



Internet Rzeczy w przykładach (2)

Konfigurowanie oprogramowania dla mikrokontrolera CC3200 i projekt „zero”

W tym artykule zaprezentujemy sposób pobrania, zainstalowania oraz skonfigurowania pakietu programów narzędziowych niezbędnych do pracy z mikrokontrolerem CC3200. Utworzymy też projekt bazowy, który tu nazwałem projektem „zero” i przygotujemy go do pracy z modułem CC3200 LaunchPad oraz technologią Internet Rzeczy.

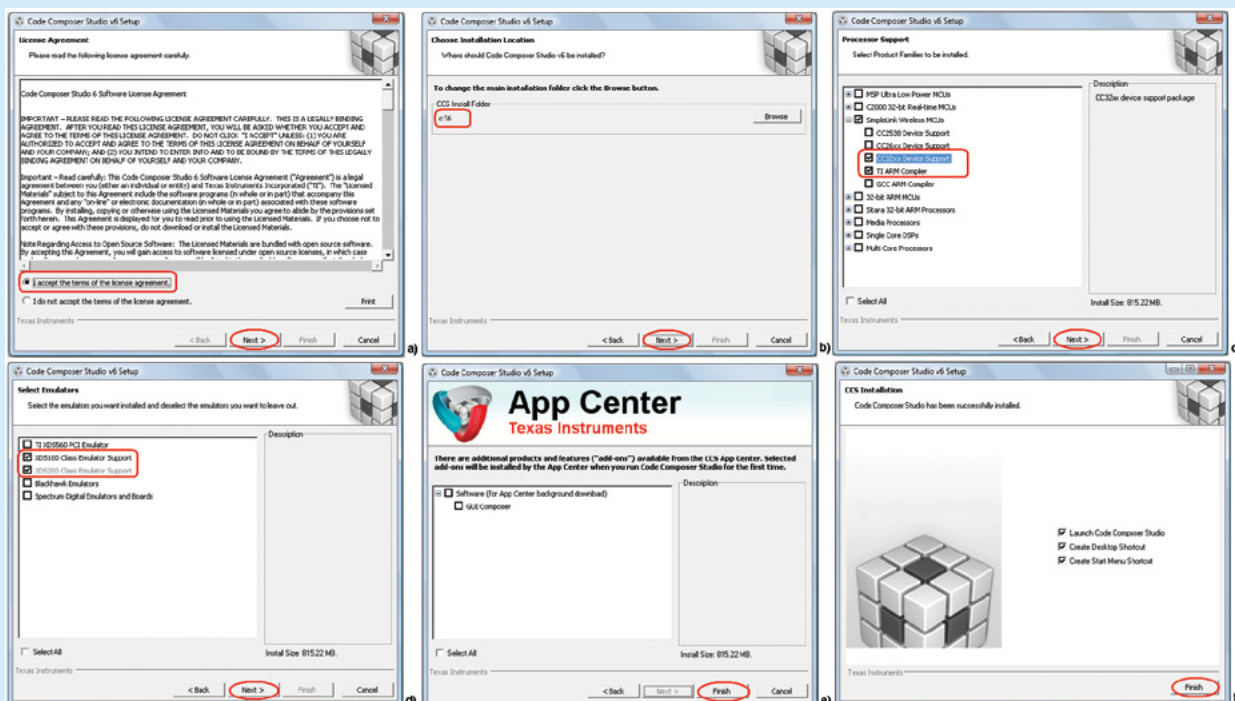
Mikrokontroler CC3200 jest obsługiwany przez kompilator GCC oraz przez kilka środowisk programistycznych, takich jak: Code Composer Studio (wersja 6.0.1+), IAR Embedded Workbench for ARM (wersja 7.20+), Energia, EmbedXCode. Wspierana jest obsługa systemów czasu rzeczywistego: FreeRTOS, TI-RTOS. Dostępny jest też przygotowany przez Texas Instruments pakiet oprogramowania CC3200 SDK zawierający ponad 50 przykładów

dla CCS, IAR, GCC, procedury SimpleLink API, biblioteki, sterowniki, schematy i szczegółową dokumentację. Do programowania zewnętrznej pamięci S-FLASH Texas Instruments przygotował aplikację *CCS UniFlash*, natomiast do konfigurowania linii wejścia-wyjścia mikrokontrolera aplikację *Pin Mux Tool*.

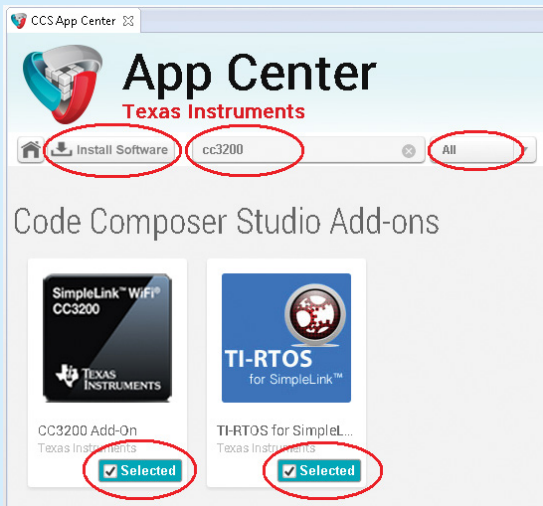
Środowisko CCSv6

Podczas kursu będziemy korzystali z rozwijanego przez Texas Instruments środowiska programistycznego Code Composer Studio wersja 6. W wersji darmowej CCSv6 dla mikrokontrolerów CC3200 ma ograniczony czas działania. Licencja darmowa jest aktywna przez 90 dni

Warunkiem koniecznym do pobrania oprogramowania ze strony Texas Instruments jest posiadanie konta w serwisie *ti.com*. Aby utworzyć konto, należy na stronie *ti.com* wypełnić formularz rejestracyjny (Login/Register). Rejestracja w serwisie *ti.com* jest bezpłatna.



Rysunek 1. Instalowanie Code Composer Studio v6: a) akceptowanie licencji, b) wskazanie lokalizacji, c) wybranie produktów do zainstalowania, d) instalowanie emulatora XDS100, e) pominięcie dodatków, f) zakończenie



Rysunek 2. Instalowanie oprogramowania CCSv6 App Center

od momentu zainstalowania oprogramowania, **jednak dla modułu startowego CC3200 LaunchPad, ograniczenie czasowe nie obowiązuje i możemy bezterminowo korzystać** z pełnej wersji oprogramowania!

Oprogramowanie CCSv6 można pobrać ze strony ti.com/tool/ccstudio. W sekcji *Download* są dostępne wersje CCSv6 dla systemów operacyjnych Windows oraz Linux. Ten kurs dotyczy wersji dla systemu Windows i to ją należy pobrać i zainstalować. W tym celu logujemy się na konto w serwisie *ti.com* i wypełniamy formularz informacyjny. Po przesłaniu formularza, na adres e-mail podany w jego treści otrzymamy link do pobrania oprogramowania (aktywny przez 6 dni).

Pobieramy oprogramowanie CCSv6 oraz uruchamiamy instalator. W kolejnych krokach konfigurujemy parametry instalacji. Akceptujemy licencję (rysunek 1a). Wybieramy miejsce instalacji oprogramowania. Zalecana lokalizacja to `c:\ti` (rysunek 1b). Zaznaczamy produkty, dla których ma być przeprowadzona instalacja. Nam potrzebne będą: *TI ARM Compiler* oraz *CC32xx Device Support* (rysunek 1c). Potwierdzamy instalację emulatora

XDS100 (rysunek 1d) i przechodzimy dalej. W tym momencie rozpoczyna się instalowanie oprogramowania. Na koniec jesteśmy pytani o dodatki. Pomijamy ten krok (rysunek 1e). Następnie potwierdzamy zakończenie instalowania CCS (rysunek 1f).

Na pulpicie i w menu start jest tworzony – po kliknięciu rozpoczyna się pierwsze uruchomienie CCS v6. Po uruchomieniu aplikacji w zakładce *View* zaznaczamy opcje *CSS App Center*. Następnie, w polu wyszukiwania *App Center* wpisujemy frazę *cc3200*, a w opcjach wyszukiwania wybieramy funkcję *All*. Zostaną wyszukane aplikacje: *CC3200 Add-On*, oraz *TI-RTOS*. Zaznaczamy obie aplikacje oraz naciskamy przycisk *Install Software* (rysunek 2). Powoduje to uruchomienie instalatora. Akceptujemy postanowienia licencyjne (dla obu aplikacji), a po zainstalowaniu obu dodatków ponownie uruchamiamy CCS.

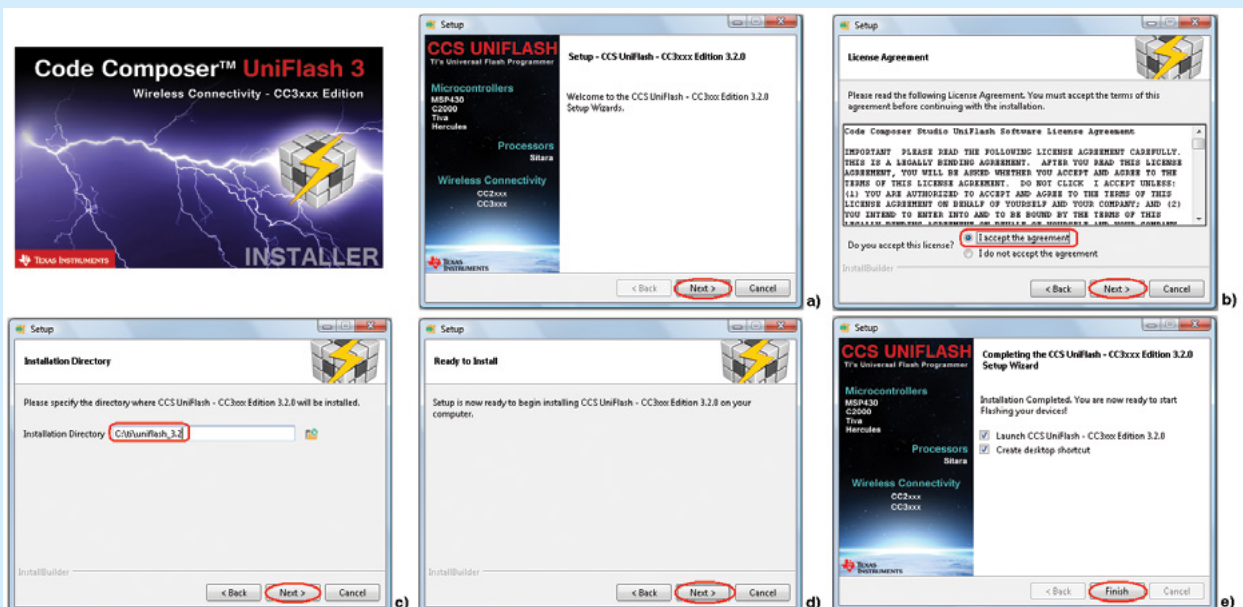
CCS UniFlash – programowanie pamięci S-Flash

Mikrokontroler CC3200 ma pamięci RAM i ROM, natomiast brak w jego strukturze programowalnej pamięci nieulotnej Flash. Aby pracować z CC3200, należy dołączyć zewnętrzną pamięć programu. W module startowym CC3200 LaunchPad zamontowano pamięć S-Flash o pojemności 1 MB. Po włączeniu zasilania mikrokontroler CC3200 kopiuje program z zewnętrznej pamięci S-Flash do wewnętrznej pamięci RAM, skąd jest on uruchamiany.

Aby umożliwić programowanie zewnętrznej pamięci S-Flash zainstalowanej w module CC3200 LaunchPad firma Texas Instruments przygotowała aplikację *CCS*

Instalowanie sterownika FTDI

Korzystając z modułu startowego LaunchPad należy zainstalować dostarczony przez Texas Instruments sterownik FTDI. Plik sterownika jest dystrybuowany razem z aplikacjami CCSv6, CC3200 SDK, SDK UniFlash. Jeśli w systemie jest zainstalowane oprogramowanie CCSv6, to po dołączeniu modułu LaunchPad do złącza USB komputera PC sterownik zostanie zainstalowany automatycznie. Powoduje to utworzenie w systemie wirtualnego portu COM o nazwie *CC3200LP Dual Port (Panel sterowania → Menedżer urządzeń → Porty COM i LPT)*. Utworzony port COM służy do programowania i sterowania pracą emulatora mikrokontrolera CC3200 oraz do przesyłania danych do PC przez UART. Dodatkowo jest on używany podczas komunikacji z zewnętrzną pamięcią S-Flash (programowanie, kasowanie itd.).



Rysunek 3. Instalowanie CCS UniFlash: a) uruchomienie instalatora, b) akceptowanie warunków licencji, c) wybór lokalizacji, d) rozpoczęcie instalowania, e) koniec pracy instalatora

Aktualizowanie oprogramowania

Aby zaktualizować oprogramowanie dla mikrokontrolera CC3200 (*) należy wykonać następujące czynności:

Konfigurujemy moduł LaunchPad w trybie programowania pamięci S-Flash (rysunek 5).

Dołączamy moduł LaunchPad do portu USB komputera PC.

Uruchamiamy program *CCS UniFlash*, podajemy numer portu COM dla modułu LaunchPad.

Z zakładki *CC31x Flash Setup and Control* programu *CCS UniFlash* (okno po lewej stronie aplikacji) wybieramy */sys/mcuimg.bin*.

W nowo otwartym oknie programu (rysunek 6) zaznaczamy opcje *Erase, Update, Verify*.

W sekcji *Url* podajemy adres pliku binarnego programu, który chcemy wgrać (naciskamy przycisk *Browse* oraz odnajdujemy plik na dysku komputera PC – rys. 6).

Z menu programu *CCS UniFlash* wybieramy *Operation*, a następnie *Program* (rys. 6).

Rozpoczyna się programowanie pamięci S-Flash. Jeśli zostaniemy poproszeni o restart mikrokontrolera, należy nacisnąć przycisk SW1.

Po zakończeniu aktualizowania oprogramowania, zamykamy program *CCS UniFlash* oraz odłączamy moduł LaunchPad od portu USB komputera PC.

Zmieniamy konfigurację sprzętową modułu LaunchPad (z pola SOP usuwamy zworkę z pozycji numer 2 „Flash”, zworki J6 i J7 ustawiamy w pozycji BP).

Po włączeniu zasilania moduł LaunchPad mikrokontroler CC3200 skopiuje program wgrany do pamięci S-Flash do wewnętrznej pamięci RAM i aplikacja zostanie uruchomiona.

(*) oprogramowanie dla CC3200 możemy również aktualizować korzystając z *CCSV6* (LaunchPad w trybie programowania pamięci S-Flash, *CCSV6* w trybie debugowania projektu).

UniFlash. Plik instalatora *CCS UniFlash* dla CC3200 pobieramy ze strony ti.com/tool/uniflash. Procedura pobierania oprogramowania jest identyczna, jak w wypadku *CCSV6*. Po zapisaniu oprogramowania na dysku uruchamiamy instalator (rysunek 3a). Akceptujemy postanowienia licencyjne (rysunek 3b). Zatwierdzamy proponowane przez instalator miejsce instalacji (rysunek 3c) i potwierdzamy chęć zainstalowania programu (rysunek 3d). W tym momencie rozpoczyna się instalowanie oprogramowania. Gdy oprogramowanie *CCS UniFlash* zostanie zainstalowane, kończymy pracę instalatora (rysunek 3e).

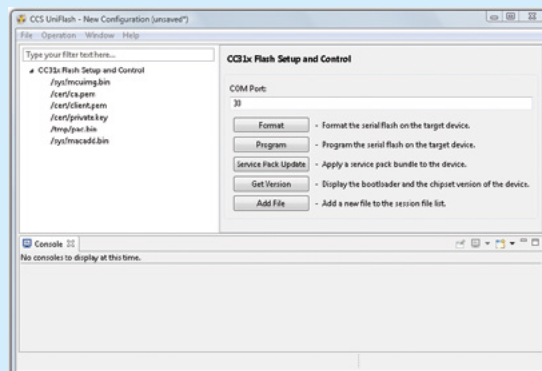
Na pulpicie jest tworzony skrót do oprogramowania. Przy pierwszym uruchomieniu programu jest wyświetlany ekran konfiguracji *CCS UniFlash*. Z dostępnych opcji konfiguracyjnych wybieramy *New Target Configuration* i zatwierdzamy ustawienia dla CC3200: *Connection* → *CC3x Serial(UART) Interface, board or device* → *SimpleLink Wifi CC3100/CC3200*. Następnie jest pokazywany ekran główny programu *CCS UniFlash*. (rysunek 4).

Dostęp do pamięci S-Flash zainstalowanej na płycie modułu LaunchPad jest możliwy po odpowiednim, sprzętowym skonfigurowaniu modułu. W miejscu portów P1...P3 (złącze 20-pinowe po lewej stronie) nie należy montować żadnych zworek. W polu SOP ustawiamy zworkę w pozycji numer 2 (Flash). Montujemy zworki w polach od J8 do J13. Zworki J6 i J7 ustawiamy w pozycji „FLASH”. Ustawienie zworek konfiguracyjnych pokazano na **rysunku 5**.

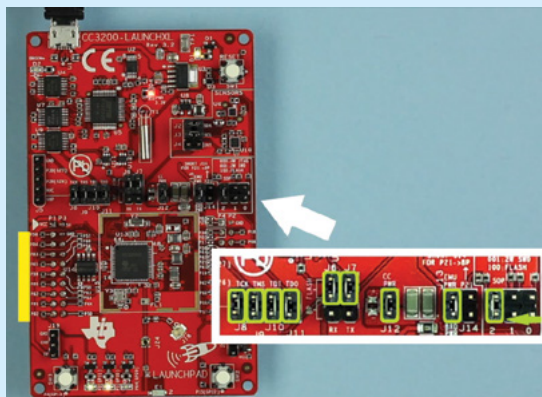
Dodatkowo, w programie *CCS UniFlash* w polu *COM Port* należy podać numer portu COM do obsługi modułu LaunchPad. Uruchamiamy Menedżer urządzeń i w zakładce *Porty (COM i LPT)* odnajdujemy *CC3200LP Dual Port*. Odczytujemy numer portu COM i wprowadzamy w aplikacji *CCS UniFlash*.

Pakiet CC3200 SDK

Pakiet oprogramowania *CC3200 SDK* (Software Development Kit) został przygotowany przez Texas Instruments w celu ułatwienia pracy z mikrokontrolerem CC3200 oraz modułem startowym CC3200 LaunchPad.



Rysunek 4. Ekran główny programu *CCS UniFlash*

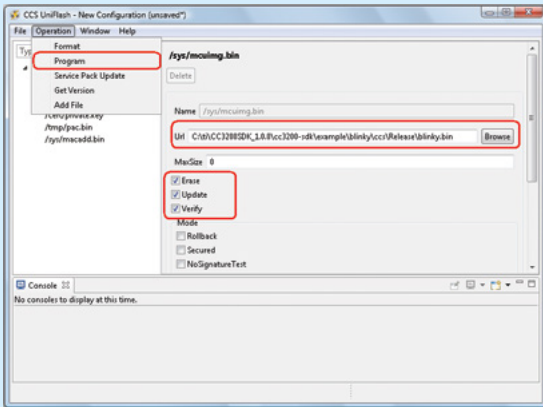


Rysunek 5. Programowanie pamięci S-FLASH. Ustawienie zworek konfiguracyjnych modułu LaunchPad

W ramach pakietu SDK jest oferowanych ponad 50 przykładowych projektów, które można uruchomić korzystając z płytki startowej LaunchPad (folder *example*). Przykłady zostały napisane w języku C i są to aplikacje sieciowe oraz aplikacje do obsługi peryferiów mikrokontrolera. Projekty utworzono dla CCS, IAR, a w niektórych wypadkach również dla GCC. Dodatkowo, w SDK udostępniono projekty do wygenerowania bibliotek obsługi: serwera http, klienta tftp, klientów smtp i xmpp (folder *netapps*). Innym elementem SDK jest biblioteka do obsługi modułów peryferyjnych mikrokontrolera CC3200 (folder *driverlib*). W pamięci ROM mikrokontrolera umieszczono procedury obsługi USART, I²C, SPI, timerów, przetwornika A/C itp. Programista może używać sterowników korzystając z dostarczonego przez Texas Instruments API (dokumentacja w folderze *docs*).

Kluczowym elementem pakietu SDK jest SimpleLink Framework (folder *simplelink*). Przygotowany przez Texas Instruments framework zawiera zbiór procedur do obsługi sieci Wi-Fi (obsługa gniazdek UDP/TCP, adresacja IP/MAC, skanowanie AP, łączenie z AP, usługa *ping*, itd.). Procedury SimpleLink mogą być obsługiwane bez użycia systemu czasu rzeczywistego lub z użyciem systemu FreeRTOS lub TI-RTOS. Ten oraz inne parametry konfigurujemy w pliku *user.h* Do obsługi SimpleLink wykorzystujemy dostarczone przez Texas Instruments API (dokumentacja w folderze *docs/simplelink_api*). W pakiecie SDK jest dostępne również oprogramowanie firm trzecich (folder *third_party*). Są to system czasu rzeczywistego FreeRTOS oraz system plików FatFs.

Plik instalacyjny CC3200 SDK pobieramy ze strony ti.com/tool/cc3200sdk. Proces pobierania oprogramowania jest identyczny jak w przypadku *CCSV6*. Podczas



Rysunek 6. Aktualizowanie oprogramowania dla CC3200

instalowania akceptujemy postanowienia licencyjne (**rysunek 7b**), zatwierdzamy proponowaną lokalizację plików `c:\ti\` (**rysunek 7c**). Kolejne kroki instalatora pokazano na **rysunku 7**.

CC3200 SDK Service Pack

Dodatek Service Pack dla pakietu CC3200 SDK zawiera plik binarny z aktualną wersją oprogramowania dla mikrokontrolera CC3200. Dodatek Service Pack należy wgrać do zewnętrznej pamięci S-Flash zainstalowanej na płycie modułu CC3200 LaunchPad.

Dodatek ServicePack dla CC3200 SDK pobieramy ze strony ti.com/tool/cc3200sdk (*CC3200SDK-ServicePack*). Proces pobierania oraz instalowania jest identyczny, jak w wypadku pakietu oprogramowania CC3200 SDK.

Aby uaktualnić Service Pack przełączamy moduł CC3200 LaunchPad w tryb programowania pamięci S-Flash (**rysunek 5**). Dołączamy moduł LaunchPad do portu USB komputera PC i uruchamiamy oprogramowanie *CCS UniFlash*. Z dostępnych opcji programu wybieramy przycisk *Format* (pojemność pamięci 1 MB). Następnie restartujemy moduł LaunchPad naciskając SW1. Rozpoczyna się kasowanie pamięci S-Flash.

Po zakończeniu kasowania z opcji programu wybieramy przycisk *ServicePack Update*. Na dysku komputera PC odnajdujemy plik binarny z oprogramowaniem Service Pack (plik z rozszerzeniem *bin* w folderze instalacyjnym Service Packa) i zatwierdzamy aktualizowanie. Restartujemy moduł LaunchPad (przycisk SW1) i czekamy na zakończenie aktualizowania oprogramowania. Po zakończeniu zamykamy aplikację *CCS UniFlash*.

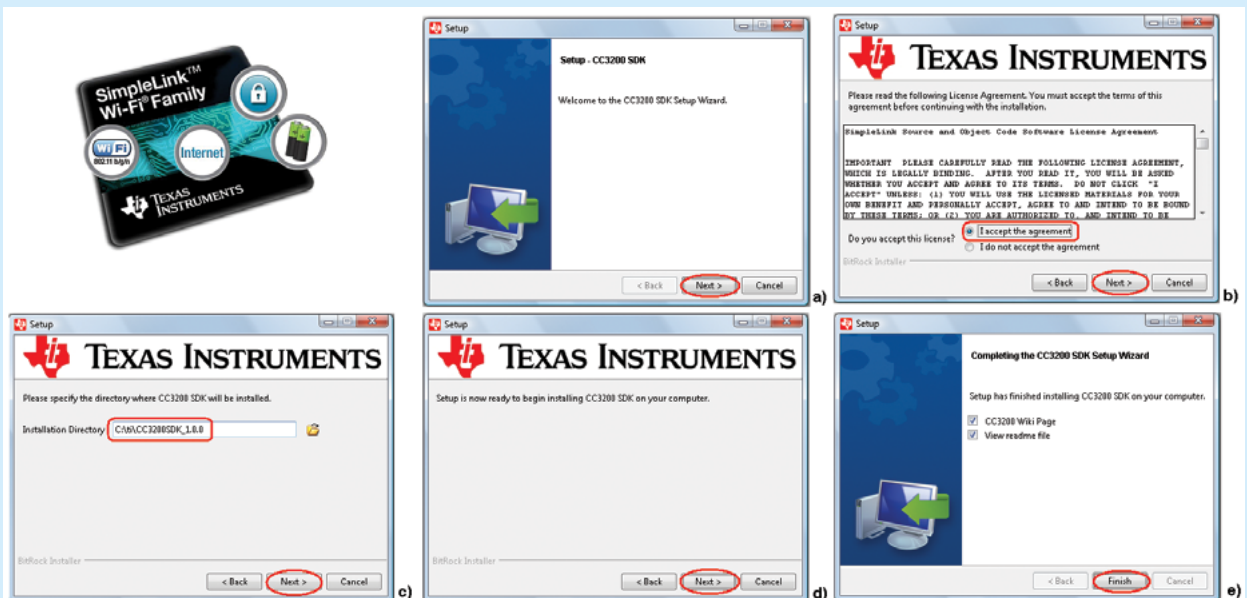
Narzędzie do konfigurowania wyprowadzeń Pin Mux Tool

Aby ułatwić konfigurowanie linii wejścia-wyjścia mikrokontrolera CC3200 firma Texas Instruments udostępniła aplikację *Pin Mux Tool*. Można ją pobrać ze strony ti.com/tool/pinmuxtool (PINMUXTOOL-V3 dla CC3200). Proces pobierania oprogramowania jest identyczny jak w przypadku CCSv6. Instalacja przebiega analogicznie jak w przypadku CCS UniFlash. Kolejne kroki instalacji pokazano na **rysunku 8**.

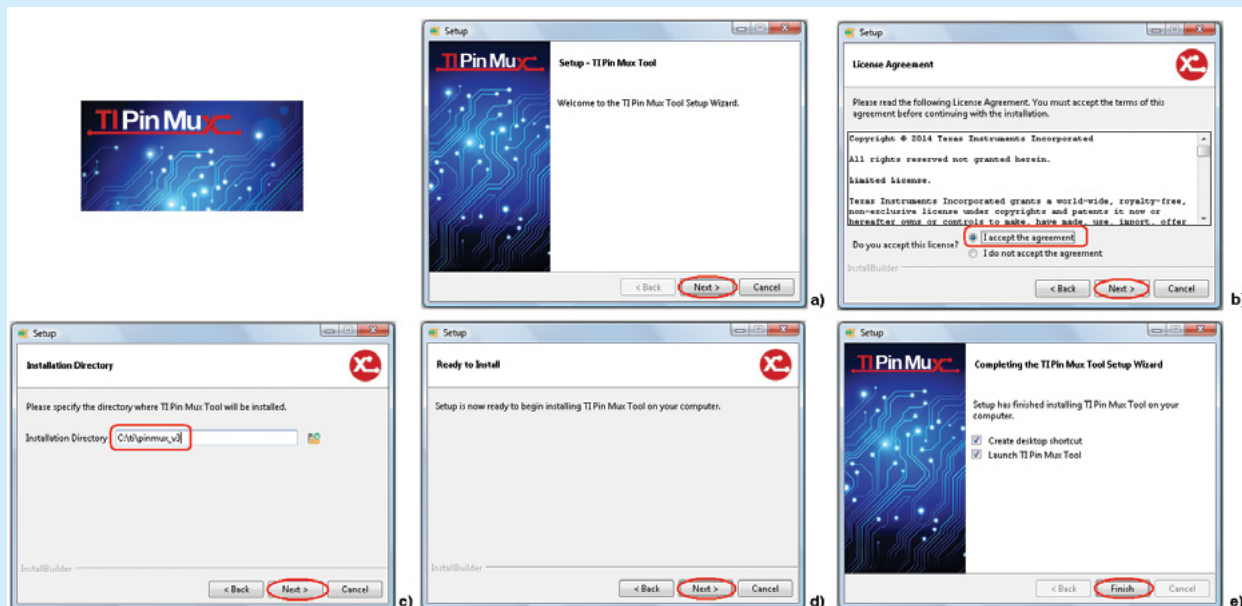
Obsługa programu jest łatwa i intuicyjna. Uruchamiamy *Pin Mux Tool*. Tworzymy nowy projekt dla mikrokontrolera CC3200 (*Start a new project* → *Device* – *CC3200* → *Start*). Gdy program zostanie uruchomiony, dodajemy moduły, dla których chcemy skonfigurować linie wejścia-wyjścia (np.: SPI, UART, GPIO, A/C). Definiujemy konfigurację dla wyprowadzeń mikrokontrolera. Na zakończenie generujemy plik wynikowy z konfiguracją (przycisk *Generate* w opcjach programu). We wskazanej lokalizacji na dysku komputera PC tworzone są pliki: *pin_mux_config.c*, *rom_pin_mux_config.c* oraz *pin_mux_config.h*. Do projektu CCSv6 dołączamy plik *pin_mux_config.h* oraz jeden z plików z rozszerzeniem „c” (*pin_mux_config.c*, jeśli chcemy korzystać z procedur obsługi linii we-wy z biblioteki drivera, *rom_pin_mux_config.c*, jeśli chcemy korzystać z procedur zapisanych w pamięci ROM).

Projekt „zero”

Projekty dla mikrokontrolera CC3200 (płytki startowa LaunchPad) będziemy tworzyli z wykorzystaniem środowiska programistycznego CCSv6. W projektach



Rysunek 7. Instalowanie CC3200 SDK: a) ekran powitalny instalatora, b) akceptowanie warunków licencji, c) wskazanie folderu, d) rozpoczęcie instalowania, e) koniec pracy instalatora



Rysunek 8. Instalowanie Pin Mux Tool: a) ekran powitalny instalatora, b) akceptowanie warunków licencji, c) wskazanie folderu, d) rozpoczęcie instalowania, e) koniec pracy instalatora

Tabela 1. Pliki dołączane do projektu dla CC3200

Lokalizacja plików	Opis
„\${CC3200_SDK_ROOT}/inc”	Pliki nagłówkowe dla cc3200
„\${CC3200_SDK_ROOT}/example/common”	Biblioteki peryferiów dla cc3200
„\${CC3200_SDK_ROOT}/driverlib”	Sterowniki peryferiów dla cc3200
„\${CC3200_SDK_ROOT}/simplelink/”	SimpleLink framework
„\${CC3200_SDK_ROOT}/simplelink/include”	
„\${CC3200_SDK_ROOT}/simplelink/source”	
„\${CC3200_SDK_ROOT}/oslib/”	System operacyjny RTOS

Tabela 2. Biblioteki dołączane do projektu dla CC3200

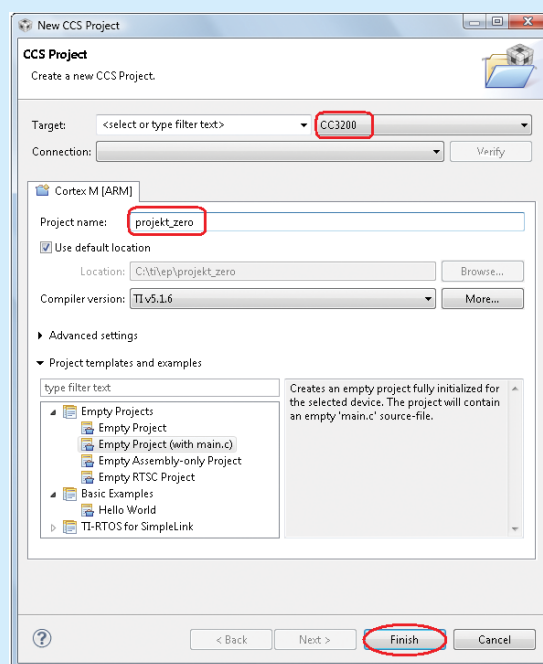
Nazwa biblioteki	Lokalizacja biblioteki	Opis
driverlib.a	„\${CC3200_SDK_ROOT}/driverlib/ccs/Release”	Drivery
simplelink.a	„\${CC3200_SDK_ROOT}/simplelink/ccs/OS”	SimpleLink
free_rtos.a	„\${CC3200_SDK_ROOT}/oslib/ccs/free_rtos”	FreeRTOS

będziemy korzystali z frameworka *SimpleLink*, ze sterowników sprzętowych *driverlib* oraz z systemu czasu rzeczywistego *freeRTOS*. Aby zademonstrować konfigurowanie projektu dla CC3200 LaunchPad utworzymy projekt startowy tzw. projekt „zero”.

Uruchamiamy środowisko programistyczne CCSv6. Ustalamy lokalizację dla projektów CCSv6 (*Select a workspace*). W naszym wypadku będzie to folder ep w lokalizacji c:\ti (wpisujemy c:\ti\ep w polu *workspace*). Następnie tworzymy nowy „pusty” projekt dla mikrokontrolera CC3200. W tym celu menu programu wybieramy opcję *Project* → *New CCS Project*. Zaznaczamy mikrokontroler CC3200. Podajemy nazwę projektu. Zatwierdzamy ustawienia przyciskiem *Finish* (rysunek 9).

Gdy projekt zostanie utworzony przechodzimy do opcji konfiguracyjnych projektu (*Project* → *Properties*). Dostępne są cztery podstawowe zakładki: *Resource*, *General*, *Build*, *Debug* (rysunek 10).

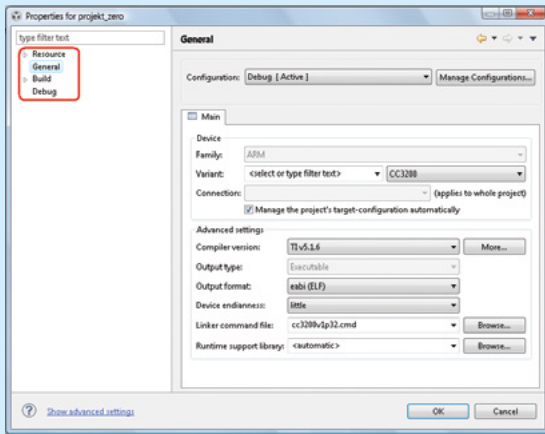
Przechodzimy do zakładki *Resource* i w polu *Linked Resources* w zakładce *Path Variables* dodajemy zmienną środowiskową *CC3200_SDK_ROOT* odnoszącą się do folderu *cc3200-sdk* z pakietu oprogramowania *CC3200 SDK*.



Rysunek 9. Tworzenie projektu dla CC3200

Następnie przechodzimy do zakładki *Build* gdzie w polu *ARM Compiler* wprowadzimy ustawienia dla kompilatora. W polu *Processor Options* ustawiamy wsparcie dla liczb zmiennoprzecinkowych (*-float_support* funkcja *fpalib*). W polu *Advanced Options* → *Predefined Symbols* definiujemy flagi: *USE_FREERTOS* (użycie systemu czasu rzeczywistego *freeRTOS*), *SL_PLATFORM_MULTI_THREADED* (framework *SimpleLink* w trybie obsługi wielu wątków). W polu *Include Options* (dołączanie plików źródłowych) wprowadzamy ustawienia, jak w tabeli 1.

Następnie, nie opuszczając zakładki *Build*, w polu *ARM Linker* wprowadzamy ustawienia dla linkera. W polu *Basic Options* ustawiamy rozmiar sterty (0x8000) oraz stosu programu (0x800). W polu *File Search Path* dodajemy pliki z bibliotekami dla driverów, *SimpleLink* oraz *FreeRTOS*. Dodatkowo, podajemy ścieżki dostępu do bibliotek. Ustawienia umieszczono w tabeli 2.



Rysunek 10. Konfiguracja projektu dla CC3200

Konfigurację debuggera (zakładka *Debug*) importujemy z pliku dostarczonego w pakiecie oprogramowania *CC3200 SDK*. Zamykamy konfigurację projektu i z menu programu wybieramy *View* → *Target Configurations*. W oknie *Target Configurations* zaznaczamy opcję *User Defined*. Naciskamy prawy przycisk myszy i z dostępnych opcji wybieramy *Import Target Configuration*. W miejscu instalacji pakietu oprogramowania *CC3200*

W projektach dla CC3200 korzystać będziemy z bibliotek: *driverlib*, *simplelink*, *free_rtos*. Kody źródłowe bibliotek w wersjach dla CCS, IAR, GCC dostępne są w pakiecie oprogramowania *CC3200-SDK*.

SDK w folderze *cc3200-sdk/tools/ccs_patch* odnajdujemy plik *CC3200.ccxml*. Importujemy plik z opcją kopiowania do projektu. Następnie w oknie *Target Configurations* zaznaczamy zaimportowany plik *CC3200.ccxml* i naciskamy prawy przycisk myszy. Z dostępnych opcji wybieramy *Set as Default*. Konfiguracja debuggera zostaje przypisana do projektu.

Podsumowanie

W tym artykule opisano sposób zainstalowania pakietu programów niezbędnych do pracy z mikrokontrolerem CC3200. Korzystając z środowiska programistycznego CCSv6 utworzyliśmy projekt startowy tzw. projekt zero dla CC3200. W kolejnym numerze EP zaprezentujemy projekt sterownika inteligentnej szafy na ubrania. Urządzenie odczyta z Internetu prognozę pogody i na jej podstawie zaproponuje użytkownikowi odpowiedni zestaw ubrań. Za bazę sprzętową projektu posłuży nam płyta startowa CC3200 LaunchPad.

Łukasz Krysiwicz, EP

Serwisy www

dla branży elektroniki i automatyki



ElektronikaB2B
Portal branżowy dla elektroników



AutomatykaB2B
Portal branżowy dla automatyków

**ELEKTRONIKA
PRAKTYCZNA**



ponad
500 000
odstów miesięcznie

ponad
140 000
użytkowników miesięcznie



ponad
11 000
subskrybentów codziennego newslettera