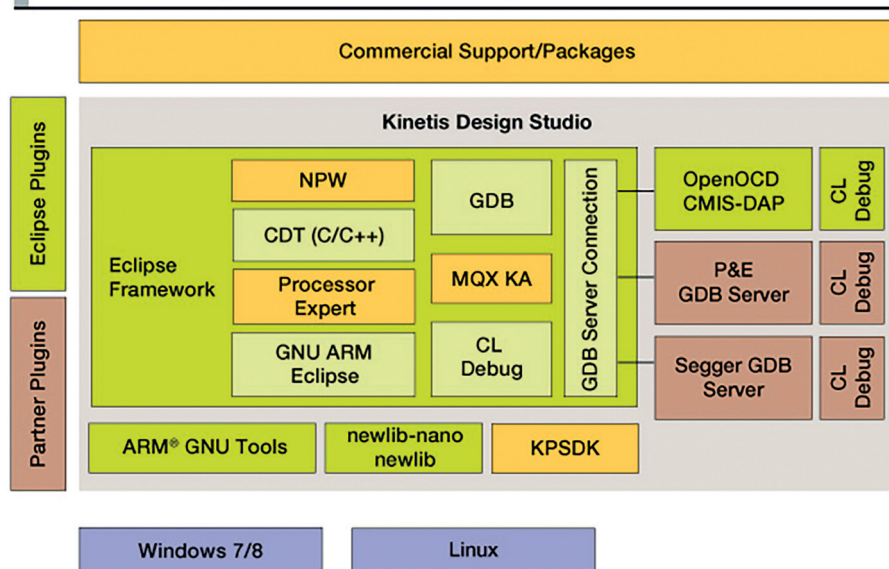


# Kinetis Design Studio (1)

## Nowe, bezpłatne środowisko programistyczne

Kilka tygodni temu firma Freescale udostępniła pierwsze, jeszcze testowe, wersje nowego środowiska programistycznego o nazwie Kinetis Design Studio, które docelowo ma zastąpić – jako narzędzie dla mikrokontrolerów Kinetis – cieszącego się dużą popularnością Code Warriora. W artykule przedstawiamy cechy i budowę tego pakietu, narzędzia udostępniane przez środowisko programistyczne, pokażemy także przykładową aplikację zrealizowaną pod jego „opieką”.

### Kinetis Design Studio Block Diagram



Rysunek 1. Schemat budowy środowiska Kinetis Design Studio

Podstawę Kinetis Design Studio stanowi Eclipse (wersja Kepler 4.3). Jest to napisana w języku Java otwarta (open-source) platforma, w oparciu o którą można tworzyć środowiska programistyczne. Eclipse samo w sobie nie dostarcza żadnych narzędzi programistycznych, oferuje jednak obsługę tak zwanych wtyczek (*plug-ins*). Wtyczki to narzędzia, które mogą być integrowane z Eclipse.

### Elementy składowe Kinetis Design Studio

Chcąc pozostać w zgodzie z kolejnością czynności procesu tworzenia oprogramowania, jako pierwsze z narzędzi Kinetis Design Studio należy wymienić Processor Experta. Jest to generator kodu konfiguracyjnego dla mikrokontrolera. Narzędzie to pozwala za pomocą graficznego interfejsu użytkownika wygenerować pliki z kodem źródłowym, w których zaimplementowany jest interfejs programistyczny (API – *Application*

*Programming Interface*) do zasobów wewnętrznych mikrokontrolera.

Kolejnym używanym przez programistę narzędziem Kinetis Design Studio jest edytor. Jak łatwo jest się domyślić pozwala on na pisanie kodu źródłowego aplikacji. Edytor Kinetis Design Studio pozwala pisać kod źródłowy w języku Assembler, C oraz C++.

Kod źródłowy nie jest interpretowalny dla układów elektronicznych, które operują na wartościach binarnych, zatem potrzebne jest narzędzie, które odpowiednio przetworzy kod z jednej postaci do drugiej. Tym narzędziem jest kompilator. Tłumaczy on pliki zawierające kod źródłowy na pliki z kodem maszynowym. W przypadku Kinetis Design Studio zastosowano kompilator GCC (GNU Compiler Collection).

Jeśli podczas kompilacji błędy nie wystąpią, dalej pliki z kodem maszynowym są konsolidowane w jeden plik wykonywalny. Za realizację tego zadania odpowiada linker. Plik wykonywalny w zapisywany jest do pa-

mieci mikrokontrolera. Za poprawne wykonanie tej czynności odpowiada programator.

Ostatnim z narzędzi jest debugger. Debugowanie, tłumaczone czasem jako odrobaczanie lub odpluskowanie, to proces systematycznego redukcji liczby błędów w oprogramowaniu. Zasada działania debugera jest prosta. Pozwala on wykonywać kod stopniowo, a nawet krok po kroku, jednocześnie pozwalając na podgląd stanu systemu, w tym np. rejestrów mikrokontrolera i zmiennych. Dzięki temu programista może szczegółowo analizować zachowanie aplikacji, co bardzo pomaga w wychwyceniu różnego rodzaju nieprawidłowości, a w konsekwencji ich usunięciu. Kinetis Design Studio wykorzystuje debugger GDB (GNU Project debugger). Obsługiwane przez debugger interfejsy sprzętowe to: Segger J-Link, P&E Multilink oraz CMSIS-DAP.

Ponadto elementem, który można doinstalować w formie wtyczki do środowiska Kinetis Design Studio jest system operacyjny czasu rzeczywistego (RTOS – Real Time Operating System). Może to być produkt wspierany od dawna przez firmę Freescale o nazwie MQX lub popularnie stosowany w różnych systemach wbudowanych FreeRTOS.

Ostatnim nieomówionym dotąd komponentem składającym się na Kinetis Design Studio jest newlib/newlib-nano. Jest to biblioteka implementująca standard języka C.

Pakiet Kinetis Design Studio został dostosowany do pracy na dwóch obecnie najpopularniejszych systemach operacyjnych: Windows (wersje 7 i 8, zarówno 32- jak też 64-bitowe) oraz Linux (Ubuntu, Redhat, Centos).

Schemat blokowy przedstawiający budowę Kinetis Design Studio przedstawiono na **rysunku 1**.

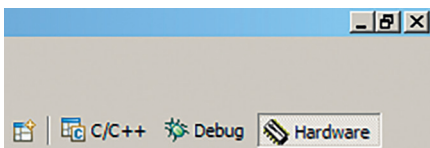
### Proces tworzenia aplikacji

Proces tworzenia aplikacji w programie Kinetis Design Studio można zasadniczo podzielić na trzy etapy. W pierwszym etapie programista korzystając z narzędzia Processor Expert tworzy interfejs programistyczny do peryferiów mikrokontrolera, których użycia wymaga aplikacja. W drugim etapie programista pisze kod źródłowy aplikacji w oparciu o wcześniej wygenerowany interfejs do peryferiów. W trzecim etapie programista sprawdza poprawność działania aplikacji poprzez uruchomienie jej i debugowanie na platformie sprzętowej.

## Pobranie i instalacja Kinetis Design Studio

Na portalu internetowym firmy Freescale ([www.freescale.com](http://www.freescale.com)) znajduje się strona poświęcona środowisku programistycznemu Kinetis Design Studio. Można do niej przejść korzystając z menu portalu, kolejno: Software and Tools → Software Development Tools → Kinetis Software and Tools → Kinetis Design Studio. Jako że strona ta jest umiejscowiona dosyć głęboko w hierarchii portalu, alternatywnym sposobem zlokalizowania jej w Internecie jest wpisanie hasła „Kinetis Design Studio” w wyszukiwarce Google, co z pewnością pozwoli łatwo znaleźć szukany adres www. Na stronie dedykowanej pakietowi Kinetis Design Studio dostępna jest dokumentacja i plik instalacyjny. W celu pobrania pliku instalacyjnego należy kliknąć zakładkę Downloads, w wyniku czego wyświetlona zostanie lista dostępnych wersji Kinetis Design Studio (każda z nich jest przeznaczona na inny system operacyjny). Po wybraniu odpowiedniej wersji rozpocznie się pobieranie pliku instalacyjnego na dysk twardej komputera.

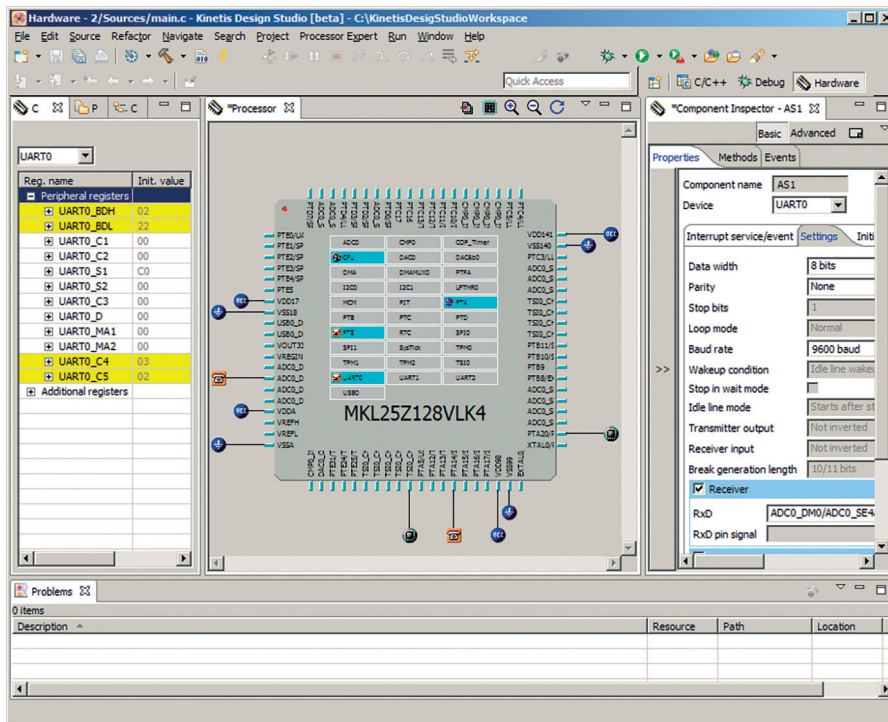
Po uruchomieniu pobranego pliku włączona zostanie aplikacja, która w kilku krokach przeprowadzi użytkownika przez proces instalacji Kinetis Design Studio. W pierwszym kroku wyświetlony zostanie komunikat powitalny, informujący o włączeniu kreatora instalacji środowiska programistycznego. Po naciśnięciu przycisku Next użytkownik poproszony zostanie o wybranie jednego z trzech rodzajów instalacji. Do wyboru są: Typical, Custom oraz Complete. Wybierając opcję Complete zainstalowane zostaną wszystkie możliwe elementy, których jest sześć: GNU Tools for Kinetis, Eclipse IDE, OpenOCD Debug Support, Segger Debug Support, P&E Debug Support oraz Documentation. Opcja Typical to instalacja tylko podstawowych składników. Z kolei decydując się na opcję Custom użytkownik ma możliwość wyboru komponentów, które zostaną zainstalowane, jak też możliwa jest zmiana domyślnej ścieżki wskazującej miejsce instalacji na dysku twardym komputera. Po wyborze jednej z wymienionych opcji rozpocznie się instalacja Kinetis Design Studio. O postępie instalacji informować będzie pasek stanu.



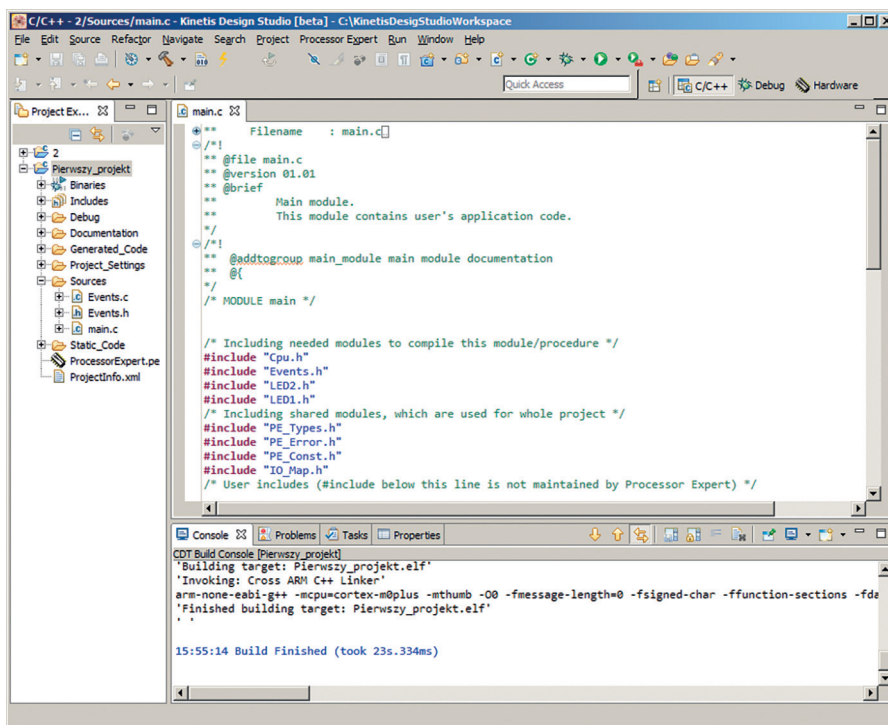
Rysunek 2. Panel zmiany perspektywy w środowisku Kinetis Design Studio

W tym miejscu należy wprowadzić pojęcie perspektywy. Perspektywa to mechanizm Eclipse określający jakie okna narzędziowe są w danej chwili widoczne. Można zatem powiedzieć, że perspektywa dostosowuje środowisko programistyczne do rodzaju czynności wykonywanej przez programistę. Środowisko Kinetis Design Studio udostępnia osobną perspektywę dla każdego z wcześniej opisanych etapów tworzenia aplikacji. Są to perspektywy Hardware, C/C++ oraz Debug odpowiednio dla pierwszego, drugiego i trzeciego etapu tworzenia aplikacji. Wybranie perspektywy odbywa się przez naciśnięcie na odpowiadającą jej ikonę w panelu wyboru perspektywy, który znajduje się w prawym górnym rogu okna programu Kinetis Design Studio (rysunek 2). W dalszej części artykułu opisano każdą z tych perspektyw.

Zacznijmy od perspektywy Hardware (rysunek 3). Dzięki niej programista może korzystać z Processor Expert i tworzyć przy jego pomocy interfejs programistyczny dla



Rysunek 3. Widok perspektywy Hardware środowiska Kinetis Design Studio



Rysunek 4. Widok perspektywy C/C++ środowiska Kinetis Design Studio

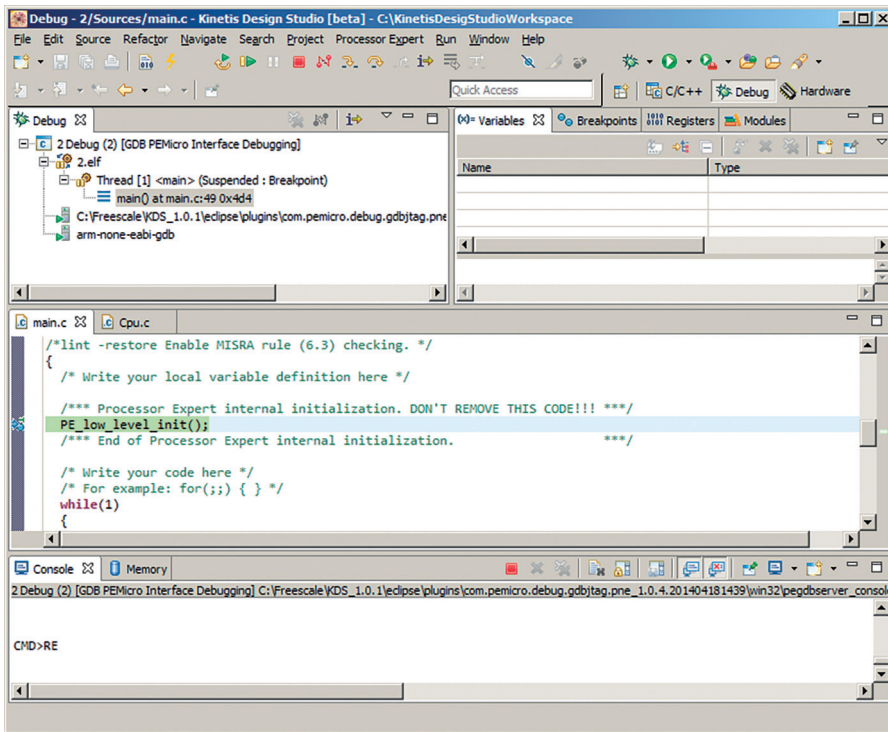
mikrokontrolera. Perspektywa Hardware udostępnia następujące okna:

- okno Configuration Registers służące do podglądu zawartości rejestrów mikrokontrolera na etapie inicjalizacji jego peryferiów,
- okno Processor przedstawiające w sposób graficzny model mikrokontrolera (lista peryferiów oraz układ wprowadzeń),
- okno Component Inspector pozwalające skonfigurować peryferia i dedykowany im interfejs programistyczny,

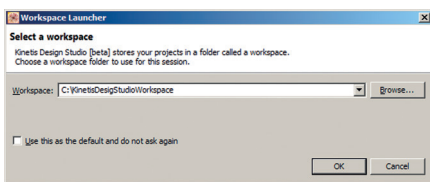
- okno Problems z listą błędów i ostrzeżeń.

Gdy interfejs programistyczny został dodany do projektu, programista zmienia perspektywę na C/C++ (rysunek 4). Służy ona do pisania kodu źródłowego. Perspektywa C/C++ udostępnia następujące okna:

- okno Project Explorer do zarządzania projektami programistycznymi i plikami projektów,
- okno edytora do edytowania plików kodu źródłowego,



Rysunek 5. Widok perspektywy Debug środowiska Kinetis Design Studio



Rysunek 6. Wybór ścieżki na dysku do przestrzeni roboczej z projektami programistycznymi pakietu Kinetis Design Studio

- okno Problems z listą błędów i ostrzeżeń powstałych na skutek kompilowania i linkowania aplikacji,
- okno Console, w którym wyświetlane są logi (np. informacje dotyczące procesu kompilacji i budowania aplikacji).

Po częściowym lub całkowitym ukończeniu pracy nad pisaniem kodu źródłowego programista może uruchomić aplikację na

platformie sprzętowej w celu przeanalizowania poprawności jej działania. Do tego celu służy trzecia perspektywa – Debug (rysunek 5). Udostępnia ona następujące okna:

- okno Debug służące do kontrolowania procesu debugowania,
- okno edytora do śledzenia na jakim etapie wykonywania kodu źródłowego jest mikrokontroler,
- okno Problems z listą błędów i ostrzeżeń powstałych na skutek kompilowania i linkowania aplikacji,
- okno Console, w którym wyświetlane są logi (np. informacje dotyczące procesu kompilacji i budowania aplikacji),
- okno Memory z podglądem zawartości pamięci mikrokontrolera,
- okno Variables do podglądu wartości typów danych używanych w kodzie źródłowym,

- okno Breakpoints z listą pułapek (wybranych przez programistę miejsc w kodzie, na których wykonywanie aplikacji ma zostać wstrzymane),
- okno Registers do podglądu wartości rejestrów mikrokontrolera,
- okno Modules z listą składających się na projekt programistyczny modułów.

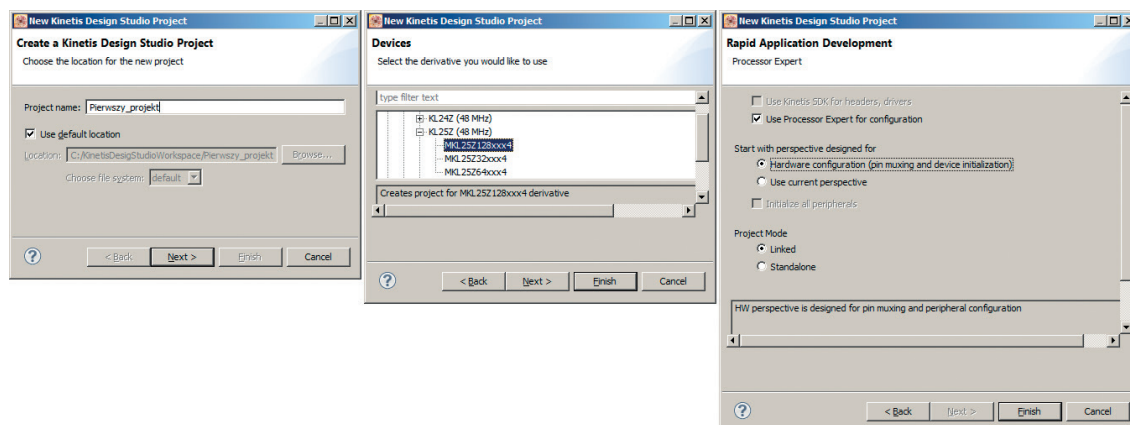
## Uruchomienie Kinetis Design Studio i stworzenie nowego projektu

Po zakończeniu instalacji Kinetis Design Studio można od przystąpić do pracy z tym narzędziem. Po uruchomieniu programu Kinetis Design Studio użytkownik zostanie poproszony o wskazanie ścieżki do przestrzeni roboczej z projektami programistycznymi, tak zwanego workspace'a (rysunek 6). Jest to adres miejsca na dysku twardym komputera, do którego Kinetis Design Studio odwoła się i wczyta wszystkie obecne tam projekty programistyczne, ponadto nowo-stworzone projekty również będą w tej lokalizacji zapisywane. Po zakończeniu wczytywania projektów (o ile pod wskazanym adresem są jakieś projekty) środowisko Kinetis Design Studio jest gotowe do pracy.

Aby stworzyć nowy projekt programistyczny w Kinetis Design Studio, należy skorzystać z menu głównego programu i wybrać File → New → Kinetis Design Studio Project. Spowoduje to otwarcie kreatora nowego projektu, który w kilku krokach przeprowadzi użytkownika przez proces tworzenia projektu.

W pierwszym kroku należy wpisać nazwę projektu (przykładowo może to być „Pierwszy\_projekt”) i ścieżkę na dysku twardym, gdzie ma on zostać zapisany (domyślną ścieżką jest adres przestrzeni roboczej z projektami programistycznymi) (rysunek 7a).

W drugim kroku należy wybrać model mikrokontrolera, dla którego projekt jest tworzony. Jako że wykorzystywaną platformą sprzętową jest FREEDOM FRDM-



Rysunek 7. Od lewej: a) Uzupełnienie nazwy projektu, b) Wybranie modelu mikrokontrolera, c) aktywowanie narzędzia Processor Expert dla projektu

KL25Z, to właśnie model z tej płytki (układ MKL25Z128) został wybrany z listy dostępnych mikrokontrolerów (rysunek 7b).

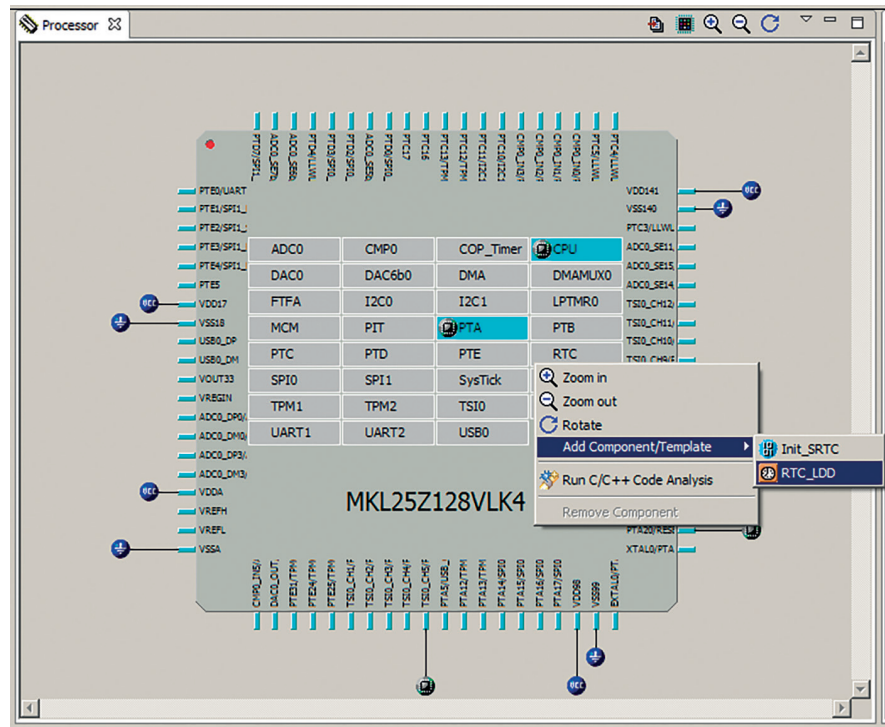
W trzecim kroku użytkownik decyduje czy podczas tworzenia oprogramowania w projekcie będzie posługiwał się Processor Expertem (rysunek 6c). Należy zaznaczyć tą opcję. Jest to już ostatnie okno kreatora projektu. W celu zakończenia jego pracy użytkownik naciska przycisk „Finish”. Kinetis Design Studio stworzy nowy projekt i automatycznie wczyta go do przestrzeni roboczej.

Dodanie do projektu programistycznego interfejsu do peryferiów mikrokontrolera

Pierwszym etapem tworzenia aplikacji jest dodanie do projektu programistycznego interfejsu do peryferiów mikrokontrolera. Programista realizuje to zadanie przy wykorzystaniu narzędzia Processor Expert. Dostęp do tego narzędzia jest możliwy po zmianie perspektywy programu CodeWarrior na Hardware.

W pierwszej kolejności programista wybiera z których peryferiów mikrokontrolera chce korzystać i jakie komponenty oprogramowania będą przeznaczone do ich sterowania. Wskazanie pojedynczego poryferium i dedykowanego mu komponentu oprogramowania odbywa się przez najechanie kursorem myszy na symbol wybranego peryferium w oknie Processor, wywołanie menu kontekstowego przez naciśnięcie prawego przycisku myszy, wybranie z tego menu opcji Add Component/Template, a następnie wybranie z listy jednego z dostępnych komponentów. Przykład wybrania komponentu oprogramowania odpowiadającego peryferium mikrokontrolera o nazwie RTC przedstawiono na rysunku 8.

Gdy komponenty oprogramowania są już wybrane, należy je skonfigurować. Aby wykonać konfigurację pojedynczego komponentu należy go w pierwszej kolejności zaznaczyć w oknie Processor. Opcje konfiguracji komponentu zostaną wyświetlone w oknie Component Inspector. Okno to składa się z co najmniej trzech zakładek (ich całkowita liczba jest różna w zależności od komponentu). Pierwsza zakładka, o nazwie Properties, udostępnia listę parametrów pracy peryferium, których wartości



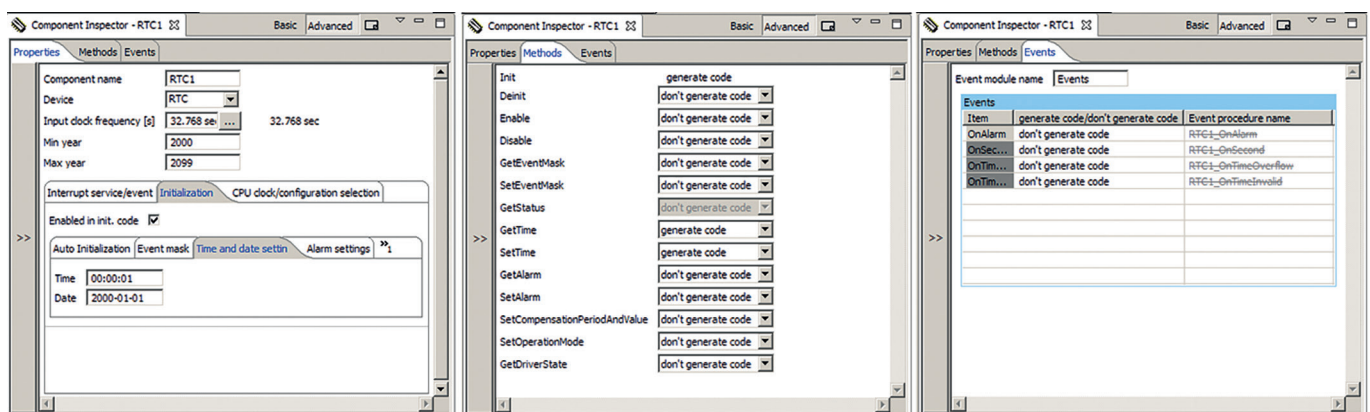
Rysunek 8. Przykład dodania do projektu programistycznego komponentu oprogramowania do konfiguracji i sterowania zegarem czasu rzeczywistego mikrokontrolera

programista może modyfikować. Przykładowo dla peryferium RTC najważniejszymi parametrami są: źródło sygnału zegarowego dla RTC, początkowa godzina/data oraz automatyczna inicjalizacja peryferium podczas rozpoczęcia wykonywania aplikacji (rysunek 9a). Druga zakładka (o nazwie Methods) udostępnia listę funkcji odpowiadających danemu peryferium (przykładowo dostępne dla peryferium RTC funkcje zaprezentowano na rysunku 9b). Programista może zdecydować, które z funkcji będą potrzebne i zaznaczyć, aby zostały włączone do plików, które zostaną dodane do projektu podczas późniejszego procesu generowania kodu źródłowego przez Processor Expert. Trzecia zakładka Events również udostępnia listę funkcji dla danego peryferium. Funkcje te to tak zwane zdarzenia (funkcje, które zostaną wykonane, jeśli określony warunek zostanie spełniony). Jako przykład dostępne funkcje tego

typu dla peryferium RTC przedstawiono na rysunku 8c. Analogicznie do zakładki Methods, programista w zakładce Events wybiera które z funkcji-zdarzeń zostaną włączone do plików projektu.

Gdy komponenty oprogramowania są już dodane do projektu i zostały skonfigurowane, należy zapisać zmiany dokonane w projekcie korzystając z menu głównego programu Kinetis Design Studio: File -> Save. Teraz można przystąpić do wygenerowania kodu źródłowego na bazie wybranych i skonfigurowanych komponentów oprogramowania. Uruchomienie generatora kodu odbywa się poprzez wybranie z menu głównego środowiska: Projekt -> Generate Processor Expert Code. Pliki z wygenerowanym kodem zostaną automatycznie dołączone do projektu programistycznego.

Szymon Panecki, EP



Rysunek 9. Od lewej: a) Lista parametrów pracy peryferium, b) Lista funkcji do sterowania peryferium, c) Lista funkcji – zdarzeń dla peryferium