

C2000 Piccolo LaunchPad (14)

Łatwe programowanie w środowisku Code Composer Studio v6

Pojawienie się nowej wersji środowiska CCSv6.0.1 oraz nowej wersji pakietu controlSUITEv3.2.7 jest doskonałą okazją do zweryfikowania pracy z tym oprogramowaniem. Środowisko Code Composer Studio (CCS) dostarczane przez firmę Texas Instruments jest przeznaczone do tworzenia programów dla wszystkich procesorów firmy Texas Instruments. Pakiet programowy controlSUITE tej firmy zawiera oprogramowanie firmware, biblioteki, opisy zestawów sprzętowych oraz projekty przykładowe dla wszystkich serii procesorów rodziny C2000.

Code Composer Studio jest zintegrowanym środowiskiem projektowym – IDE (*Integrated Development Environment*) dostarczonym przez firmę Texas Instruments. Podstawowe narzędzia do tworzenia programów dla układów procesorowych rodziny TMS320C2000 to kompilator języka C/C++ oraz asembler i linker. Kompilator C/ C++ narzędzi programowych środowiska CCSv6 jest w pełni zgodny ze standardem ISO C/C++. Jednak ze względu na efektywność wykorzystania specyficznych cech zastosowanego układu procesorowego sygnałowego rodziny TMS320C2000 zostały wprowadzone dodatkowe cechy.

Pakiet programowy „firmware” (F2802x Firmware Development Package), będący częścią pakietu programowego controlSUITE, dostarcza wsparcia dla dwóch modeli programowania układów procesorowych serii Piccolo F2802x [20]. Jest to:

- model bezpośredniego dostępu do rejestrów (header files),
- model driverów programowych (library).

Każdy z tych modeli może być zastosowany osobno lub łącznie. Opis jest zamieszczony w dokumentach [20, 21] dostępnych w ścieżce `\doc` pakietu programowego „firmware”.

Biblioteka *driverlib* dostarcza API do sterowania modułami peryferyjnymi układów procesorowych serii Piccolo F2802x. Jednak praca z projektami z użyciem tej biblioteki powoduje pewne kłopoty.

Odinstalowanie środowiska CCS oraz pakietu controlSUITE

Przed rozpoczęciem instalowania należy odinstalować poprzednie środowisko CCS oraz pakiet *controlSUITE*. Najpierw w oknie *Programy i funkcje w Panelu sterowania* należy wykonać odinstalowanie w następującej kolejności: wszystkie pakiety sterowników FTDI, pakiet sterowników Spectrum Digital, drajwer Blackhawk Emulation, pakiet DSPBIOS oraz pakiet controlSUITE (jeśli jest). Dopiero teraz odinstaluj CCS. Może to potrwać dosyć długo. Na koniec należy wejść do foldera `C:\ti` i usunąć z niego wszystko. Dopiero teraz można rozpocząć instalowanie nowego oprogramowania. A to nie zapewnia dobrego wyczyszczenia komputera. Pozostają dotychczasowe bazy projektów przykładowych. Można je będzie usunąć po zainstalowaniu nowego oprogramowania.

Pobieranie, instalacja, uruchamianie i aktualizacja środowiska CCSv6.x

Opis wymagań systemowych, pobierania i instalowania oraz wyboru wersji licencji CCSv6 jest zamieszczony na stronie TI CCS [15] oraz na stronie TI Wiki [18]. Na stronie *Download CCS* [16] dostępna jest aktualna wersja plików instalacyjnych CCSv6 do pobrania (osobno dla systemu Windows i Linux).

Obecnie aktualna jest wersja CCSv6.0.1.00039 (datowana *July 22, 2014*). Są dwie opcje instalacji:

- **Web Installer** – pobierany jest mały plik programu instalatora, który automatycznie pobiera i instaluje środowisko CCSv6.

Dotychczas w EP na temat zestawu ewaluacyjnego C2000 Piccolo LaunchPad:

- [1] „Zestaw ewaluacyjny C2000 Piccolo LaunchPad”, EP 01/2013
- [2] „C2000 Piccolo LanuchPad (1) – Pierwszy program w środowisku programowym CCS v5”, EP 02/2013
- [3] „C2000 Piccolo LanuchPad (2) – łatwe programowanie z pakietem controlSUITE”, EP 03/2013
- [4] „C2000 Piccolo LanuchPad (3) – łatwe programowanie do pamięci Flash”, EP 04/2013
- [5] „C2000 Piccolo LanuchPad (4) – łatwa obsługa szyny SPI”, EP 05/2013
- [6] „C2000 Piccolo LanuchPad (5) – łatwa obsługa szyny I²C”, EP 07/2013
- [7] „C2000 Piccolo LanuchPad (6) – łatwa inicjalizacja systemowa procesora serii Piccolo F2802x”, EP 09/2013
- [8] „C2000 Piccolo LanuchPad (7) – łatwa obsługa wyświetlacza LCD”, EP 11/2013
- [9] „C2000 Piccolo LanuchPad (8) – Budowanie biblioteki *drivelib* dla procesorów serii Piccolo F2802x”, EP 12/2013
- [10] „C2000 Piccolo LanuchPad (9) – łatwa obsługa modułu PWM procesora serii Piccolo F2802x”, EP 1/2014
- [11] „C2000 Piccolo LanuchPad (10) – łatwa obsługa modułu eCAP procesora serii Piccolo F2802x”, EP 2/2014
- [12] „C2000 Piccolo LanuchPad (11) – łatwe sterowanie diodami LED-RGB mocy”, EP 3/2014
- [13] „C2000 Piccolo LanuchPad (12) – łatwy pomiar koloru”, EP 4/2014
- [14] „C2000 Piccolo LanuchPad (13) – łatwe programowanie systemów czasu rzeczywistego”, EP 5/2014

- **Off-line Installer** – pobierany jest plik spakowany zip (ok. 765 MB) zawierający kompletne pliki instalacyjne środowiska CCSv6. Dostępny jest również plik ze skrótem MD5 pobieranego pliku zip.

Do pobrania pliku instalacyjnego jest potrzebna wcześniejsza rejestracja na stronie myTI [17]. Po kliknięciu na odnośnik odpowiedniej opcji wyboru otwierane jest okno formularza aprobaty eksportowej. Należy wypełnić go w języku angielskim. Po wysłaniu formularza otwiera się okno aprobaty z odnośnikiem ładowania pliku. Dodatkowo wysyłany jest e-mail z tym samym odnośnikiem (aktywny 72 godziny).

Bezpieczniej jest pobrać kompletny plik instalacyjny. Pozwoli to na uniknięcie kłopotów, ponieważ typowa instalacja trwa około 30 min. Pobieranie pliku instalacyjnego wymaga każdorazowej rejestracji i użycia aprobaty.

Poprawność pobranego pliku *CCS6.0.1.00039_win32.zip* można zweryfikować z zastosowaniem pobranego skrótu MD5. Można się na przykład posłużyć darmowym programem WinMD5Free dostępnym pod adresem <http://www.winmd5.com/>.

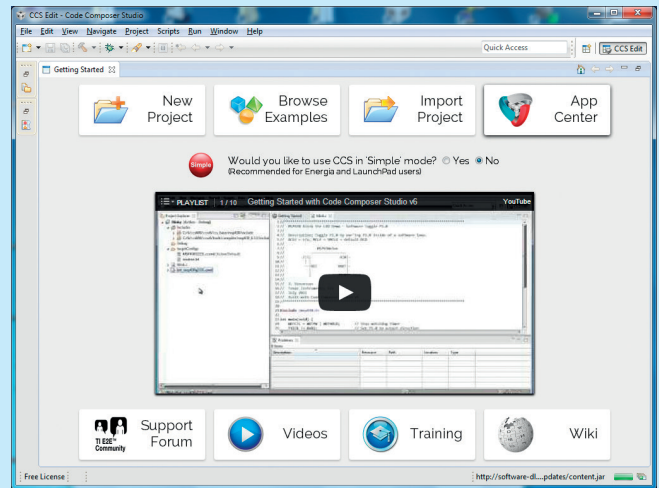
Należy koniecznie wyłączyć program antywirusowy na czas instalowania oraz aktualizacji środowiska. Plik *CCS6.0.1.00039_win32.zip* należy rozpakować. Następnie z foldera */CCS6.0.1.00039_win32* należy uruchomić program *ccs_setup_6.0.1.00039.exe*.

Środowisko CCSv6 należy zainstalować w proponowanej ścieżce *C:\ti*. Zmiana ścieżki może spowodować problemy z doinstalowaniem innych składników środowiska, np. bibliotek. Wydaje się, że należy zainstalować wersję pełną z proponowanymi domyślnymi opcjami (ok. 2.4 GB). Potem i tak można selektywnie doinstalować resztę składników. Na planszy *App Center* należy zaznaczyć wszystkie opcje. I tak pliki nie są pobierane przy instalowaniu CCSv6. Tworzone są tylko pozycje w oknie środowiska do późniejszego selektywnego doinstalowania.

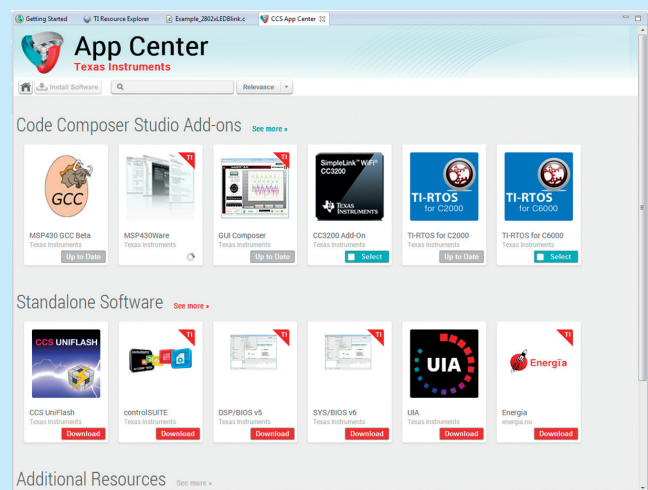
W systemie Windows 7 może w trakcie instalacji zachodzić konieczność ręcznego zezwolenia na wprowadzenie zmian w systemie. Instalowanie trwa dość długo, ok. pół godziny (na komputerze dwurdzeniowym 2.3 GHz) przy obciążeniu średnio 50%. Na końcu pracy pojawia się informacja, że po zakończeniu instalowania należy wykonać ponowne uruchomienie systemu operacyjnego Windows.

Po pierwszym uruchomieniu CCSv6.x wybierana jest automatycznie darmowa licencja o nazwie *FREE LICENCE*, która pracuje bez ograniczenia rozmiaru kodu z zestawami uruchomieniowymi TI z układem ewaluacyjnym oraz z emulatorem klasy XDS100 i mikroprocesorami TI wszystkich rodzin. Procesory rodziny MSP430 obsługuje z ograniczeniem rozmiaru kodu do 16 kB.

Po podaniu ścieżki i nazwy foldera roboczego w oknie *Workspce Launcher* wyświetlane jest okno aplikacji z oknem *Getting Started* (**rysunek 1**). Umożliwia ono bezpośredni dostęp do najbardziej potrzebnych operacji po uruchomieniu środowiska, jak tworzenie nowego projektu, wyszukiwanie projektów przykładowych lub obsługa dodatkowych aplikacji. Znajdują się tam również odnośniki do stron forum wsparcia TI, prezentacji wideo, materiałów treningowych oraz



Rysunek 1. Okno środowiska CCSv6 po pierwszym uruchomieniu



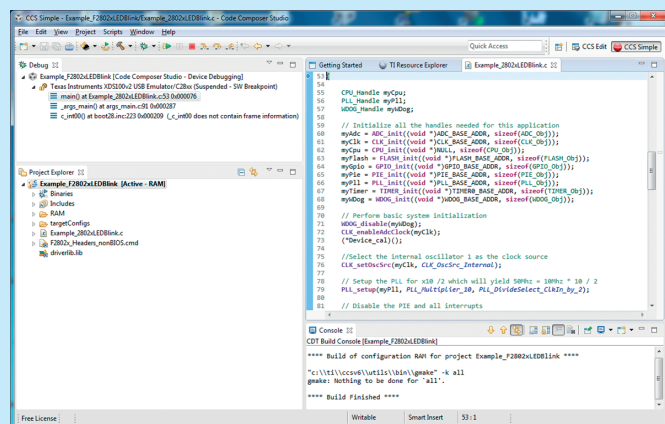
Rysunek 2. Okno App Center środowiska CCSv6

strony TI Wiki. W samym środku jest wyświetlane okno z prezentacją wideo (10:21), w której zostały pokazane podstawowe kroki użytkownika środowiska CCSv6. Jest to pierwsza prezentacja z dziesięciu.

Pokazane na **rysunku 2** kno *CCS App Center* jest nowym rozwiązaniem w CCS. Jest ono przeznaczone dla łatwego instalowania i zarządzania dodatkowym oprogramowaniem koniecznym przy pracy z wybraną rodziną procesorów, np. MSP430Ware, controlSUITE oraz TivaWare.

W oknie *CCS App Center* wyświetlane są odnośniki zgrupowane w trzech kategoriach:

- **Code Composer Studio Add-ons** – rozszerzenia środowiska CCSv6. Pokazywane jest tylko 6 najbardziej przydatnych. Reszta dodatków jest pokazywana po kliknięciu na odnośnik „See more”. CCSv6 automatycznie sprawdza wersje zainstalowane każdego dodatku oraz najnowsze wersje dostępne do pobrania. Jeśli wersja zainstalowana jest aktualna wyświetlany jest napis „Up to Date”.
- **Standalone Software** – pakiety programowe uzupełniające środowisko CCS ale ściśle z nim powiązane, np. controlSUITE lub Energia.
- **Additional Resources** – odnośniki do dodatkowych zasobów jak np. strona wsparcia modułów LaunchPad.



Rysunek 3. Okno perspektywy CCS Simple

Po wybraniu rozszerzenia rozpoczyna się pełna procedura aktualizacji środowiska CCS lub pobrania i zainstalowania kodu dodatkowej aplikacji. Typowo wymaga to ponownego uruchomienia CCS.

Przy uruchamianiu CCSv6 sprawdzana jest w sieci dostępność aktualizacji środowiska. Jeśli zostały wykryte nowe lub aktualniejsze komponenty to wyświetlane jest okno „Updates Available”. Po kliknięciu na link znajdujący się w tym oknie rozpoczyna się pełna procedura pobrania i zainstalowania nowego kodu.

Tryb Simple

W górnej części okna *Getting Started* (rys. 1) jest możliwość przełączania trybu pracy środowiska z pomiędzy trybem normalnym oraz trybem o nazwie „Simple mode”. W tym trybie jest używana tylko jedna perspektywa CCS Simple zawierająca wszystkie okna potrzebne do edycji i debugowania projektu. Została też zredukowana liczba przycisków, okien i pozycji menu. Ten tryb jest przeznaczony dla użytkowników rozpoczynających pracę z zestawem *LaunchPad* lub użytkowników środowiska *Energia*. *Energia* jest wersją środowiska *Arduino* przeznaczoną dla procesorów rodziny MSP430, Tiva i C2000. CCSv6 umożliwia importowanie projektów utworzonych w środowisku *Energia*.

Po przełączeniu na tryb *Simple* po lewej stronie okna *Getting Started* pojawiają się nowe okna *Debug* (ze standardowej perspektywy *CCS Debug*) oraz



Rysunek 4. Okno TI Resource Explorer z otwartą zakładką Home

Project Explorer (ze standardowej perspektywy *CCS Edit*). Reszta działania jest taka sama jak w przypadku perspektyw standardowych. Widok perspektywy *CCS Simple* po wykonaniu załadowania i zbudowania projektu jest pokazany na **rysunku 3**. Przyciśnięcie przycisku perspektywy *CCS Edit* powoduje tylko powiększenie okna edycyjnego z plikiem źródłowym do rozmiarów okna aplikacji.

Także przy pracy ze standardowymi perspektywami środowiska CCSv6 można dodatkowo utworzyć perspektywę *CCS Simple*. Może być ona przydatna dla osób przyzwyczajonych do pracy z aplikacją jednokienkową.

Instalacja i aktualizacja pakietu programowego controlSUITE

Na stronie *controlSUITE* dostępna jest aktualna wersja pakietu [19]. Obecnie najnowsza jest wersja *controlSUITE v3.2.7* (datowana 17 Jun 2014). Do pobrania pliku instalacyjnego jest potrzebna wcześniejsza rejestracja na stronie myTI [17]. Po kliknięciu na link pobierania przeprowadzane jest logowanie poprzez system myTI i udostępniany jest plik do pobrania. Jest on dość duży (obecnie 929MB) i jest przesyłany raczej powoli. Pakiet jest dość często aktualizowany. Pobrany plik *sprca85.zip* należy rozpakować. Następnie należy uruchomić program *controlSUITE3.2.7setup.exe*. Instalowanie tej wersji pakietu programowego *controlSUITE* przebiega poprawnie (2.13 GB).

Po przejściu do okna *CCS App Center* i przyciśnięciu na panelu *controlSUITE* przycisku *Download* otwierana jest tylko strona *controlSUITE* [19]. Pobranie i instalowanie pakietu trzeba wykonać samemu w sposób opisany powyżej.

Kliknięcie w oknie *Getting Started* przycisku *Browse Examples* powoduje otwarcie okna *TI Resource Explorer*, dobrze znanego z poprzedniej wersji CCS i o takim samym działaniu. Istotne informacje są zgrupowane na zakładce *Home*. Można ją otworzyć po kliknięciu w oknie *TI Resource Explorer* na ikonkę *Home*. Po lewej stronie okna pokazywane jest drzewo dokumentacji i dostępnych projektów przykładowych (**rysunek 4**).

Informacje o dostępnych projektach przykładowych jest przechowywana w bazie. Baza powinna być generowana automatycznie, lecz często tak nie jest. Wtedy albo nie jest pokazywana linia *controlSUITE* dostępu do przykładowych projektów, albo jest ona pokazywana, ale zawiera starą i nieaktualną informację, np. z instalacji poprzedniej wersji.

Jeśli po instalacji pokazywana jest tylko jedna linia *controlSUITE* to udostępnia ona tylko dokumentację pakietu w języku angielskim i chińskim. W takim przypadku trzeba samemu wykonać procedurę budowania bazy informacji o projektach przykładowych. W przypadku, gdy jest druga linia to i tak najlepiej ponownie wygenerować bazę.

1. Na dole strony *Home* należy kliknąć na odnośnik *Configure Resource Explorer*.
2. Jeśli w białym polu wyboru okna dialogowego *Package Configuration* jest pokazywana nazwa *controlSUITE* to należy na nią kliknąć a następnie należy kliknąć przycisk *Remove* oraz przycisk *OK*.

Okno jest zamykane i środowisko CCS usuwa niepoprawnie zbudowaną bazę informacji o projektach przykładowych.

3. Następnie na dole strony *Home* należy ponownie kliknąć na odnośnik *Configure Resource Explorer*.

4. Jeśli w białym polu wyboru okna dialogowego *Package Configuration* jest pusto to trzeba kliknąć na *Add*.

5. Następnie trzeba wskazać folder *C:\ti\controlSUITE* i kliknąć *OK*.

6. Nazwa *controlSUITE* pojawi się w oknie wyboru. Należy kliknąć na nią a następnie na *OK*.

Po dłuższej chwili pojawi się w drzewie okna *TI Resource Explorer* druga linia *controlSUITE* zawierająca pozycje: *development kits*, *device_support* oraz *libs*.

Problemy z importowaniem projektów przykładowych z pakietu controlSUITE

7. W oknie *Getting Started* kliknij na przycisk *Browse Examples*.

Pojawia się okno *TI Resource Explorer* z rozwiniętą pozycją *controlSUITE* rozpoczynającą się od pola *English*. Jest to prezentacja informacyjna i nie nadaje się do pracy z projektami przykładowymi.

8. Dla pracy z rodziną układów procesorowych *Piccolo F2802x* rozwinij w oknie *TI Resource Explorer* drugą pozycję *controlSUITE*. Następnie rozwinij w tym oknie drzewo *controlSUITE* → *device_support* → *f2802x* → *v220* → *f2802x_examples_drivers*. Potem kliknij na nazwę wybranego projektu *Example_F2802xLEDBlink*.

W prawym oknie zostanie wyświetlona instrukcja jak krok po kroku zbudować i uruchomić projekt (rysunek 5).

Krok1: Importowanie projektu Example_F2802xLEDBlink do CCSv6

Krok1 umożliwia zaimportowanie wybranego projektu do CCSv6.

9. W oknie *TI Resource Explorer* kliknij na odnośnik kroku 1.

Po wystartowaniu pobierania pokazują się dwa okna „*Operation Summary*” sygnalizujące problemy występujące podczas importowania projektu.

Pierwszy komunikat dotyczy odwołania do pakietu XDAIS:

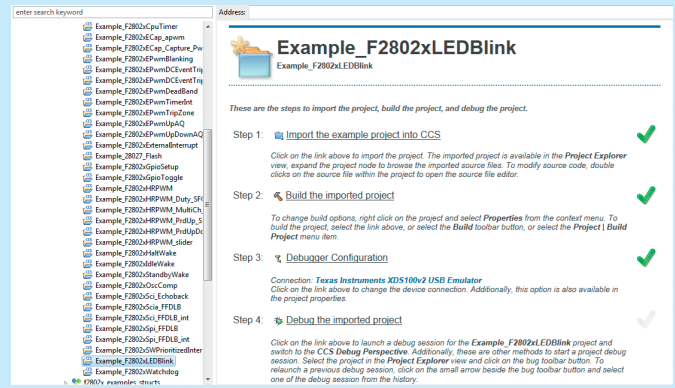
```
„Unrecognized product of type 'com.ti.rtsc.XDAIS' is required by project 'Example_F2802xLEDBlink' - please install at least one product of this type before building this project.”
```

Drugi komunikat dotyczy ustawionej zależności (dependency) do innego projektu:

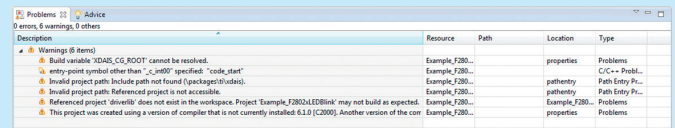
```
„Warning: Project 'driverlib', referenced from 'Example_F2802xLEDBlink', could not be found. Please import this project manually.”
```

Na razie należy kliknąć na przycisk *OK* w obu przypadkach.

Po zakończeniu wykonania importowania w oknie *Project Explorer* pojawia się drzewo projektu i w oknie *TI Resource Explorer* pokazywany jest zielony znaczek ✓ na prawo od linii nazwy kroku.



Rysunek 5. Okno *TI Resource Explorer* z otworzoną zakładką *Home*



Rysunek 6. Okno *Problems* po wykonaniu pierwszego budowania projektu

Projekt *Example_F2802xLEDBlink* został zaimportowany z kopiowaniem projektu i pliku *Example_2802xLEDBlink.c* do foldera roboczego projektu.

Krok2: Budowanie projektu Example_F2802xLEDBlink

Krok2 umożliwia wykonanie budowania wybranego projektu.

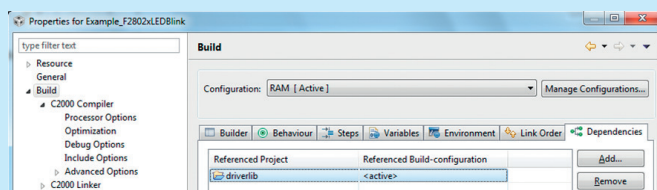
10. W oknie *TI Resource Explorer* kliknij na odnośnik kroku 2.

W oknie *Console* pokazywane są bieżące informacje o postępie budowania. W oknie *Problems* pokazywane są opisy błędów, ostrzeżeń i informacji. Po poprawnym wykonaniu budowania pokazywany jest w oknie *TI Resource Explorer* zielony znaczek ✓ na prawo od linii nazwy kroku.

Kliknięcie na odnośnik kroku 2 powoduje automatyczne budowanie projektu – podobnie jak po przyciśnięciu przycisku *Build*. Budowanie projektu *Example_F2802xLEDBlink* zostało zakończone poprawnie. Został utworzony wynikowy plik binarny *Example_2802xLEDBlink.out* (zobacz okno *Console*). Zostały jednak zgłoszone ostrzeżenia (zobacz okno *Problems*).

Ostrzeżeń jest sześć (rysunek 6):

1. entry-point symbol other than "_c_int00" specified: "code_start"
2. Referenced project 'driverlib' does not exist in the workspace. Project 'Example_F2802xLEDBlink' may not build as expected.
3. Build variable 'XDAIS.CG_ROOT' cannot be resolved.
4. This project was created using a version of compiler that is not currently installed: 6.1.0 [C2000]. Another version of the compiler will be used during build: 6.2.6. Please go to CCS App Center to install the compiler of the required version,



Rysunek 7. Okno ustawiania zależności projektu

or migrate the project to one of the available compiler versions by adjusting project properties.

5. Invalid project path: Include path not found (\packages\ti\xdais).
6. Invalid project path: Referenced project is not accessible.

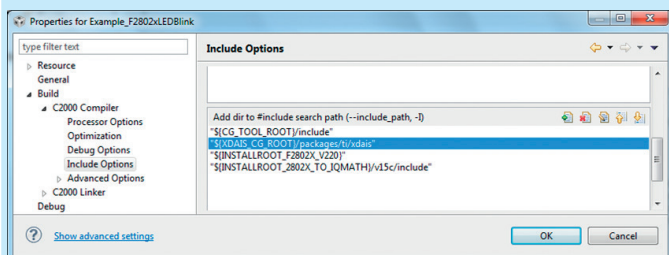
Ostrzeżenie nr 1 jest nieistotne. Celowo został zastosowany inny symbol adresowy (zmienna) punktu startowego kodu projektu niż standardowy. Dokładny opis jest zamieszczony w książkach [22, 23].

Ostrzeżenie nr 2 informuje, że do poprawnego zbudowania projektu *Example_F2802xLEDBlink* potrzebne może być zaimportowanie do foldera roboczego zależnego projektu *driverlib*.

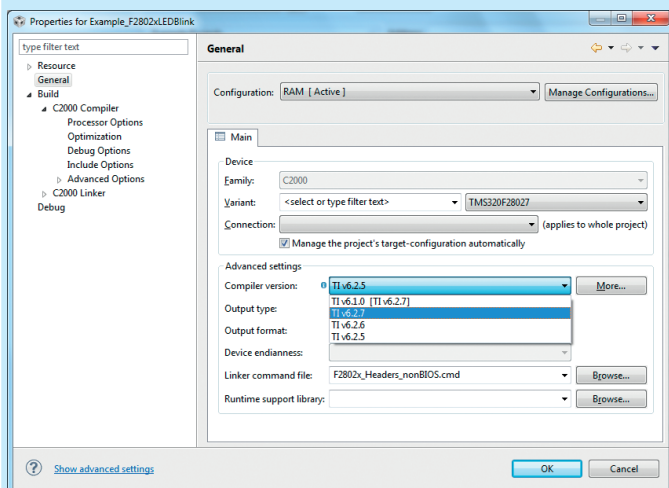
Typowo, projekt zależny jest biblioteką modyfikowaną podczas uruchamiania projektu aktualnego (głównego) w którym jest wykorzystywana. Gdy wykonywane jest budowanie projektu głównego, to najpierw budowany jest projekt zależny (np. biblioteka) a dopiero potem projekt główny (z aktualną wersją biblioteki).

W naszym przypadku nie jest potrzebne aktualizowanie biblioteki *driverlib*. Dlatego najlepszym sposobem rozwiązania problemu jest usunięcie zależności.

11. W oknie *Project Explorer* kliknij prawym klawiszem myszy na linię z nazwą projektu *Example_*



Rysunek 8. Okno ustawiania ścieżek dostępu dla kompilatora projektu



Rysunek 9. Okno ustawiania wersji kompilatora

F2802xLEDBlink. Z menu podręcznego wybierz *Properties*→*Build*→*Dependencies* (rysunek 7).

12. Zaznacz linię projektu *driverlib* i kliknij *Remove* a następnie *OK*.

Ostrzeżenie nr 3 informuje, że w projekcie zdefiniowano niepoprawną ścieżkę.

13. W oknie *Project Explorer* kliknij prawym klawiszem myszy na linię z nazwą projektu *Example_F2802xLEDBlink*. Z menu podręcznego wybierz *Properties*→*Include Options* (rysunek 8).

14. Zaznacz linię ze zmienną „XDAIS_CG_ROOT” i kliknij ikonkę *Delete* i następnie *OK*.

15. Takie postępowanie zapewni również usunięcie problemu sygnalizowanego w ostrzeżeniu nr 5.

Ostrzeżenie nr 4 informuje, że projekt został utworzony z użyciem wersji kompilatora, która nie jest w obecnej instalacji CCSv6 dostępna. Rozwiązaniem jest zastosowanie najnowszej wersji kompilatora.

16. W oknie *Project Explorer* kliknij prawym klawiszem myszy na linię z nazwą projektu *Example_F2802xLEDBlink*. Z menu podręcznego wybierz *Properties*→*General* (rysunek 9).

17. Rozwiń listę *Compiler version* i wybierz najnowszą wersję a następnie kliknij *OK*.

Ostrzeżenie nr 6 informuje o błędnej ścieżce do projektu zależnego. Ponieważ w punktach 5-9 usunęliśmy już tę niepotrzebną zależność to pora na sprawdzenie skuteczności tego działania.

18. Wykonaj samo budowanie projektu (bez ponownego startowanie sesji debugowej). W perspektywie *CCS Edit* kliknij na przycisk *Build*. Nie używaj przycisku *Debug*.

W wyniku ponownego budowania projektu pozostały dwa ostrzeżenia: dotychczasowe ostrzeżenie nr1 (dalej nieistotne) oraz dalej dotychczasowe ostrzeżenie nr6. Jego przyczyna jest trochę bardziej skomplikowana.

Sprawdź w oknie *Project Explorer*, że jako aktywna ustawiona jest konfiguracja budowania o nazwie *RAM*. W projekcie została zdefiniowana również druga konfiguracja budowania o nazwie *Flash*. Ona również zawiera własny komplet ustawień.

19. W oknie *Project Explorer* kliknij prawym klawiszem myszy na linię z nazwą projektu *Example_F2802xLEDBlink*. Z menu podręcznego wybierz *Properties*→*Build*→*Dependencies*.

20. Rozwiń listę *Configuration* i wybierz *Flash*.

21. Zaznacz linię projektu *driverlib* i kliknij *Remove* a następnie *OK*.

22. Wykonaj ponownie samo budowanie projektu. Tym razem pozostało tylko jedno, nieistotne ostrzeżenie.

Czy oznacza to, że projekt *Example_F2802xLEDBlink* jest wolny od błędów kompilacji? Otóż nie!

Kliknij na trójkącik na prawo od przycisku *Build*. Wybierz pozycję *Flash*. Zostanie wykonane budowanie z konfiguracją budowania o nazwie *Flash*. W oknie *Problems* pojawiły się 4 ostrzeżenia o znanych opisach. Należy usunąć resztę błędów również w i tej konfiguracji budowania.

23. W oknie *Project Explorer* kliknij prawym klawiszem myszy na linię z nazwą projektu *Example_F2802xLEDBlink*. Z menu podręcznego wybierz *Properties*→*General*.

24. Rozwiń listę *Compiler version* i wybierz najnowszą wersję a następnie kliknij *OK*.

25. W oknie *Project Explorer* kliknij prawym klawiszem myszy na linię z nazwą projektu *Example_F2802xLEDBlink*. Z menu podręcznego wybierz *Properties*→*Include Options*.

26. Zaznacz linię ze zmienną „*XDAIS_CG_ROOT*” i kliknij ikonkę *Delete* a następnie *OK*.

27. Wykonaj ponownie samo budowanie projektu. Tym razem pozostało tylko jedno, nieistotne ostrzeżenie.

W oknie *Project Explorer* jest pokazane, że wybrana i aktywna jest konfiguracja budowania *Flash*. Należy to zmienić.

28. W oknie *Project Explorer* kliknij prawym klawiszem myszy na linię z nazwą projektu *Example_F2802xLEDBlink*. Z menu podręcznego wybierz *Build Configurations*→*Set Active*→*RAM*.

Krok3: Definiowanie konfiguracji sprzętowego systemu docelowego

Krok3 umożliwia zdefiniowanie konfiguracji sprzętowej systemu docelowego dla projektu. Na początku pole *Connection* pokazuje typ „none”.

29. W oknie *TI Resource Explorer* kliknij na odnośnik kroku 3.

W oknie dialogowym *Debugger Configuration* rozwiń listę wyboru.

30. Wybierz pozycję *Texas Instruments XDS100v2 USB Emulator*. Kliknij *OK*.

W oknie *TI Resource Explorer* pole *Connection* pokazuje teraz typ *Texas Instruments XDS100v2 USB Emulator*. Zielony znaczek ✓ pokazywany jest na prawo od linii nazwy kroku.

Krok4: Uruchamianie sesji debugowej dla projektu Example_F2802xLEDBlink

Krok4 umożliwia uruchomienie sesji debugowej dla projektu. Dotychczas praca środowiska CCSv6 nie wymagała fizycznej obecności sprzętu docelowego. Wykonanie kroku 4 wymaga wcześniejszego dołączenia zestawu ewaluacyjnego *C2000 Piccolo LaunchPad* do komputera z zainstalowanym środowiskiem CCSv6 [1].

31. W oknie *TI Resource Explorer* kliknij na odnośnik kroku 4.

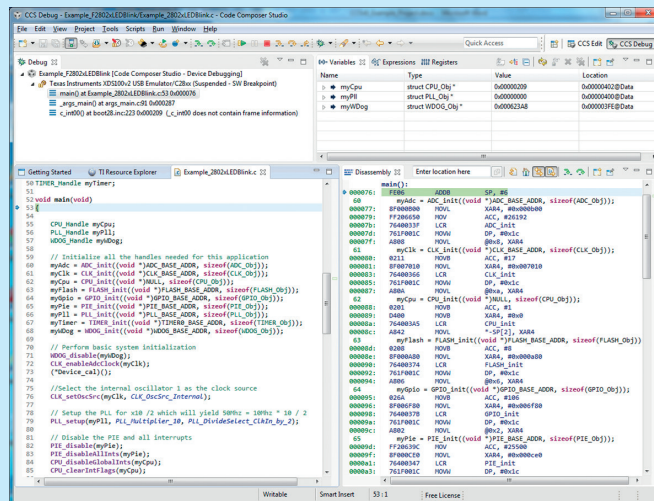
Kliknięcie na odnośnik kroku 4 powoduje automatyczne rozpoczęcie sesji debugowej – podobnie jak po przyciśnięciu przycisku *Debug*.

Postęp działania środowiska CCSv6 można obserwować na pasku stanu w prawym dolnym rogu okna. Może to trwać dosyć długo i należy koniecznie poczekać przed rozpoczęciem dalszej pracy na zakończenie ładowania kodu i pokazanie się okna perspektywy *CCS Debug*.

Wgląd w projekt Example_F2802xLEDBlink

32. W perspektywie *CCS Debug* zauważ w oknie edytora, że praca programu została zatrzymana na pierwszej linii kodu funkcji *main()*.

33. Otwórz okno *Disassembly* z menu *View*→*Disassembly*. W tym oknie można dokładnie



Rysunek 10. Okno perspektywy *CCS Debug*

zobaczyć jak naprawdę pracuje układ procesorowy Piccolo F28027 (rysunek 10).

Perspektywa *CCS Debug* nie uległa wielkiej zmianie. Jedyne ikonki debugowania (jak *Suspend*, *Resume* itd.) zostały przeniesione z paska okna *Debug* na główny pasek perspektywy.

Polecenie *Search* działa teraz globalnie (np. na całym folderze roboczym) a nie tylko w ramach jednego pliku.

Na pasku narzędziowym pojawiła się ikonka *Open Perspective* pozwalająca na swobodne otwieranie okien innych perspektyw, np. *CCS Simple*.

Działanie reszty środowiska CCSv6 wydaje się bardzo podobne jak w wersji CCSv5.5.

Problemy w pracy z biblioteką driverlib

W pracy z pakietem programowym *controlSUITE* występują też różne problemy. Niektóre z nich zostaną poniżej omówione.

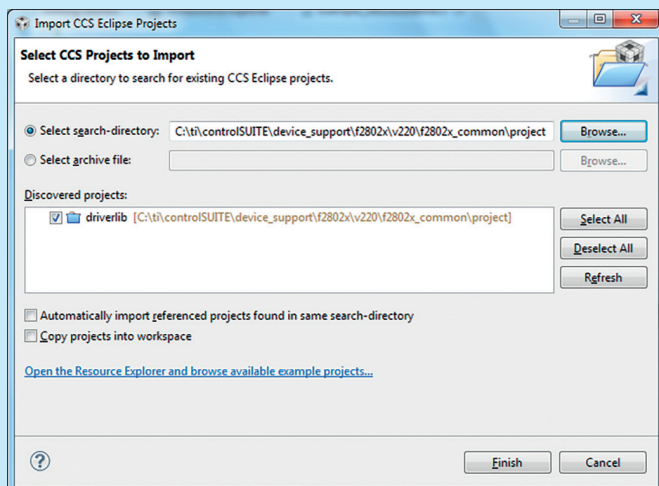
Pliki dotyczące biblioteki *driverlib* znajdują się w standardowej ścieżce *C:\TI\controlSUITE\device_support\f2802x\v220\f2802x_common*

Plik *driverlib.lib* znajduje się w folderze *\lib* tej ścieżki. Pliki źródłowe znajdują się w folderze *\source* tej ścieżki. Projekt CCS znajduje się w folderze *\project* tej ścieżki.

Debugowanie projektów z użyciem biblioteki *driverlib* powoduje kłopoty. Przy próbie wyświetlenia kodu źródłowego funkcji z biblioteki *driverlib* wyświetlany jest komunikat o braku dostępu do plików źródłowych. Problem jest spowodowany wygenerowaniem biblioteki *driverlib* w lokalizacji innej niż standardowa ścieżka instalowania pakietu *controlSUITE*.

Dla układów procesorowych rodziny Piccolo F2802x pakiet „firmware” z biblioteką *driverlib* występuje w dwóch lokalizacjach:

- W standardowej lokalizacji pakietów dla poszczególnych serii układów procesorowych rodziny C2000. Dla serii Piccolo F2802x jest to ścieżka do ostatniej wersji pakietu *C:\TI\controlSUITE(device_support\f2802x\v220)\f2802x_common*
- W lokalizacji dla zestawu ewaluacyjnego *C2000 Piccolo LaunchPad* jest to ścieżka *C:*



Rysunek 11. Okno importowania projektu biblioteki *driverlib*

TI\controlSUITE\development_kits\C2000_LaunchPad\f2802x_common

Rozwiązaniem problemu jest ponowne zbudowanie biblioteki z poprawnymi ścieżkami. Wymaga to importowania projektu biblioteki do środowiska CCSv6 (bez kopiowania) i ponownego wykonania jego budowania. Wygenerowanie nowego pliku *driverlib.lib* zapewnia poprawne debugowania wszystkich projektów z pakietem narzędziowym *f2802x\v220*. Umożliwia też wykonanie budowania z zastosowaniem nowszej wersji pakietu generacji kodu (w tym kompilatora). Może to być korzystne a nawet konieczne przy dużych zmianach wersji programów narzędziowych.

Importowanie i budowanie projektu *driverlib* pakietu **F2802x**

34. W perspektywie *CCS Edit* z menu *Project* wybierz pozycję *Import Existing CCS Eclipse Project*.

35. W otworzonym oknie kliknij na przycisk *Browse*, wskaż odpowiednią ścieżkę:

- Dla lokalizacji projektu w ścieżce `\device_support` jest to ścieżka `C:\ti\controlSUITE\device_support\f2802x\v220\f2802x_common\project`
- Dla lokalizacji projektu w ścieżce `\development_kits\C2000_LaunchPad` jest to ścieżka `C:\ti\controlSUITE\development_kits\C2000_LaunchPad\f2802x_common\project`

36. Kliknij *OK*.

37. W nowym oknie (**rysunek 11**) zauważ, że **NIE** jest wybrana opcja *Copy projects into workspace*. Oznacza to, że pliki źródłowe i wynikowe projektu pozostaną w lokalizacji początkowej. Kliknij na przycisk *Finish*.

Po poprawnym wykonaniu importowania w oknie *Project Explorer* pojawia się drzewo projektu *driverlib*.
38. W oknie *Project Explorer* rozwiń drzewo projektu i kliknij na jego nazwę. Konfiguracja budowania projektu o nazwie *Release* została ustawiona jako aktywna.

39. Kliknij prawym klawiszem myszy na linię nazwy projektu. Z podręcznego menu wybierz *Properties*.

40. W oknie *Properties for driverlib* wybierz gałąź *General*.

41. Z listy rozwijanej *Compiler version* wybierz najnowszą wersję kompilatora. Kliknij *OK*.

42. Wykonaj samo budowanie projektu bez startowanie sesji debugowej.

Budowanie projektu *driverlib* zostało zakończone poprawnie. Został utworzony plik wynikowy *driverlib.lib* w ścieżce `C:\TI\controlSUITE\device_support\f2802x\v220\f2802x_common\lib` (zobacz okno *Console*). Nie zostały zgłoszone błędy lecz jedno ostrzeżenie (zobacz okno *Problems*).

Zauważ, że plik wynikowy *driverlib.lib* został wpisany do innego foldera niż folder projektu `\f2802x_common\project`. Oznacza to, że przy budowaniu projektu *driverlib* w konfiguracji *Debug* wyprodukowany plik wynikowy *driverlib.lib* został nadpisany

na stary plik w folderze `\f2802x_common\lib`.

Importowanie i budowanie projektu *driverlib* dla zestawu ewaluacyjnego

Należy ponownie wygenerować plik *driverlib.lib* w lokalizacji dla zestawu ewaluacyjnego *C2000 Piccolo LaunchPad*.

43. Usuń projekt *driverlib* z projektu.

44. W oknie *Project Explorer* kliknij prawym klawiszem myszy na linię z nazwą projektu *driverlib*. Z menu podręcznego wybierz *Delete*. Kliknij *OK*.

Dalsze działanie przebiega w taki sam sposób jak to zostało opisane powyżej.

Z biblioteką *driverlib* zastosowaną do obsługi zestawu ewaluacyjnego *C2000 Piccolo LaunchPad* jest też dodatkowy problem. Plik biblioteki i pliki źródłowe są ze starej wersji `/F2802x/v210` a pliki dokumentacji z jeszcze starszej wersji `/F2802x/v200`.

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

Bibliografia

[15] *Code Composer Studio*, strona produktu

<http://goo.gl/PaADwr>

[16] *Download CCS*, strona pobierania

<http://goo.gl/0n3wWN>

[17] *my.TI Account*, strona rejestracji Texas

Instruments <http://goo.gl/rtpMqC>

[18] *TI Wiki:Code Composer Studio v6*

<http://goo.gl/Ex1978>

[19] *controlSUITE*, strona pobierania <http://goo.gl/Jdfz00>

[20] *F2802x Firmware Development Package USER'S*

GUIDE v. 210 [*f2802x-FRM-EX-UG.pdf*], pakiet

controlSUITE

[21] *F2802x Peripheral Driver Library USER'S GUIDE*

v. 210 [*f2802x-DRL-UG.pdf*], pakiet *controlSUITE*

[22] *Henryk A. Kowalski, Procesory DSP dla*

praktyków, BTC, Warszawa, 2011 <http://goo.gl/j14jns>

[23] *Henryk A. Kowalski, Procesory DSP*

w przykładach, BTC, Warszawa, 2012 <http://goo.gl/j14jns>