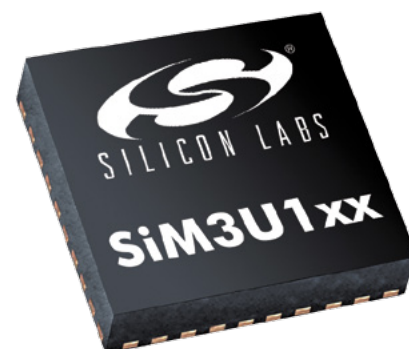


# Narzędzia dla Precision32 (2)

## Narzędzia programistyczne

*Miesiąc temu w ramach cyku artykułów dotyczących mikrokontrolerów Precision32 firmy Silicon Labs rozpoczęliśmy opis narzędzi projektowo-rozwojowych dla tej rodziny układów. Po przedstawieniu narzędzi sprzętowych tym razem zajmiemy się narzędziami programistycznymi.*



Dostępne dla mikrokontrolerów Precision32 narzędzia projektowo-rozwojowe można podzielić na trzy grupy: narzędzia sprzętowe, narzędzia programistyczne oraz oprogramowanie w postaci kodu źródłowego (rysunek 1).

Do omówionych w poprzednim miesiącu narzędzi sprzętowych należą:

- Modułowa platforma ewaluacyjna UDP (Unified Development Platform), którą użytkownik może rekonfigurować łącząc płytkę bazową UDP Motherboard z wybranymi przez siebie płytkami trzech różnych rodzajów: z mikrokontrolerem (UDP MCU card), rozszerzeniowych oraz komunikacji radiowej,

- Interfejsy sprzętowe (programatory/debugery): Silicon Labs USB Debug Adapter, Keil/ARM ULINK i Segger J-Link.

W niniejszej, drugiej części artykułu o narzędziach dla mikrokontrolerów Precision32 przedstawione zostaną narzędzia programistyczne: program AppBuilder oraz pakiety do tworzenia i rozwoju oprogramowania.

### Program AppBuilder

Każdy kod źródłowy będący oprogramowaniem dla mikrokontrolera można podzielić na dwa bloki:

- Blok oprogramowania odpowiedzialny za wykonywanie przez mikrokontroler

czynności, które są mu przewidziane w systemie, w którym pracuje.

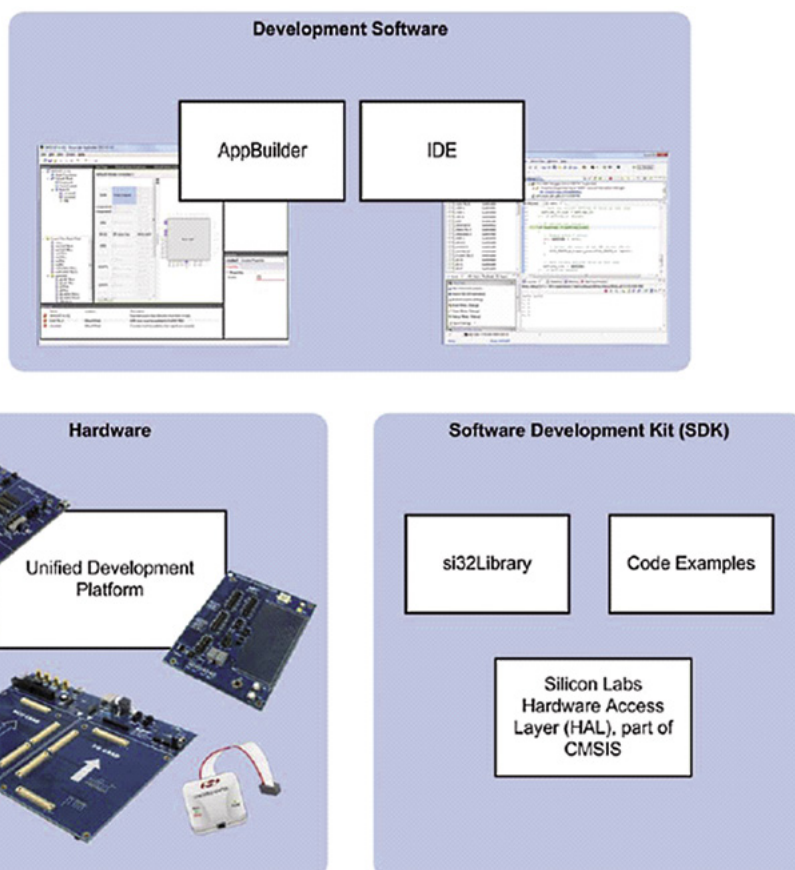
- Blok oprogramowania odpowiedzialny za skonfigurowanie mikrokontrolera.

Pierwszy z tych bloków może obejmować np. odczyt danych z czujników, realizację transmisji danych zgodnie z określonym protokołem komunikacyjnym, wyświetlanie informacji na wyświetlaczu itp. Tworzenie kodu źródłowego wymaga od programisty niejednokrotnie tylko małej (albo nawet w niektórych wypadkach – w ogóle nie wymaga) wiedzy o mikrokontrolerze, gdyż ten kod cechuje się wysokim poziomem abstrakcji i przez to operuje w niedużym stopniu bezpośrednio na zasobach sprzętowych układu.

Odwrotnie jest przy tworzeniu bloku oprogramowania odpowiedzialnego za skonfigurowanie mikrokontrolera. Blok ten odpowiada za przygotowanie zasobów sprzętowych układu do pracy, a więc włączenie ich, a następnie ustawienie parametrów funkcjonowania. Programista pisze ten kod źródłowy operując na rejestrach mikrokontrolera, bądź na niskopoziomowych funkcjach i z tego powodu musi mieć bardzo szczegółową wiedzę o mikrokontrolerze. Z racji tego, że każdy współczesny mikrokontroler jest układem złożonym, więc proces pisania kodu źródłowego będącego blokiem oprogramowania konfiguracyjnego mikrokontrolera może być zadaniem nietrywialnym, a przez to czasochłonnym.

Firma Silicon Labs ma świadomość złożoności procesu pisania oprogramowania odpowiedzialnego za skonfigurowanie mikrokontrolera, dlatego opracowała narzędzie, które ułatwia stworzenie odpowiedniego kodu konfiguracyjnego dla mikrokontrolerów z rodziny Precision32. Jest to program komputerowy o nazwie AppBuilder. Za jego pomocą, przy wykorzystaniu graficznego interfejsu użytkownika, programista może kolejno:

- Wybrać peryferia mikrokontrolera, które mają zostać włączone (odbywa się to poprzez przeciągnięcie kursorem myszy wybranych peryferiów z okna listy dostępnych dla wybranego modelu mikrokontrolera do okna aktywnych).



Rysunek 1. Narzędzia projektowo-rozwojowe dla mikrokontrolerów Precision32: narzędzia sprzętowe, narzędzia programistyczne oraz oprogramowanie w postaci kodu źródłowego

Tabela 1. Ogólne cechy pakietów do rozwijania oprogramowania dla mikrokontrolerów Precision32

Producent, nazwa pakietu, nazwa IDE	Keil/ARM, MDK-ARM, $\mu$ Vision	IAR Systems, IAR EWARM, IAR EWARM IDE	Silicon Labs, Precision32 development suite, Precision32 IDE
Główne elementy pakietu i IDE	Kompilator C/C++, Edytor, Asembler, Linker, Debugger, Przykładowe kody źródłowe.		
	Symulator, Oprogramowanie pośredniczące (Middleware): system operacyjny czasu rzeczywistego, stos TCP/IP, sterowniki USB, system plików, biblioteki GUI, biblioteki dla standardu transmisyjnego CAN (tylko w MDK-ARM).		-
Mikrokontrolery, dla których można rozwijać oprogramowanie	Rodziny mikrokontrolerów z rdzeniem: ARM Cortex—M, ARM7, ARM9.		Rodzina mikrokontrolerów Precision32.
	ARM Cortex—R4.	ARM Cortex—A, ARM Cortex—R, ARM11.	
Interfejs sprzętowy do debugowania kodu i flashowania pamięci mikrokontrolerów Precision32	Keil/ARM ULINK (dostępny w wersji podstawowej - ULINK2, oraz zaawansowanej – ULINKpro, która w przeciwieństwie do wersji podstawowej pozwala korzystać z narzędzi analizy pakietu MDK-ARM), Silicon Labs USB Debug Adapter.	Segger J-link, Segger I-jet.	Silicon Labs USB Debug Adapter.
Wersje pakietu	MDK-Lite – wersja z ograniczeniem generowania kodu wynikowego do 32 kB, jest dostępna bezpłatnie, MDK-Basic, MDK-Standard, MDK-Professional - oferowana wraz z oprogramowaniem pośredniczącym.	Pełna wersja, Pełna wersja, działająca przez 30 dni, dostępna bezpłatnie, Wersja z ograniczeniem generowania kodu wynikowego do 32 kB, dostępna bezpłatnie.	Dostępna bezpłatnie pełna wersja.

- Wybrać źródło sygnału zegarowego taktujące rdzeń mikrokontrolera i włączyć sygnały zegarowe peryferiów (odbywa się to poprzez zaznaczenie źródła sygnału zegarowego rdzenia oraz sygnałów zegarowych peryferiów na schemacie blokowym bloku zegarowego mikrokontrolera).
- Ustalić parametry działania peryferiów, które zostaną włączone (odbywa się to poprzez wybranie parametrów w oknie będącym

listą parametrów, oddzielnie dla każdego z peryferiów).

- Wykonać przyporządkowanie peryferiów do pinów mikrokontrolera (odbywa się to poprzez zaznaczenie jednego z dostępnych przyporządkowań, oddzielnie dla każdego z peryferiów).

Przykładowy zrzut ekranu programu App-Builder pokazano na **rysunku 2**.

Po wykonaniu wymienionych czynności program AppBuilder automatycznie wygeneru-

je odpowiadający wybranym przez użytkownika ustawieniom kod konfiguracyjny dla wybranego modelu mikrokontrolera i dla wybranego zintegrowanego środowiska programistycznego (IDE - *Integrated Development Environment*).

### Pakiety do tworzenia oprogramowania

Kolejnymi omówionymi narzędziami programistycznymi będą narzędzia umożliwiające tworzenie i rozwój oprogramowania dla mikro-

Tabela 2. Wybrane narzędzia dodatkowe IDE i debugera pakietów do rozwijania oprogramowania dla mikrokontrolerów Precision32

Producent, nazwa pakietu, nazwa IDE	Keil, MDK-ARM, $\mu$ Vision	IAR Systems, IAR EWARM, IAR EWARM IDE	Silicon Labs, Precision32 development suite, Precision32 IDE
Wybrane dodatkowe narzędzia IDE	Source Browser – zaawansowany spis stworzonych w projekcie struktur, zmiennych, wskaźników itp., Configuration Wizard – narzędzie będące edytorem parametrów, które są definiowane przez użytkownika za pomocą specjalnych adnotacji w komentarzach kodu źródłowego. SVCS (Software Version Control System) – system kontroli wersji,	visualSTATE – wyposażone w graficzny interfejs użytkownika narzędzie do tworzenia maszyn stanów, Mechanizm umożliwiający powiązanie IDE z systemami kontroli wersji, takimi jak np. Subversion.	Dashboard (w wolnym tłumaczeniu - tablica rozdzielcza) – okno z odnośnikami do materiałów dotyczących mikrokontrolerów Precision32 (działa: Hardware, Software, Support, Product Documentation, Training, MCU Product Search).
	Wybrane dodatkowe narzędzia debugera	Event Viewer – narzędzie wyświetlające na wykresie czas i kolejność wykonania zadań przez system operacyjny aplikacji, Code Coverage – narzędzie oznaczające linie kodu (te, które się już wykonały, nie wykonały oraz nie zostaną w ogóle wykonane), Performance Analyzer - narzędzie wyświetlające czas spędzony przez aplikację w każdej funkcji lub bloku programu, Execution Profiler - narzędzie dostarczające statystycznych danych (np. o liczbie wykonanych poszczególnych instrukcji), Logic Analyzer - narzędzie wyświetlające w formie wykresu wartości zmiennych.	Narzędzie wyświetlające w formie wykresu wartości zmiennych, Power Debugging - narzędzie wyświetlające na wykresie wartość prądu pobieranego przez mikrokontroler.

kontrolerów Precision32. Najbardziej znane są obecnie trzy pakiety:

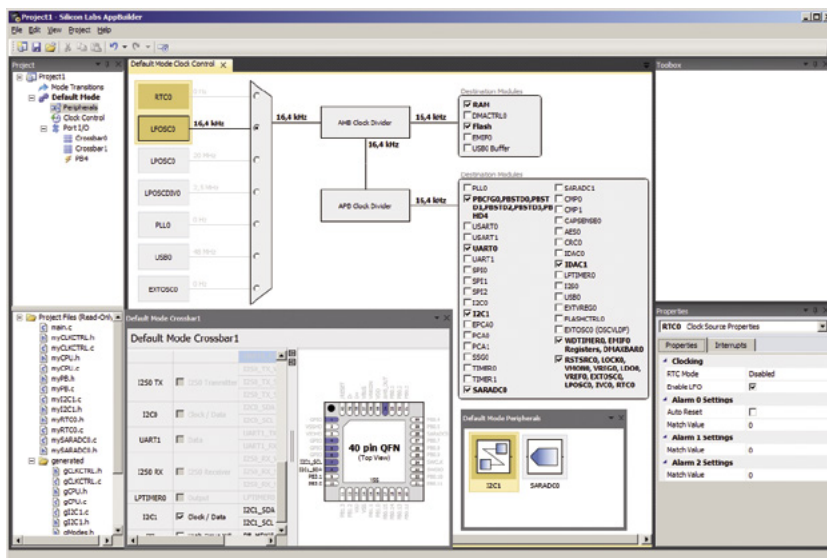
- Pakiet firmy Keil: MDK-ARM (Microcontroller Development Kit) z IDE o nazwie  $\mu$ Vision.
- Pakiet firmy IAR Systems: IAR EWARM (Embedded Workbench for ARM) z IDE o nazwie EWARM IDE.
- Pakiet firmy Silicon Labs: Precision32 development suite z IDE i nazwie Precision32 IDE.

Wszystkie pakiety umożliwiają programiście pisanie/edytowanie, kompilowanie i debugowanie kodu źródłowego oraz wgrzywanie kodu wynikowego do układu docelowego (proces ten inaczej zwany jest flashowaniem pamięci układu). Skoro te same czynności można wykonywać przy użyciu każdego z wymienionych pakietów, to można odnieść wrażenie, że są one do siebie bardzo podobne. Jest tak tylko na pozór, dlatego warto poświęcić chwilę, aby się im bliżej przyjrzeć. Dokładniejsze poznanie każdego z pakietów pozwoli czytelnikowi wybrać spośród nich ten optymalny dla siebie. Odpowiednie zestawienie porównawcze umieszczono w tabeli 1 i tabeli 2.

Na podstawie informacji z powyższych tabel możliwe jest krótkie scharakteryzowanie pakietów do tworzenia i rozwoju oprogramowania dla mikrokontrolerów Precision32.

MDK-ARM i EWARM to pakiety komercyjne, co oznacza, że aby móc ich używać, jest wymagany zakup licencji. Oprócz wersji płatnych, zarówno MDK-ARM jak i EWARM, są również dostępne bezpłatnie, ale w wersjach z ograniczeniem czasu użytkowania lub z ograniczeniem wielkości generowanego kodu wynikowego. W przeciwieństwie do MDK-ARM i EWARM, *Precision32 Development Suite* jest pakietem niekomercyjnym, stworzonym na bazie środowiska Eclipse, które jest napisaną w Javie, otwartą platformą (*framework*), niemającą żadnych narzędzi służących do tworzenia kodu i budowania aplikacji. Ma ona jednak obsługę wtyczek (*plug-ins*) rozszerzających jej funkcjonalność o możliwość rozwijania oprogramowania w różnych językach i dla różnych platform sprzętowych. Dzięki temu producenci narzędzi programistycznych dla mikrokontrolerów integrują Eclipse z opracowanymi przez siebie wtyczkami oferując w ten sposób rozwiązania programistyczne przeznaczone do tworzenia i rozwoju oprogramowania dla wybranej rodziny (lub rodzin) mikrokontrolerów. W myśl tej idei powstał pakiet *Precision32 Development Suite*, w którym firma Silicon Labs zintegrowała platformę Eclipse i opracowane przez siebie IDE (Precision32 IDE). Pakiet ten jest bezpłatnie dostępny w pełnej, pozbawionej jakichkolwiek ograniczeń wersji.

MDK-ARM i EWARM to pakiety, które mogą zostać użyte do tworzenia i rozwijania oprogramowania nie tylko dla mikrokontrolerów Precision32, ale też dla innych mikrokontrolerów z rdzeniem ARM. Jest to z pewnością



Rysunek 2. Program AppBuilder

atut dla konstruktorów używających w swoich projektach mikrokontrolerów z rdzeniem ARM od kilku producentów, gdyż w tym przypadku nie trzeba kupować dla każdej rodziny mikrokontrolerów oddzielnego pakietu do tworzenia oprogramowania. Pod tym względem na ich tle pakiet *Precision32 Development Suite* wypada słabo. Jak wskazuje jego nazwa, może zostać wykorzystany tylko do tworzenia i rozwijania oprogramowania dla mikrokontrolerów Precision32. Nie należy jednak uznawać tej cechy za dużą wadę, gdyż wielu konstruktorów korzysta tylko z jednej, wybranej przez siebie rodziny mikrokontrolerów.

Pakiety MDK-ARM i EWARM zostały wyposażone w liczne narzędzia dodatkowe. Pozwalają one sprawniej tworzyć i debugować kod źródłowy. Precision32 development suite pozbawiony jest tych narzędzi, natomiast ciekawym dodatkiem zintegrowanym w tym pakiecie jest okno z odnośnikami do aktualnych materiałów dotyczących mikrokontrolerów Precision32.

EWARM i Precision32 development suite to pakiety, które mogą używać tylko dedykowanego dla siebie interfejsu sprzętowego. Pozytywnie wyróżnia się na ich tle pakiet MDK-ARM, który współpracuje z dedykowanym interfejsem sprzętowym oraz dodatkowo może współpracować z interfejsem sprzętowym przeznaczonym dla pakietu Precision32 development suite.

## Podsumowanie

W artykule przedstawiono narzędzia programistyczne dostępne dla mikrokontrolerów z rodziny Precision32 firmy Silicon Labs. Pierwszym z narzędzi jest łatwy w obsłudze i wyposażony w graficzny interfejs program komputerowy AppBuilder, który na podstawie zaznaczonych przez użytkownika opcji wygeneruje odpowiadający im kod źródłowy konfiguracji mikrokontrolera. Dzięki temu użytkownik może w sposób szybki rozpocząć implementację właściwej aplikacji, nie tracąc czasu na przygotowanie do pracy zasobów wewnętrznych układu.

Kolejnymi narzędziami programistycznymi są pakiety do tworzenia oraz rozwoju oprogramowania IAR EWARM, Keil/ARM MDK-ARM i *Silicon Labs Precision32 Development Suite*. Największą zaletą pakietu *Precision32 Development Suite* jest bezpłatna dostępność pełnej wersji narzędzia. Największymi zaletami pakietów IAR EWARM i Keil/ARM MDK-ARM są zaawansowane narzędzia dodatkowe oraz możliwość tworzenia oraz rozwoju oprogramowania dla wielu rodzin mikrokontrolerów.

Była to druga część artykułu o narzędziach dla mikrokontrolerów Precision32. Trzecia część poświęcona będzie oprogramowaniu w postaci kodu źródłowego. Tematyka kolejnych artykułów to:

- Rozpoczęcie pracy z mikrokontrolerem Precision32 krok po kroku, przy wykorzystaniu zestawu ewaluacyjnego,
- Porównanie układów Precision32 z innymi rodzinami mikrokontrolerów z rdzeniem ARM Cortex-M3,
- Autorski projekt prostego zestawu startowego.

Dystrybutorem mikrokontrolerów Silicon Labs Precision32 oraz pakietu Keil MDK-ARM w Polsce jest firma WG Electronics. Autor składa podziękowanie Panu Tadeuszowi Górnickiemu, prezesowi firmy WG Electronics oraz Panu Sándor Csüllög, inżynierowi aplikacyjnemu z firmy Silicon Labs, za pomoc w realizacji artykułu.

**Szymon Panecki**  
Wydział Elektroniki,  
Politechnika Wrocławska  
szymon.panecki@pwr.wroc.pl

## Bibliografia:

- [1] [www.silabs.com](http://www.silabs.com), AN675 „Precision32 development suite overview”
- [2] [www.silabs.com](http://www.silabs.com), AN670 „Getting started with the Silicon Labs Precision32 AppBuilder”
- [3] [www.silabs.com](http://www.silabs.com), AN667 „Getting started with the Silicon Labs Precision32 IDE”