

Systemy dla Internetu Rzeczy (11)

Bezprzewodowa sieć czujników z transmisją dwupasmową

Zestawy CC1350 Launch Pad oraz CC1350 SensorTag są dostarczane z wpisanym do pamięci Flash procesora programem, który umożliwia transmisję z zastosowaniem protokołu BLE 4.2. W pakiecie programowym SimpleLink CC13x0 SDK 1.50 [12] jest dostarczany komplet warsztatów o nazwie SimpleLink Academy for SimpleLink CC13x0 SDK [13]. Do pierwszych prób z transmisją dwupasmową najbardziej przydatny jest warsztat Creating a CC13xx based ultra low power Wireless Sensor Network. Oprogramowanie umożliwia utworzenie bezprzewodowej sieci czujników (Wireless Sensor Network – WSN) w paśmie ISM 868 MHz (sub-1GHz). Umożliwia też pracę zestawu z komunikacją dwupasmową (Dual Band) ISM 2,4 GHz oraz ISM 868 MHz. Sieć zawiera jeden węzeł centralny – Concentrator (WSN Concentrator) oraz wielu węzłów czujników – Node (WSN Node). Węzły Node przesyłają wyniki pomiarów do węzła Concentrator poprzez łącze w paśmie ISM 868 MHz. Dodatkowo węzły Node mogą przeprogramować swoje radio i wysyłać pakiety rozgłaszania BLE.

Zorganizowanie przykładowej bezprzewodowej sieci czujników z zastosowaniem transmisji dwupasmowej pokazano na rysunku w nagłówku artykułu [8]. Sieć składa się z trzech węzłów

czujników typu *Node* oraz jednego węzła centralnego typu *Concentrator* pracujących w paśmie Sub-1GHz. Typowo węzeł centralny jest dołączony do Internetu. Na rysunku zostało pokazane

Pozostałe artykuły kursu

- S1. Systemy dla Internetu Rzeczy (1). Zestaw CC2650 SensorTag, „Elektronika Praktyczna”, 12/2016
- S2. Systemy dla Internetu Rzeczy (2). Użytkowanie zestawu CC2650 SensorTag, „Elektronika Praktyczna”, 1/2017
- S3. Systemy dla Internetu Rzeczy (3). Moduły rozszerzeń DevPack dla zestawu SensorTag, „Elektronika Praktyczna”, 2/2017
- S4. Systemy dla Internetu Rzeczy (4). Zestaw CC1310 Launchpad, „Elektronika Praktyczna”, 3/2017
- S5. Systemy dla Internetu Rzeczy (5). System operacyjny czasu rzeczywistego TI-RTOS – pierwszy program, „Elektronika Praktyczna”, 4/2017
- S6. Systemy dla Internetu Rzeczy (6). System operacyjny czasu rzeczywistego TI-RTOS – zadania i przerwania, „Elektronika Praktyczna”, 5/2017
- S7. Systemy dla Internetu Rzeczy (7). Bluetooth Low Energy, „Elektronika Praktyczna”, 6/2017
- S8. Systemy dla Internetu Rzeczy (8). Zestaw startowy CC2650 Launchpad, „Elektronika Praktyczna”, 7/2017
- S9. Systemy dla Internetu Rzeczy (9). Zestaw startowy CC1350 Launchpad, „Elektronika Praktyczna”, 9/2017
- S10. Systemy dla Internetu Rzeczy (10). Zestaw CC1350 SensorTag, „Elektronika Praktyczna”, 10/2017

połączenie jednego węzła z komunikatorem typu smartfon poprzez łącze BLE. Ale każdy węzeł *Node* może wykonywać transmisję BLE.

Sprzęt

Do zestawienia połączenia radiowego potrzebne są co najmniej dwa zestawy sprzętowe. Jeden zestaw pracuje jako *Concentrator* a drugi jako *Node*. Jako węzły sieci WSN mogą pracować zestawy CC1350 Launchpad. Do zestawu może być dołączony wyświetlacz LCD. Próby pokazały, że zestaw CC1350 SensorTag też może pracować jako węzeł WSN *Node*. Z oprogramowaniem dedykowanym dla zestawu CC1350 Launchpad. Dostępne są również projekty programowe sieci WSN przeznaczone dla zestawu CC1310 Launchpad. Obsługują one wtedy tylko główny kanał transmisji dla ISM 868 MHz, bez obsługi BLE.

Zestaw startowy CC1350 Launchpad zawiera tylko minimum elementów dodatkowych: wyprowadzenia układu scalonego CC1350, przyciski, diody LED, zasilanie, dwie anteny PCB oraz emulator sprzętowy [5]. Zwory na płytce umożliwiają łatwą modyfikację jej zastosowania. Jest pełny dostęp do wszystkich sygnałów. Złącze rozszerzeń umożliwia łatwe dołączanie kolejnych modułów sprzętowych.

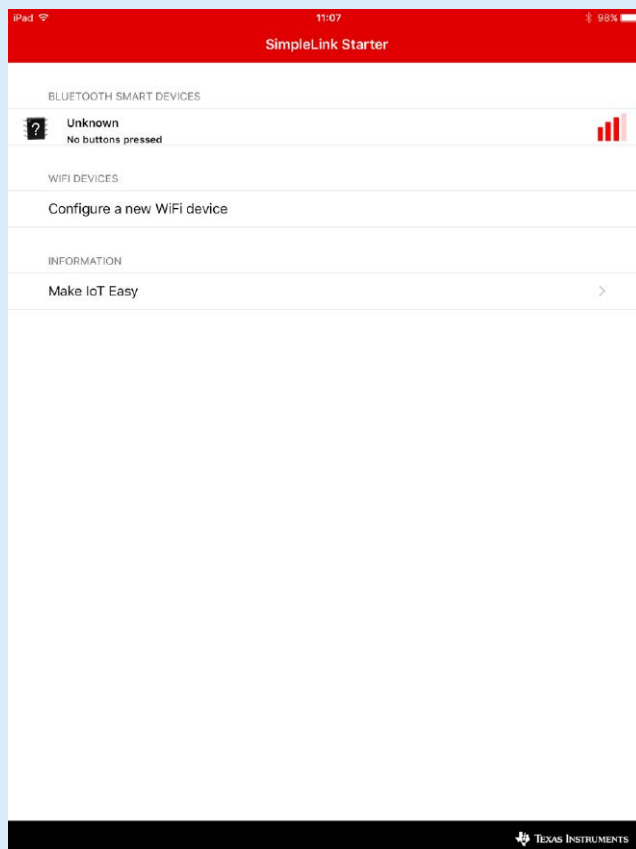
Zestaw CC1350 Launchpad jest dostępny w trzech wersjach wykonania:

- LAUNCHXL-CC1350US: Przystosowany do pracy w paśmie 915 MHz ISM (USA) [S9].
- LAUNCHXL-CC1350EU: Przystosowany do pracy w paśmie 868 MHz ISM (Europa) [S9].
- LAUNCHXL-CC1350-4: Przystosowany do pracy w paśmie 433 MHz ISM (USA/Europa/Chiny) [S9]. W portalu TI spotykana jest też nazwa LAUNCHXL-CC1350CN.

Zestaw CC1350 SensorTag (SimpleLink CC1350 SensorTag Bluetooth and Sub-1GHz Long Range Wireless Development Kit CC1350STK) – obsługuje komunikację bezprzewodową Bluetooth Low Energy w paśmie ISM 2,4 GHz oraz komunikację dużego zasięgu w pasmach ISM poniżej 1 GHz (868/915 MHz) [4].

Zestaw CC2650 SensorTag jest dostępny w dwóch wersjach:

- CC1350STKUS: zoptymalizowany do pracy w paśmie 915 MHz [S10].
- CC1350STKEU: zoptymalizowany do pracy w paśmie 868 MHz [S10].



Rysunek 1. Główne okno w aplikacji SimpleLink Starter

Moduł rozszerzeń Debug DevPack zawiera układ emulatora sprzętowego typu XDS110 [6]. Moduł Debug jest nakładany na zestaw CC1350 SensorTag. Wtedy zasilanie kompletu CC1350 SensorTag + Debug DevPack jest dostarczane ze złącza USB modułu Debug [S3].

Moduł rozszerzeń 430BOOST-SHARP96 LCD Boosterpacks zawiera monochromatyczny wyświetlacz LCD (Sharp LS013B4DN04) z panelem TFT [9]. Ma on przekątną 1,35', rozdzielczość 96×96 pikseli (9216) i pobiera tylko 2 μ A prądu. Panel jest dołączony do złącza rozszerzeń standardu LAUNCH firmy Texas Instruments. Moduł jest nakładany na płytke zestawu Launchpad.

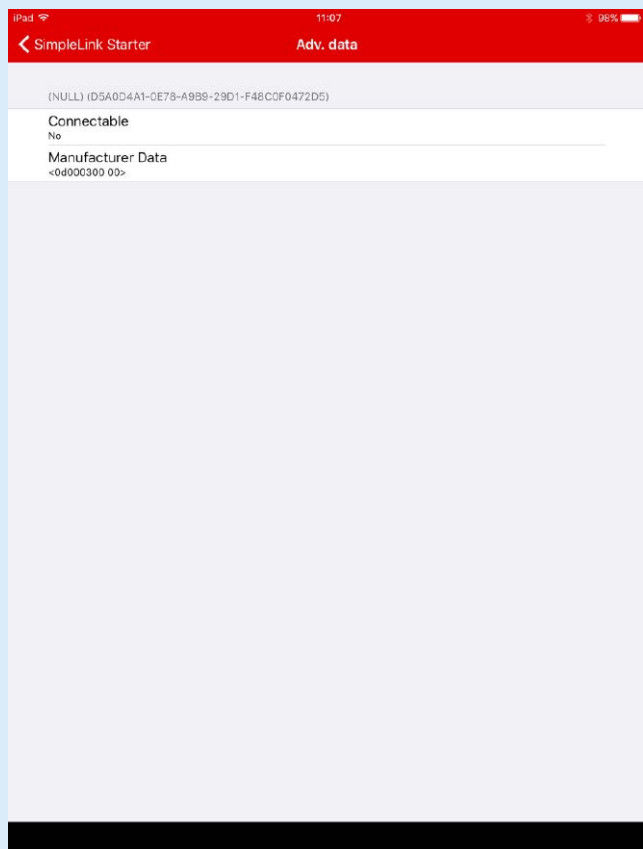
Zestaw startowy CC1310 Launchpad zawiera tylko minimum elementów dodatkowych: wyprowadzenia układu scalonego CC1310, przyciski, diody LED, zasilanie, antenę PCB oraz emulator sprzętowy [S4]. Zwory na płytce umożliwiają łatwą modyfikację jej zastosowania. Jest pełny dostęp do wszystkich sygnałów. Złącze rozszerzeń umożliwia łatwe dołączanie kolejnych modułów sprzętowych. Zestaw jest w pełni zgodny sprzętowo i programowo z zestawem startowym CC1350 Launchpad.

Dołączanie zestawu CC1350 LaunchPad/SensorTag do urządzenia mobilnego

Zestaw CC1350 Launchpad/SensorTag, pracujący jako węzeł *Node* sieci WSN, może zostać dołączony poprzez łącze radiowe standardu Bluetooth LE do urządzenia mobilnego (smartfon/iPad) lub innego z obsługą tego standardu. Opis i odnośniki na stronie *Meet the CC1350 SensorTag* [8].

Postępowanie przy uruchamianiu komunikacji z urządzeniem mobilnym:

1. **Pobierz** darmową aplikację *SimpleLink Starter* dla swojego urządzenia mobilnego (smartfon/iPad), obsługującego transmisję bezprzewodową Bluetooth LE ver.4.0 (lub nowszą).
 - a. Apple App Store dla urządzeń z systemem operacyjnym iOS 8.0 i nowszym (iPhone i iPad) [14].



Rysunek 2. Okno danych w aplikacji *SimpleLink Starter*

- a. Z Google Play dla urządzeń z systemem operacyjnym Android 5.0 i nowszym [7].
2. **Włącz zasilanie** zestawu CC1350 Launchpad/SensorTag. Czerwona dioda LED zaczyna błyskać. Sygnalizuje to, że zestaw CC1350 Launchpad/SensorTag jest w stanie rozgłaszania.
3. **Uruchom** aplikację *SimpleLink Starter* na swoim urządzeniu mobilnym. Może to wymagać włączenia na urządzeniu radia Bluetooth. Aplikacja najlepiej działa na iPadzie. W przypadku Androida można spotkać się z kłopotami i ograniczeniami funkcjonalności.
4. **Wyszukaj** na liście „Bluetooth Smart Devices”, w głównym oknie aplikacji, swój zestaw CC1350 Launchpad/SensorTag. Niestety, każdy zestaw jest pokazywany jako „Unknown” (rysunek 1).
5. **Kliknij** na linię nazwy i wybierz *Advertisement Data*. Zostanie wyświetlone okno prezentujące bieżące wartości odczytu z zestawu (rysunek 2).

Oprogramowanie narzędziowe

Do pracy z projektami przeznaczonymi dla sieci WSN potrzebne jest kilka komponentów:

- A. Najpierw należy zainstalować środowisko Code Composer Studio. Darmową aktualną wersję pełną CCS można pobrać ze strony Texas Instruments [10]. Obecnie (10.2017) jest dostępna wersja CCS7.3.0.00019. Jest ona zrealizowana przy zastosowaniu nowej wersji środowiska Eclipse. Podczas instalowania należy na liście produktów do zainstalowania zaznaczyć (wybrać) opcje *SimpleLink CC13xx and CC26cc Wireless MCUs* oraz *TI XDS Debug Probe Support*. W trakcie instalowania oprócz foldera `c:/ti/ccsv7` są tworzone foldery `/xdctools_3_32_02_25_core` oraz `/xdctools_3_50_03_33_core` z zainstalowanymi środowiskami debugowymi.

- B. Po zainstalowaniu środowiska CCS należy zainstalować pakiet *SimpleLink CC13x0 SDK* [12].

Oprogramowanie źródłowe

Dla procesorów komunikacyjnych rodziny CC13x0 przeznaczony jest pakiet programowy SimpleLink CC13x0 SDK 1.50.00.08 (28-Sep-2017) [12]. Jest mały problem z nazewnictwem. Procesor CC1350 jest nazywany SimpleLink Ultra-Low Power Dual Band Wireless Microcontroller, czyli dwupasmowy. Za to w pakiecie SDK (szczególnie w poprzedniej wersji 1.40) transmisja w dwóch pasmach ISM nazywana jest Dual Mode, czyli mowa o dwóch standardach transmisji. Faktycznie stosowane są dwa standardy komunikacji: BLE oraz IEEE 802.15.4g. Podobnie jak w przypadku procesora CC2650, SimpleLink multi-standard 2.4 GHz ultra-low power wireless MCU. Jednak istotą jest obsługa dwóch różnych pasm częstotliwości, czyli raczej Dual Band. W obecnej wersji SDK mowa jest tylko o sieci WSN bez wskazania, że oprócz pasma ISM sub-1GHz obsługiwane jest też pasmo ISM 2.4 GHz.

W pakiecie SDK są dostępne projekty WSN przeznaczone dla zestawu CC1310 Launchpad. Obsługują one wtedy tylko główny kanał transmisji dla ISM 868 MHz, bez obsługi BLE.

W pakiecie SDK dostępne są również projekty WSN dodatkowo obsługujące metodę Over the Air Update (OAD), gdzie nowe oprogramowanie można przesłać do zestawu drogą bezprzewodową. Jednak to zagadnienie jest na tyle obszerne, że zostanie omówione w osobnym artykule.

W pakiecie programowym SimpleLink CC13x0 SDK 1.50 jest dostarczany komplet warsztatów o nazwie *SimpleLink Academy for SimpleLink CC13x0 SDK* [12]. Do pierwszych prób z transmisją dwupasmową najbardziej przydatny jest warsztat *Getting started with CC1350 Launchpad in Dual Mode* [13] (obecnie X.2017 przejściowo z poprzedniej wersji pakietu SDK v1.40).

Projekty pakietu SimpleLink CC13x0 SDK 1.50 były uruchamiane w środowisku IDE z następującymi wersjami komponentów:

- Code Composer Studio: 7.3.0.00019,
- TI-RTOS for CC13XX: 3_30_00_13,
- C Compiler for ARM: 16.9.4.LTS,
- XDCTools: 3.50.03.33.

Pakiet SimpleLink CC13x0 SDK można pobrać i zainstalować na trzy sposoby:

- Poprzez witrynę *TI Resource Explorer* [11].
- Poprzez okno *Resource Explorer* (zalecane) w programie Code Composer Studio [10].
- Poprzez pobranie pliku instalacyjnego ze strony TI [12].

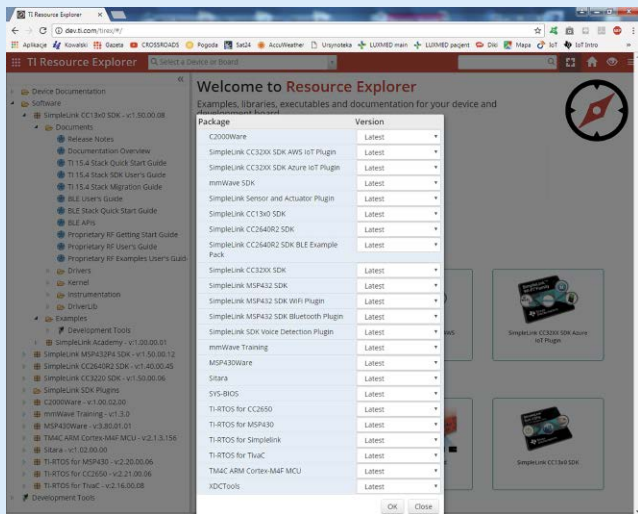
Podczas instalowania pakietu SDK sprawdzana jest obecność w komputerze komponentów w odpowiednich wersjach i w przypadku ich braku wykonywane jest ich doinstalowanie.

Witryna TI Resource Explorer

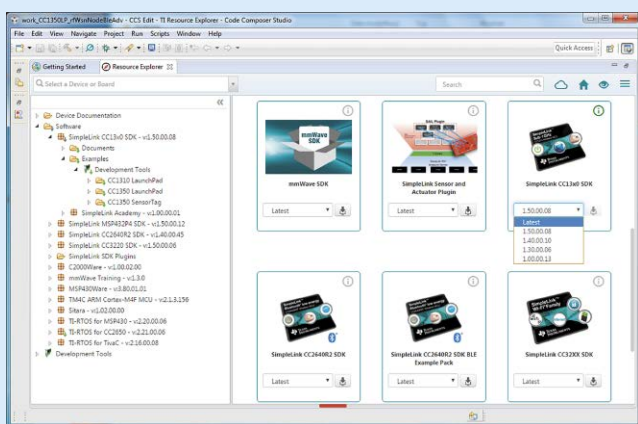
W witrynie TI Resource Explorer są obecnie zgromadzone (prawy wszystkie) informacje i źródła dotyczące procesorów komunikacyjnych rodziny CC13x0 [11]. Daje to pełny i uporządkowany dostęp do zasobów w aktualnych wersjach (domyślnie). Jest także umożliwiony dostęp do ich poprzednich wersji. W menu strony należy kliknąć na ikonkę *Package Picker*. Zostaje rozwinięta lista pakietów programowych (rysunek 3). Typowo jest wybrana ostatnia wersja (Latest). Po wybraniu archiwalnej wersji i kliknięciu na przycisk *OK* pokazywana jest wybrana wersja. Jej ikonka jest specjalnie zaznaczona.

Okno Resource Explorer programu CCSv7.3

Okno *Resource Explorer* w programie CCSv7.3 realizuje dostęp do witryny *TI Resource Explorer*. Najlepiej go otwierać poprzez panel *Getting Started*, używając przycisku *Browse Examples* [10].



Rysunek 3. Witryna TI Resource Explorer (TIREX)



Rysunek 4. Okno Resource Explorer programu CCS 7.3.0

Uruchomienie tego okna z menu *View* → *Resource Explorer* często powoduje kłopoty. Typowo zgłasza on problem połączenia z portalem TI po pierwszej próbie. Czasami pomagają zamknięcie tego okna i ponowne jego uruchomienie.

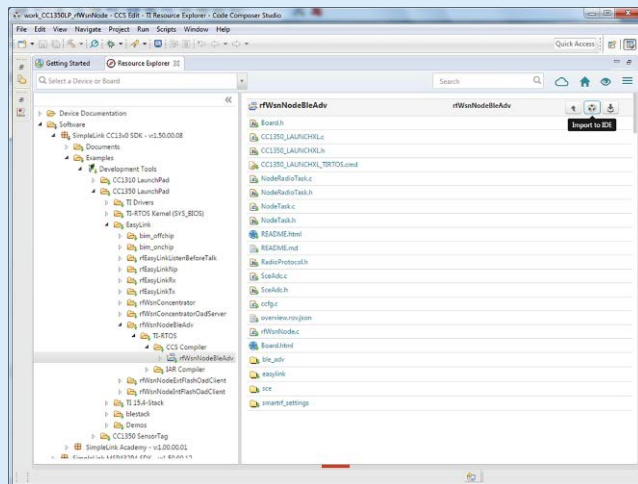
Przy otwieraniu okna *Resource Explorer* pobierana jego aktualna zawartość z witryny TIREX [11].

Ikonki folderów w lewym panelu okna mogą być pokazywane na trzy sposoby (**rysunek 4**):

- Zawartość nie jest zainstalowana lokalnie, czyli dostępna tylko w sieci.
- Część zawartości jest zainstalowana i dostępna lokalnie.
- Zawartość jest zainstalowana lokalnie i w pełni dostępna lokalnie.

Typowo, w lewym panelu okna pokazywana jest ostatnia wersja (Latest) pakietu programowego. Po kliknięciu na pasku menu okna na ikonkę Home dostępne są pola poszczególnych pakietów programowych. Po rozwinięciu listy można wybrać potrzebną wersję archiwalną pakietu (rys. 4). Ta wersja jest wtedy pokazywana w lewym panelu. Jej ikonka jest specjalnie zaznaczona. Po wybraniu projektu w lewym panelu, w prawym panelu pojawiają się pasek z ikonkami (**rysunek 5**):

- *Install resources* – Pobierz i zainstaluj cały pakiet programowy wybrany w lewym panelu okna.
- *Import to IDE* – Powoduje utworzenie w aktualnym folderze roboczym CCS odpowiednich folderów z kopią lokalną plików. Jednocześnie sprawdzane są wersje zależnych narzędziowych pakietów programowych zainstalowanych lokalnie. W razie potrzeby nowsze wersje są pobierane i instalowane.



Rysunek 5. Pobieranie projektu rfWsnNodeBleAdv z witryny internetowej TI Resource Explorer [11]

- *Open web resources* – Otwiera aktualną zawartość prętego panelu w osobnym oknie domyślnej przeglądarki. Typowo jest używany inny adres internetowy.

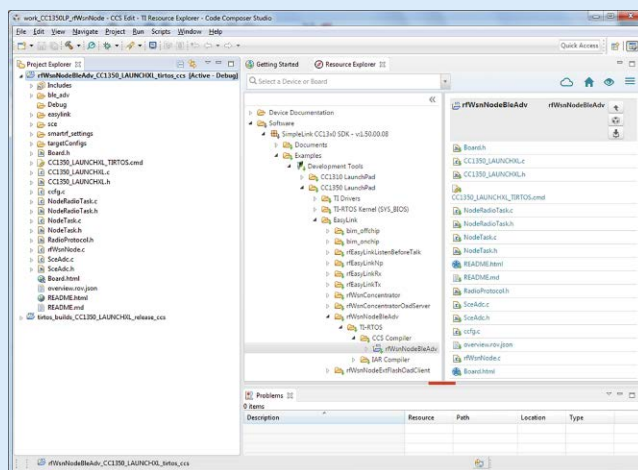
Plik instalacyjny pakietu SDK

Ze strony SimpleLink Sub-1 GHz CC13x0 Software Development Kit [12] można pobrać plik instalacyjny `simplelink_cc13x0_sdk_1_50_00_08.exe` pakietu SDK. Postępowanie podczas lokalnego instalowania pakietu jest takie samo w przypadku dwóch poprzednich sposobów instalowania zdalnego. Co więcej, w oknie *Resource Explorer* w programie CCSv7.3, jego zainstalowanie jest rozpoznawane.

Oprogramowanie węzła Node

Pakiet programowy SDK oferuje wersje projektów dla trzech typów zestawów sprzętowych z wyborem dwóch typów kompilatorów. W wypadku stosowania jako węzeł *Node* zestawu sprzętowego CC1350 Launchpad w oknie *Resource Explorer* rozwinąć ścieżkę (rys. 5): *Software* → *SimpleLink CC13x0 SDK – v:1.50.00.08* → *Examples* → *Development Tools* → *CC1350 Launchpad* → *EasyLink* → *rfWsnNodeBleAdv* → *TI-RTOS* → *CCS Compiler* → *rfWsnNodeBleAdv*. W prawym panelu okna *Resource Explorer* kliknij na ikonkę *CCS* – operacja *Import to IDE*.

Po załadowaniu projektu i wyświetleniu okna *Project Explorer* w perspektywie *CCS Edit* należy jeszcze poczekać na zakończenie wszystkich prac prowadzonych w tle przez środowisko Eclipse (**rysunek 6**). Często jest wtedy wyświetlany znaczek wykrzyknika



Rysunek 6. Załadowany projekt rfWsnNodeBleAdv

na ikonke projektu w tym oknie. Może trwać to dosyć długo, aż skończą się pojawiać na pasku stanu kolejne informacje.

Po wydaniu polecenia kompilacji, z menu *Project* → *Build All*, postęp jest bardzo powolny w okolicach 1–2%. Jest to spowodowane kompilacją projektu zależnego – z systemem operacyjnym TI-RTOS. Potem już jest szybko 25%, 45% i koniec kompilacji. Jeśli został poprawnie wygenerowany plik kodu wynikowego `rfWsnNodeBleAdv_CC1350_LAUNCHXL_tirtos_ccs.out`, to można przystąpić do programowania procesora. Z menu *Run* → *Debug* zostanie uruchomiony proces pełnego budowania projektu, nastąpi dołączenie debugera do procesora, załadowanie kodu (zaprogramowanie wewnętrznej pamięci Flash) oraz uruchomienie działania aż do pierwszej instrukcji funkcji `main()`.

Przy próbie debugowania może zostać wyświetlona informacja o konieczności aktualizacji oprogramowania dołączonego emulatora XDS110 (moduł rozszerzeń Debug DevPack). Należy to koniecznie wykonać. Dokładny opis postępowania jest zamieszczony w odcinku 7 kursu [S7]. Po zakończeniu aktualizacji można bez problemu ponowić próbę debugowania. Praktyka pokazała, że kodem projektu `rfWsnNodeBleAdv` przeznaczonego dla CC1350 Launchpad można programować procesor CC1350 zestawu sprzętowego CC1350 SensorTag. Dla zestawu sprzętowego CC1310 Launchpad jest przeznaczony w pakiecie SDK projekt `rfWsnNode`. Węzeł *Node* obsługuje wtedy tylko główny kanał transmisji dla ISM 868 MHz, bez obsługi BLE.

Projekt programu dla węzła *Node* dla zestawu CC1350 Launchpad składa się z projektu głównego `rfWsnNodeBleAdv_CC1350_LAUNCHXL_tirtos_ccs` (program aplikacyjny) oraz projektu zależnego `tirtos_builds_CC1350_LAUNCHXL_release_ccs` (używane w projekcie aplikacyjnym składniki systemu operacyjnego TI-RTOS). Pliki źródłowe projektu znajdują się lokalnie na dysku komputera w ścieżce `C:\ti\simplelink_cc13x0_sdk_1_50_00_08\examples\rtos\CC1350_LAUNCHXL\easylink\rfWsnNodeBleAdv`. W pliku `README.html` jest zamieszczony jedyny opis organizacji i działania tego oprogramowania. Kod źródłowy projektu zawiera się głównie w trzech plikach: `rfWsnNode.c`, `NodeTask.c` oraz `NodeRadioTask.c`. Kod programu jest umieszczony w wewnętrznej pamięci Flash procesora z intensywnym wykorzystaniem oprogramowania umieszczonego w pamięci ROM tego procesora.

Procesor CC1350 ma trzy rdzenie: Cortex M3 (CM3), Sensor Controller Engine (SCE) oraz Cortex M0 (CM0) [1]. Rdzeń CM3 wykonuje kod aplikacyjny oraz obsługę stosu komunikacyjnego [2]. Rdzeń SCE wykonuje zadanie czasu rzeczywistego, które obsługuje pracę przetwornika ADC. Rdzeń CM0 obsługuje pracę części radiowej procesora komunikacyjnego [3].

Oprogramowanie węzła *Node* jest zorganizowane w postaci trzech zadań czasu rzeczywistego [11]. Dwa zadania systemu operacyjnego czasu rzeczywistego TI-RTOS [11] *NodeTask* (aplikacyjne) oraz *NodeRadioTask* (stosu komunikacyjnego) pracują na rdzeniu CM3. Trzecie zadanie *SceAdc* pracuje na rdzeniu SCE. Jest ono realizowane sprzętowo jako odpowiedź na przerwanie. Rdzeń CM0 ma własne oprogramowanie firmowe.

Procesor CC1350 po włączeniu zasilania rozpoczyna pracę od wykonywania funkcji `main()` (plik `rfWsnNode.c`). Najpierw są inicjalizowane zasoby sprzętowe (**listing 1**) zastosowanego

```
Listing 1. Funkcja main() (plik rfWsnNode.c)
int main(void)
{
    /* Call driver init functions. */
    Board_initGeneral();
    Display_init();
    /* Initialize sensor Node tasks */
    NodeRadioTask_init();
    NodeTask_init();
    /* Start BIOS */
    BIOS_start();
    return (0);
}
```

zestawu sprzętowego. Następnie inicjalizowane są moduły peryferyjne procesora UART i SPI. Jeśli do procesora CC1350 dołączony jest moduł LCD, to też jest on inicjalizowany. Następnie inicjalizowane są dwa zadania systemu TI-RTOS: obsługujące transmisję radiową (*NodeRadioTask*) oraz obsługujące komunikację sieciową (*NodeTask*). Na koniec wywoływany jest system operacyjny TI-RTOS i wykonywanie kodu nigdy nie powraca do funkcji `main()`.

Zadanie SceAdc

Zadanie *SceAdc* pracuje na rdzeniu SCE. Jest ono realizowane sprzętowo. W trakcie inicjalizacji wykonywanej przez zadanie *NodeTask*, pracujące na rdzeniu CM3, ładowany jest kod drajwera do rdzenia SCE. Najpierw ustawiany jest timer modułu zegara czasu rzeczywistego (RTC) procesora na zgłaszanie przerwania, co ustalony okres (1 sekunda). Następnie aktywowana jest obsługa przerwania, ustalone parametry pracy i proces *SceAdc* jest uruchamiany.

Co jedną sekundę zegar RTC zgłasza przerwanie. Budzi ono rdzeń SCE i startuje procedura obsługi przerwania. Wykonywane jest pojedyncze przetwarzanie ADC i odczytywana jest nowa wartość wyniku. Jeśli nowa odczytana wartość wykazuje zmianę do poprzedniej wartości o ustalony w parametrach poziom, to rdzeń SCE budzi rdzeń CM3. W przeciwnym wypadku rdzeń CM3 nie jest budzony, aż upłynie zadany parametrami czas.

Zadanie NodeRadioTask

Zadanie *NodeRadioTask* obsługuje stos komunikacyjny. Funkcja *NodeRadioTaskFunction* (plik `NodeRadioTask.c`) definiuje działanie zadania *NodeRadioTask*. Najpierw inicjalizowany jest sprzętowy moduł radiowy procesora z wykorzystaniem biblioteki `SimpleLink-EasyLink`. Potem zostaje uruchomiony sprzętowy generator liczb losowych i zostaje wygenerowany adres sieciowy (ID). Adres węzła *Node* musi być inny od adresu modułu koncentratora (ustalonego jako `0x00`). Dodatkowo sprawdzana (filtrowana) jest tabela adresów.

Po inicjalizacji struktur danych zadania uruchamiana jest główna pętla komunikacji bezprzewodowej. Jest to pętla nieskończona zawieszana oczekiwaniem na zdarzenie. Po przyjsciu zdarzenia wymagającego wykonania komunikacji zadanie *NodeRadioTask* jest aktywowane przez system operacyjny. Nowa wartość odczytana przez proces *SceAdc* jest wysyłana w pakiecie do węzła *Concentrator* z zastosowaniem ustalonego formatu, domyślnie IEEE 802.15.4g 50 kbit. Po wysłaniu pakietu odbierane jest potwierdzenie (ACK) od węzła *Concentrator* i wysyłane jest własne potwierdzenie. Przy braku odebranego potwierdzenia ACK ponawiana jest trzy razy transmisja pakietu.

Zadanie NodeTask

Zadanie *NodeTask* realizuje obsługę aplikacyjną działania węzła *Node*. Funkcja *NodeTaskFunction* (plik `NodeTask.c`) definiuje działanie zadania *NodeTask*. Najpierw inicjalizowane są parametry pracy zadania *NodeTask*. Potem inicjalizowany jest wyświetlacz LCD i wyświetlane są początkowe napisy. Inicjalizowane jest też łącze UART. Następnie inicjalizowane jest zadanie *SceAdc*. Dalej jest startowana główna pętla aplikacji (**listing 2**). Na początku pętli jest wywołanie funkcji `Event_pend`. Zawiesza ona wykonanie zadania do czasu wykrycia przez system operacyjny zdarzenia. Co więcej, rdzeń CM3 też wchodzi w uśpienie. Jest on budzony przez zadanie *SceAdc* tylko wtedy, gdy trzeba wykonać transmisję. Wtedy zadanie *NodeRadioTask* jest aktywowane przez system operacyjny.

Jeśli zostało wygenerowane zdarzenie `NODE_EVENT_NEW_ADC_VALUE`, to odczytywana jest wartość ostatniego przetwarzania A/C, wartość ostatniego odczytu jest wyświetlana (razem z adresem sieciowym) na wyświetlaczu LCD. Dla zdarzenia `NODE_EVENT_UPDATE_LCD` wykonywane jest tylko wyświetlanie. Wyświetlany tekst jest także wysyłany do łącza UART. Wartość

Listing 2. Pętla główna procesu NodeTask (plik NodeTask.c)

```

while (1)
{
    /* Wait for event */
    uint32_t events = Event_pend(NodeEventHandle, 0, NODE_EVENT_ALL, BIOS_WAIT_FOREVER);
    /* If new ADC value, send this data */
    if (events & NODE_EVENT_NEW_ADC_VALUE) {
        /* Toggle activity LED */
        PIN_setOutputValue(ledPinHandle, NODE_ACTIVITY_LED, !PIN_getOutputValue(NODE_ACTIVITY_LED));
        /* Send ADC value to concentrator */
        NodeRadioTask_sendAdcData(latestAdcValue);
        /* Update display */
        updateLcd();
    }
    /* If new ADC value, send this data */
    if (events & NODE_EVENT_UPDATE_LCD)
    {
        /* update display */
        updateLcd();
    }
}

```

Listing 3. Obsługa wyświetlacza LCD (plik NodeTask.c)

```

/* print to LCD */
Display_clear(hDisplayLcd);
Display_printf(hDisplayLcd, 0, 0, „NodeID: 0x%02x”, NodeAddress);
Display_printf(hDisplayLcd, 1, 0, „ADC: %04d”, latestAdcValue);
/* print to LCD */
Display_printf(hDisplayLcd, 2, 0, „Adv Mode:”);
Display_printf(hDisplayLcd, 3, 0, „%s”, advMode);
Display_printf(hDisplayLcd, 4, 0, „Adv successful | failed”);
Display_printf(hDisplayLcd, 5, 0, „%04d | %04d”,
                bleAdvStats.successCnt + bleAdvStats.failCnt);

```

ostatniego odczytu jest wysyłana w pakiecie danych do koncentratora z zastosowaniem podstawowego kanału ISM 868 MHz. Pakiet danych zawiera ostatni odczyt, poziom baterii oraz stan przycisku BTN-1.

Jeśli wymagana jest też transmisja w kanale BLE, to wysyłany jest pakiet rozgłaszania w odpowiednim formacie.

Obsługa wyświetlacza LCD i łącza UART dla węzła Node

Informacja jest generowana na dwa sposoby:

- Dla wyświetlacza LCD.
- Dla łącza UART.

W trakcie inicjalizacji oprogramowania wykonywana jest próba inicjalizacji łącza UART (preferowane) oraz wyświetlacza LCD (poprzez łącze SPI). Dla zestawu CC1350/1310 Launchpad inicjalizowane są oba łącza. Dla zestawu CC1350 SensorTag oba łącza nie działają.

Po inicjalizacji do wyświetlacza LCD jest wysyłany tekst

- do wyświetlacza LCD – „Waiting for ADC...”.
- do łącza UART – „Waiting for SCE ADC reading...”.

Są cztery napisy identyfikujące typy ramek rozgłaszania: „BLE MS”, „Eddystone URL”, „Eddystone UID”, „None”. Sposób generowania tekstu dla wyświetlacza LCD pokazano na **listingu 3** (funkcja updateLcd). Wyświetlacz LCD (Sharp LS013B4DN04) ma rozdzielczość 96×96 pikseli. Linia tekstu jest na tyle krótka, że nie mieści się napis „failed”. Sposób generowania tekstu dla łącza UART pokazano na **listingu 4** (funkcja updateLcd).

W terminalu, na początku wyświetlania tekstu każdej transmisji, obecny tekst jest przesuwany do góry. Na pustym ekranie terminalu wyświetlany jest nowy tekst. W ostatniej linii tekstu podawana jest statystyka: liczba pomyślnie wysłanych ramek rozgłaszania oraz liczba wszystkich ramek.

Dla terminalu UART liczby wydają się poprawne. Tylko skąd błędy przy nadawaniu ramek rozgłaszania? Dla wyświetlacza LCD pierwsza pokazywana liczba wydaje się liczbą wszystkich ramek rozgłaszania. A druga pokazywana najczęściej liczba to „-001”. A czasami to samo, co liczba pierwsza. Przyczyną takiego błędnego działania jest błędne wywołanie funkcji Display_printf w ostatniej linii obsługi wyświetlacza LCD (list. 3). Zadeklarowane są do wyświetlania dwa pola znakowe, a dostarczana jest tylko jedna wartość. Skorygowanie wywołania pokazano na **listingu 5**.

Zmiana oprogramowania spowodowała poprawne wyświetlanie danych na wyświetlaczu LCD. Zniknęły też „błędne” ramki rozgłaszania. Nie do końca. Pojawiają się one też w sytuacji, gdy węzeł Node nie jest połączony z węzłem Concentrator.

Oprogramowanie węzła Concentrator

W przypadku stosowania jako węzeł Concentrator zestawu sprzętowego CC1350 Launchpad w oknie Resource Explorer rozwiń ścieżkę

Listing 4. Obsługa łącza UART (plik NodeTask.c)

```

/* print to UART clear screen, put cursor to beginning of terminal and print the header */
Display_printf(hDisplaySerial, 0, 0, „033[2J 033[0;0HNode ID: 0x%02x”, NodeAddress);
Display_printf(hDisplaySerial, 0, 0, „Node ADC Reading: %04d”, latestAdcValue);
/* print to UART */
Display_printf(hDisplaySerial, 0, 0, „Advertiser Mode: %s”, advMode);
Display_printf(hDisplaySerial, 0, 0, „Advertisement success: %d out of %d”,
                bleAdvStats.successCnt,
                bleAdvStats.successCnt + bleAdvStats.failCnt);

```

Listing 5. Poprawa ostatniej linii obsługi wyświetlacza LCD (plik NodeTask.c)

```

Display_printf(hDisplayLcd, 5, 0, „%04d | %04d”,
                bleAdvStats.successCnt,
                bleAdvStats.successCnt + bleAdvStats.failCnt);

```

REKLAMA

Prenumerujesz
„Elektronikę Praktyczną”
i „Elektronikę dla Wszystkich”?
Masz prawo do
bezpłatnej prenumeraty
miesięcznika „Elektronik”
w promocji 1+1=3
www.avt.pl/prenumerata



(rysunek 7) Software → SimpleLink CC13x0 SDK – v:1.50.00.08 → Examples → Development Tools → CC1350 Launchpad → EasyLink → rfWsnConcentrator → TI-RTOS → CCS Compiler → rfWsnConcentrator. W prawym panelu kliknij na ikonkę CCS – operacja Import to IDE. Po załadowaniu projektu i odczekaniu na zakończenie wszystkich prac prowadzonych w tle przez środowisko Eclipse należy wybrać polecenie kompilacji z menu Project → Build All (rys. 7). Jeśli został poprawnie wygenerowany plik kodu wynikowego rfWsnConcentrator_CC1350_LAUNCHXL_tirtos_ccs.out, to można przystąpić do programowania procesora z menu Run → Debug. Dla zestawu sprzętowego CC1310 Launchpad jest przeznaczony w pakiecie SDK projekt rfWsnConcentrator. Węzeł Concentrator obsługuje wtedy tylko główny kanał transmisji dla ISM 868 MHz, bez obsługi BLE.

Projekt programu dla węzła Concentrator ma budowę bardzo podobną jak projektu programu dla węzła Node.

Oprogramowanie składa się z projektu głównego rfWsnConcentrator_CC1350_LAUNCHXL_tirtos_ccs (program aplikacyjny) oraz projektu zależnego tirtos_builds_CC1350_LAUNCHXL_release_ccs (używane w projekcie aplikacyjnym składniki systemu operacyjnego TI-RTOS). Kod programu jest umieszczony w wewnętrznej pamięci Flash procesora z intensywnym wykorzystaniem oprogramowania umieszczonego w pamięci ROM tego procesora.

Oprogramowanie węzła Concentrator jest zorganizowane w postaci dwóch zadań czasu rzeczywistego. Oba zadania system operacyjnego czasu rzeczywistego TI-RTOS [11] ConcentratorTask (aplikacyjne) oraz ConcentratorRadioTask (stosu komunikacyjnego) pracują na rdzeniu CM3. Rdzeń CM0 ma własne oprogramowanie firmowe.

Procesor CC1350 po włączeniu zasilania rozpoczyna pracę od wykonywania funkcji main() (plik ConcentratorTask.c). Najpierw są inicjalizowane zasoby sprzętowe zastosowanego zestawu sprzętowego. Następnie inicjalizowane są moduły peryferyjne procesora UART i SPI. Jeśli do procesora CC1350 dołączony jest moduł LCD, to też jest on inicjalizowany.

Następnie inicjalizowane są dwa zadania systemu TI-RTOS: obsługujące transmisję radiową (ConcentratorRadioTask) oraz obsługujące komunikację sieciową (ConcentratorTask).

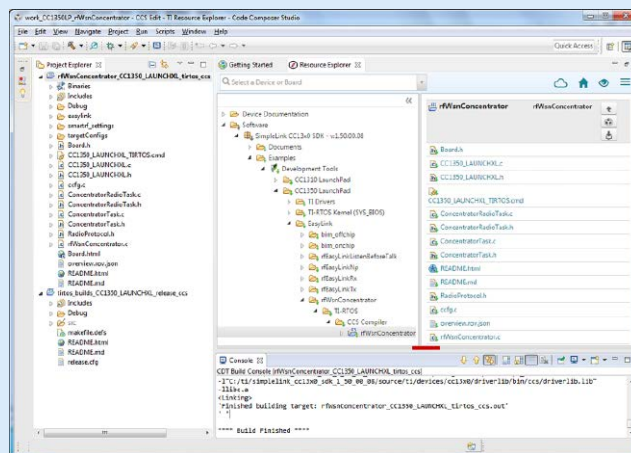
Na koniec wywoływany jest system operacyjny TI-RTOS i wykonywanie kodu nigdy nie powraca do funkcji main().

Zadanie ConcentratorRadioTask

Zadanie ConcentratorRadioTask obsługuje stos komunikacyjny. Funkcja concentratorRadioTaskFunction (plik ConcentratorRadioTask.c) definiuje działanie zadania ConcentratorRadioTask. Najpierw inicjalizowany jest sprzętowy moduł radiowy procesora z wykorzystaniem biblioteki SimpleLink-EasyLink. Potem zostaje ustawiony domyślny adres sieciowy (ID) jako 0x00.

Po inicjalizacji struktur danych zadania uruchamiana jest główna pętla komunikacji bezprzewodowej. Jest to pętla nieskończona zawieszana oczekiwaniem na zdarzenie. Po przyjsciu zdarzenia sygnalizującego odbiór przez radio pakietu ustawionego formatu – domyślnie IEEE 802.15.4g 50kbit, zadanie ConcentratorRadioTask jest aktywowane przez system operacyjny. Wysyłany jest pakiet potwierdzenia (ACK) do węzła Node. I aktywowane

```
Listing 6. Pętla główna procesu ConcentratorTask (plik ConcentratorTask.c)
while(1)
{
    /* Wait for event */
    uint32_t events = Event_pend(concentratorEventHandle, 0, CONCENTRATOR_EVENT_ALL, BIOS_WAIT_FOREVER);
    /* If we got a new ADC sensor value */
    if(events & CONCENTRATOR_EVENT_NEW_ADC_SENSOR_VALUE) {
        /* If we knew this Node from before, update the value */
        if(isKnownNodeAddress(latestActiveAdcSensorNode.address))
        {
            updateNode(&latestActiveAdcSensorNode);
        }
        else
        {
            /* Else add it */
            addNewNode(&latestActiveAdcSensorNode);
        }
        /* Update the values on the LCD */
        updateLcd();
    }
}
```



Rysunek 7. Załadowany projekt rfWsnConcentrator

jest zadanie ConcentratorTask. Odbiór pakietu jest sygnalizowany zmianą świecenia (przełączeniem) diody LED.

Zadanie ConcentratorTask

Zadanie ConcentratorTask realizuje obsługę aplikacyjną działania węzła Concentrator. Funkcja concentratorTaskFunction (plik ConcentratorTask.c) definiuje działanie zadania ConcentratorTask. Najpierw inicjalizowane są parametry pracy zadania ConcentratorTask. Potem inicjalizowany jest wyświetlacz LCD i wyświetlane są początkowe napisy. Inicjalizowane jest też łącze UART. Dalej jest uruchamiana główna pętla aplikacji. Na początku pętli jest wywołanie funkcji Event_pend. Zawiesza ona wykonanie zadania do czasu wykrycia przez system operacyjny zdarzenia (listing 6).

Jeśli zostało wygenerowane zdarzenie CONCENTRATOR_EVENT_NEW_ADC_SENSOR_VALUE, to sprawdzane jest, czy węzeł o tym numerze jest już w sieci. Jeśli tak, to aktualizowane są jego dane. Jeśli jest to węzeł o nowym identyfikatorze, to jest on dodawany do danych sieci. Wartość uzyskana w ostatniej transmisji jest wyświetlana (razem z adresem sieciowym) na wyświetlaczu LCD oraz wysłana przez kanał UART.

Obsługa wyświetlacza LCD i łącza UART dla węzła Concentrator

Informacja jest generowana na dwa sposoby:

- Dla wyświetlacza LCD.
- Dla łącza UART.

W trakcie inicjalizacji oprogramowania wykonywana jest próba inicjalizacji łącza UART (preferowane) oraz wyświetlacza LCD (poprzez łącze SPI). Dla zestawu CC1350/1310 Launchpad inicjalizowane są oba łącza.

Po inicjalizacji do wyświetlacza LCD i łącza UART jest wysyłany tekst „Waiting for Nodes...”. Sposób generowania tekstu dla wyświetlacza LCD i łącza UART jest taki sam i został pokazany

na **listingu 7** (funkcja `updateLcd`). Wyświetlacz LCD (Sharp LS013B4DN04) ma rozdzielczość 96×96 pikseli. Linia tekstu jest na tyle krótka, że nie mieści się napis „RSSI”.

Na początku wyświetlania tekstu każdej transmisji obecny tekst przesuwany jest w terminalu do góry. Na pustym ekranie terminalu wyświetlany jest nowy tekst.

W pierwszej linii wyświetlany jest opis, który oznacza:

- Nodes – adres sieciowy węzła *Node*.
- Value – wartość odczytu ADC węzła.
- SW – stan przycisku BTN-1 zestawu CC1350/1310 Launchpad węzła *Node* (1 – wciśnięty, 0 – zwolniony).
- RSSI – poziom odbieranego z węzła sygnału radiowego.

W następnych liniach wyświetlane są dane, po jednej linii dla każdego wykrytego w sieci węzła. Dla wyświetlacza LCD wartości RSSI nie zmieściły się w linii i nie są wyświetlane. Można to łatwo poprawić poprzez ścięnięcie nagłówka i linii danych.

Obsługa węzła Node sieci WSN dla zestawu CC1350 Launchpad

Do obsługi węzła *Node* sieci WSN używane są dwa przyciski oraz dwie diody LED (**rysunek 8**) [5]. Przyciski są wyraźnie opisane na płytce drukowanej zestawu CC1350 Launchpad [S9].

BTN1 (left) – przyciśnięcie zmienia tryb wysyłania:

- Wolny tryb wysyłania: Dane pomiarowe są wysyłane co 5 s lub gdy wartość odczytana zmieni się powyżej progu. Ustawiony domyślnie.
- Szybki tryb wysyłania: Dane pomiarowe są wysyłane co 1 sekundę bez względu na zmianę poziomu wartości odczytanej.

```
Listing 7. Obsługa wyświetlacza LCD i łącza UART (plik ConcentratorTask.c)
/* Clear the display and write header on first line */
Display_clear(hDisplayLcd);
Display_printf(hDisplayLcd, 0, 0, „Nodes Value SW RSSI”);
//clear screen, put cursor to beginning of terminal and print the header
Display_printf(hDisplaySerial, 0, 0, „\033[2J\033[0;0HNodes Value SW RSSI”);
/* Write one line per Node */
/* print to LCD */
Display_printf(hDisplayLcd, currentLcdLine, 0, „0x%02x %04d %d %04d”,
NodePointer->address, NodePointer->latestAdcValue, NodePointer->button,
NodePointer->latestRssi);
```

BTN2 (right) – przełącza pomiędzy trybami formatu pakietu rozgłaszania. Diody LED na zestawie sygnalizują wykonywane transmisji:

- **LED1** (DIO6, red) – Przełączana, gdy wysyłany jest pakiet danych w kanale Sub1-GHz.
- **LED2** (DIO7, green) – Przełączana, gdy wysyłany jest pakiet rozgłaszania w kanale BLE ISM 2,4 GHz.

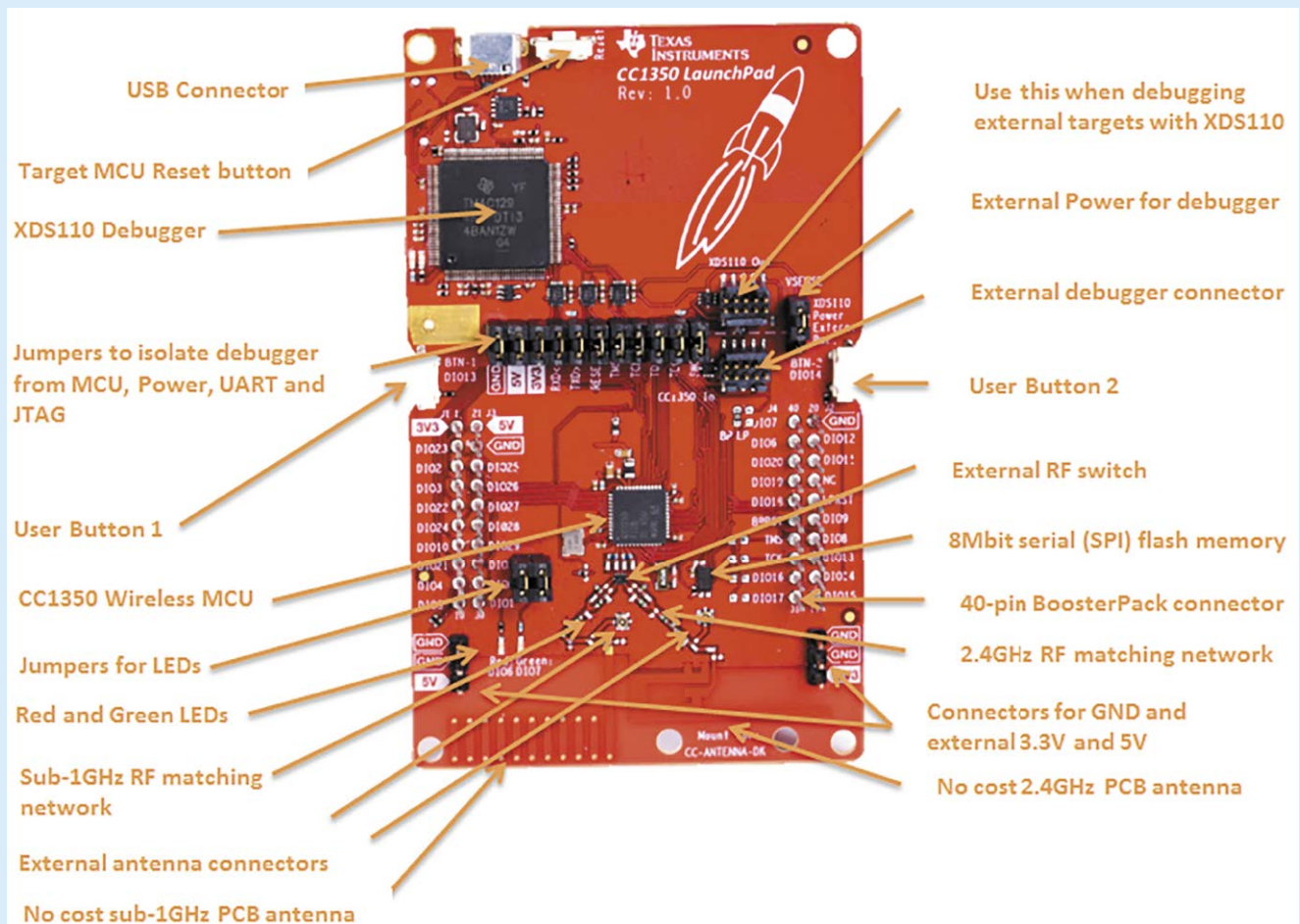
Należy zwrócić uwagę, że wykonanie transmisji jest sygnalizowane przełączeniem diody LED (a nie włączeniem). Czyli długi czas świecenia diody oznacza tylko długi czas bez wykonywania transmisji.

Obsługa zestawu CC1310 Launchpad

Dla zestawu CC1310 Launchpad są udostępnione w pakiecie CC13x0 SDK 1.50 następujące przydatne dwa projekty WSN. Są one kompatybilne z siecią WSN dwupasmową.

rfWsnNode – projekt dla węzła *Node*. Pracuje jak węzeł dla sieci dwupasmowej, lecz bez obsługi kanału BLE. Wysłanie pakietu danych sygnalizuje zmianą stanu diody LED (red). Nie obsługuje wyświetlacza LCD. Obsługuje przycisk BTN-1 oraz łącze UART.

rfWsnConcentrator – projekt dla węzła Concentrator. Pracuje jak węzeł dla sieci dwupasmowej, lecz bez obsługi kanału BLE. Odebranie pakietu danych sygnalizuje zmianą stanu diody LED (red). Obsługuje łącze UART i wyświetlacz LCD.



Rysunek 8. Elementy zestawu CC1350 Launchpad [8]

Działanie sieci WSN

Przykładowy komplet sprzętowy dla realizacji najprostszej sieci WSN dwupasmowej może wyglądać następująco:

Węzeł *Node*:

- CC1350EU LP (Launchpad).
- 430BOOST-SHARP96 LCD Boosterpacks.

Węzeł Concentrator:

- CC1350EU LP (Launchpad).
- 430BOOST-SHARP96 LCD Boosterpacks.

Formaty rozgłaszania BLE

Węzeł *Node* może zostać skonfigurowany do wysyłania pakietu rozgłaszania (beacon) BLE. Wybór formatu jest wykonywany cyklicznie poprzez naciśnięcie przycisku BTN-2 zestawu CC1350 Launchpad.

Dostępne są następujące formaty:

- BLE MS (Manufacturer Specific), domyślny format transmisji po włączeniu zasilania węzła *Node*. Przydatny do odbioru danych przez aplikację SimpleLinkStarter.
- Eddystone URL, format Google Open Source.
- Eddystone UID, format Google Open Source.
- None, bez nadawania.

Działanie węzła Node

Węzeł *Node* wysyła dane bezprzewodowo na dwa sposoby:

- Podstawowym kanałem przesyłania danych jest transmisja 50 kbps, GFSK (IEEE 802.15.4g) w kanale ISM 868 MHz. Transmisja jest dwukierunkowa.
- Drugim kanałem przesyłania danych jest transmisja formatu BLE w pasmie ISM 2,4 GHz.

Węzeł *Node* odczytuje dane z zastosowaniem rdzenia Sensor Controller co jedną sekundę. Potem (jeśli trzeba) wysyła dane poprzez łącze

```
COM103 - PuTTY
Node ID: 0x9f
Node ADC Reading: 0603
Advertiser Mode: BLE MS
Advertisement success: 4165 out of 4165
```

Rysunek 9. Węzeł *Node* (UART)



Rysunek 10. Węzeł *Node* (LCD)

Sub-1GHz. Następnie (jeśli trzeba) jest rekonfigurowany moduł radiowy do trybu BLE i wysyłany pakiet rozgłaszania. Po wykonaniu tej sekwencji węzeł *Node* przechodzi w uśpienie. Węzeł *Node* zostaje wybudzony przez timer po czasie jednej sekundy od rozpoczęcia poprzedniej sekwencji. Dla zestawu CC1350 Launchpad pakiet danych zawiera ostatni odczyt ADC, stan wciśnięcia przycisku BTN-1, czas od wysłania pierwszej transmisji (po uruchomieniu działania) oraz liczbę pakietów.

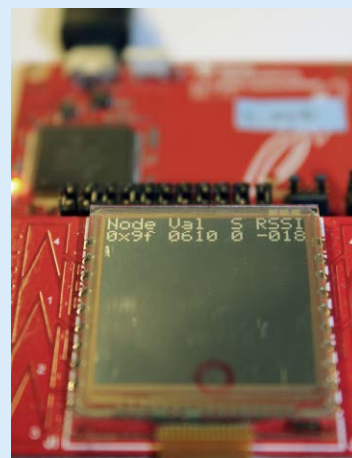
Poprzez łącze UART jest wysłana informacja tekstowa zawierająca identyfikator (adres sieciowy sieci WSN), wartość odczytu ADC oraz numer pakietu (rysunek 9). Jeśli do zestawu CC1350 Launchpad jest dołączony moduł LCD [9], to ta sama informacja jest na nim wyświetlana (rysunek 10). Pokazywane są tylko dane ostatniego transferu. Domyślnie jest też wysyłana informacja poprzez łącze BLE. Zawiera ona adres i pakiet danych.

Działanie węzła Concentrator

Węzeł *Concentrator* obsługuje dane bezprzewodowo z transmisją 50 kbps, GFSK (IEEE 802.15.4g) w kanale 868 MHz. Węzeł *Concentrator* pozostaje cały czas aktywny i gotowy do odbioru przychodzącej transmisji przez odbiornik. Obsługiwana jest też transmisja dwukierunkowa. Odebrane dane są wyświetlane na wyświetlaczu LCD oraz wysyłane poprzez łącze UART. Po włączeniu zasilania węzła *Concentrator* oczekuje on na odebranie transmisji od węzła sieci. Węzeł, który

```
COM104 - PuTTY
Nodes Value SW RSSI
0x9f 0636 0 -020
```

Rysunek 11. Węzeł *Concentrator* (UART)



Rysunek 12. Węzeł *Concentrator* (LCD)



Rysunek 13. Dwa węzły *Node* wykryte w sieci WSN (LCD)

Literatura

- [1] CC1350 SimpleLink Ultra-Low Power Dual Band Wireless Microcontroller, Product Page <https://goo.gl/irnJF2>
- [2] CC1350 SimpleLink™ Ultra-Low-Power Dual-Band Wireless MCU (SWRS183A) 21 Nov 2016 <https://goo.gl/HkimVw>
- [3] CC13xx, CC26xx SimpleLink Wireless MCU Technical Reference Manual (SWCU117G) 21 Feb 2017 <https://goo.gl/VdHLY9>
- [4] Simplelink CC1350 SensorTag Bluetooth and Sub-1GHz Long Range Wireless Development Kit CC1350 STK, Tools Page <https://goo.gl/LbKhhq>
- [5] Simplelink™ Sub-1 GHz and Bluetooth® low energy CC1350 wireless MCU Launchpad™ Development Kit, LAUNCHXL-CC1350, Tools Page <https://goo.gl/Ch6BAB>
- [6] SensorTag Debugger DevPack, CC-DEVPACK-DEBUG <https://goo.gl/63ddvK>

- [7] Google Play, Simplelink SensorTag, Texas Instruments Inc. (Version: 3.8, 4 listopada 2016) <https://goo.gl/QVTh3k>
- [8] Meet the CC1350 Launchpad <https://goo.gl/DmfMXE>
- [9] Sharp Memory LCD BoosterPack, 430BOOST-SHARP96 <https://goo.gl/mUC89w>
- [10] Download CCS (wersja CCS7.3.0.00019) <https://goo.gl/6Ho26s>
- [11] TI Resource Explorer <http://dev.ti.com/tirex/#/>
- [12] SimpleLink Sub-1 GHz CC13x0 Software Development Kit, SIMPLELINK-CC13X0-SDK, Ver 1.50.00.08 , 28-Sep-2017 <https://goo.gl/Rhxfr1>
- [13] SimpleLink Academy for SimpleLink CC13x0 SDK, Ver. 1.14.02.04, October 12th 2017 <https://goo.gl/Q9dRkU>
- [14] TI SimpleLink™ Starter By Texas Instruments (Version: 4.92, Sep 26, 2016) <https://goo.gl/53zfvv>

prześle pierwszy pakiet danych, jest wstawiany na listę sieci. Poprzez łącze UART jest wysłana informacja tekstowa zawierająca identyfikator (adres sieciowy sieci WSN), wartość odczytu A/C, stan wciśnięcia przycisku BTN-1 oraz poziom odbieranego przez węzeł Concentrator sygnału radiowego (rysunek 11). Jeśli do zestawu CC1350 Launchpad jest dołączony moduł LCD [9], to ta sama informacja jest na nim wyświetlana (rysunek 12).

Nodes	Value	SW	RSSI
0x9f	0604	0	-048
0xc7	3097	0	-032

Rysunek 14. Dwa węzły Node wykryte w sieci WSN (UART)

Sytuacja z dwoma wykrytymi węzłami Node sieci WSN jest pokazana na rysunku 13 i rysunku 14. Drugi węzeł pracuje z zestawem CC1350 SensorTag. Wyłączenie i ponowne włączenie zasilania węzła Node powoduje wygenerowanie jego nowego identyfikatora ID (adresu). Węzeł koncentratora dodaje go do listy sieci. Pomimo braku komunikacji z węzłem o poprzednim identyfikatorze jest on dalej wyświetlany na liście węzłów sieci WSN. Przykłady budowy dwupasmowej sieci WSN z zastosowaniem zestawu CC1350 Launchpad (oraz CC1350 SensorTag) są bardzo ciekawe i dobrze pokazują, w którą stronę może pójść rozwój sieci Internetu Rzeczy. Zagadnienie jest młode i na obecnym etapie cierpi na chorobę częstych i nagłych zmian.

Henryk A. Kowalski

REKLAMA

Klub Aplikantów Próbek

to inicjatywa redakcji Elektroniki Praktycznej.

W kontaktach z firmami redakcja często otrzymuje do przetestowania próbki podzespołów, modułów, a nawet całych urządzeń elektronicznych. Są to zwykle najnowsze typy/modeli produktów na rynku. Z chęci podzielenia się z Czytelnikami tymi próbkami zrodziła się inicjatywa pod nazwą Klub Aplikantów Próbek.

Członkiem KAP staje się każdy, kto zgłosi chęć przetestowania próbki. Wykaz i krótki opis próbek, którymi dysponuje redakcja EP, można znaleźć poniżej (www.ep.com.pl/KAP). Wystarczy wybrać rodzaj próbek i zwrócić się majłem (na adres: Szefer Pracowni Konstrukcyjnej grzegorz.becker@ep.com.pl) z prośbą o przesłanie bezpłatnych próbek, podając ich nazwę i adres wysyłki. Warto dopisać jaki jest plan zastosowania tych próbek. Nie jest to konieczne, ale może mieć znaczenie przy podziale próbek w przypadku większej liczby zgłoszeń. Mile widziane, choć nieobowiązkowe, jest też przysłanie do redakcji EP opisu wykonanej aplikacji próbek, oczywiście po jej wykonaniu z zastosowaniem otrzymanej próbki. Autorom przysłanych opisów przyznamy punkty, które będą im dawały pierwszeństwo przy ubieganiu się o kolejne próbki. Najciekawsze opisy aplikacji opublikujemy na forum ep.com.pl lub na łamach Elektroniki Praktycznej. Dla pełnej jasności jeszcze raz podkreślamy, że próbki przekazujemy bezpłatnie i nie trzeba ich zwracać do redakcji.



www.ep.com.pl/kap