykładowej aplikacji dmm_wsnnode_remote_display (patrz dalej Zadanie 2) [14], zastosowanej w tym artykule. Dla wygody nazywana jest dalej jako aplikacja DMM. DMM pośredniczy pomiędzy stosami i drajwerami RF. Szeregowanie poleceń jest dla użytkownika niewidoczne. Wymagane jest tylko, aby DMM Scheduler był informowany przez użytkownika o zmianach stanu stosów. Jeśli jeden stos pracuje w paśmie 2,4 GHz a drugi w paśmie poniżej 1 GHz to typowo wymagane jest przełączanie anteny. DMM bezpośrednio nie steruje przełączaniem anteny. Typowo zajmuje się tym sama aplikacja.

Praca z DMM wymaga użycia środowiska Code Composer Studio (CCS) w wersji co najmniej 8.0 oraz procesora CC1352R1F3 [6]. Podczas eksperymentów została zastosowana aplikacja dmm_wsnnode_remote_display z pakietu programowego SimpleLink CC13x2 SDK pracująca na procesorze zestawu startowego CC1352R1 LaunchPad [10]. Środowisko w wersji CCS v8.0 pracuje z CC13x2 SDK v2.10 [6].

Opis oprogramowania narzędziowego dla układów CC26xx i CC13xx platformy SimpleLink został zamieszczony w EP11/2017 [S12]. Opis oprogramowania dedykowanego dla procesora CC1352R1F3 został zamieszczony w EP5/2018 [S15].

Wnioski

Dynamiczne przełączanie protokołów jest nową własnością udostępnianą przez producentów układów scalonych SoC. Daje to ogromne możliwości w wielu dziedzinach zastosowań. Zaletą tego rozwiązania jest możliwość zastosowania protokołu Bluetooth LE wersji 5. Ma to duże znaczenie przy obserwowanym „zatykaniu się” pasma 2,4 GHz przez punkty dostępowe WiFi. Dlatego rosnące znaczenie ma transmisja w paśmie poniżej 1 GHz (Sub-1 GHz).



Rysunek 7. Praca dwóch stosów na dwóch układach scalonych CC1352R1F3 [10]

Jednoczesna transmisja w obu pasmach wydaje się bardzo przyszłościowym rozwiązaniem.

Każdy z producentów zaprezentowanych trzech rodzajów układów scalonych wspiera użycie systemu operacyjnego czasu rzeczywistego FreeRTOS. Może to pokazywać kierunek dalszego rozwoju tej dynamicznej dziedziny, która właśnie powstaje. Widać też silne powiązania z zastosowaniem aplikacji mobilnych. To już nie jest moda. To konieczność.

Henryk A. Kowalski
kowalski@ii.pw.edu.pl

REKLAMA

Nie przegap! interesujących materiałów w siostrzanym czasopiśmie



www.elportal.pl

**A może masz pomysł na ciekawy artykuł lub projekt?
Skonstruowałeś urządzenie,
które jest godne zaprezentowania szerszej publiczności?**

Możesz napisać artykuł edukacyjny?

Czcesz podzielić się doświadczeniem?

**W takim razie zapraszamy do współpracy
na łamach Elektroniki dla Wszystkich.**

Kontakt: edw@elportal.pl

EdW możesz zamówić na

www.ulubionykiosk.pl

Do kupienia również w Empikach

i wszystkich większych kioskach z prasą.

W sierpniowym wydaniu

Elektroniki dla Wszystkich między innymi:

Minizegar Nixie

Prosty i wykonany z łatwo dostępnych elementów czterocyfrowy zegar z atrakcyjnym wyświetlaczem. Umożliwia odczyt czasu, a dzięki zastosowaniu układu zegara czasu rzeczywistego (RTC) nie traci wskazań po zaniku napięcia zasilania. **Czteropasmowy odbiornik STAS – odbiornik nasłuchowy CW/SSB** Choć sprzęt fabryczny jest powszechnie dostępny, nadal nie słabnie zainteresowanie opisami prostych odbiorników radiowych oraz radiokomunikacyjnych, w szczególności umożliwiających odbiór sygnałów jednowęstwegowych i telegraficznych.

Przetwornice przepustowe

Aby omówić bardzo interesujący sposób resetowania rdzenia zwany „active clamp” musimy cofnąć się do nieobciążonych przetwornic stykowych, a w szczególności do występujących w nich przebiegów.

Arduino: Pomiar wilgotności, łącze I2C, a także Snap4Arduino

W szóstym odcinku kursu Arduino poznajemy popularne łącze I2C, rozwiązujemy związane z nim problemy i przekonujemy się, jakim ułatwieniem jest korzystanie z przygotowanych przez innych, gotowych bibliotek, jakich tysiące mamy do dyspozycji.

Zasilacz warsztatowy o zmniejszonej mocy strat Klasyczny zasilacz analogowy, uzupełniony przez prosty preregulator napięcia, który zapewni znacznie niższe straty cieplne, niż popularne rozwiązania.

Ponadto w numerze:

- Symetryczny zasilacz do wzmacniaczy audio
- Tranzystory: JFET-y dużej mocy?
- Chłodzenie i radiatory. Praca impulsowa
- Maksymalne prądy w przewodach i ścieżkach
- Silniki prądu stałego – pełny model elektryczny
- Warsztatowe patenty. Imadeiko, czyli... krokodyl
- Szkoła Konstruktorów – rozwiązanie zmniejszające skutki braku energii w sieci 230V
- Szkoła Konstruktorów – Zapropnuj urządzenie, układ lub artykuł związany z Arduino