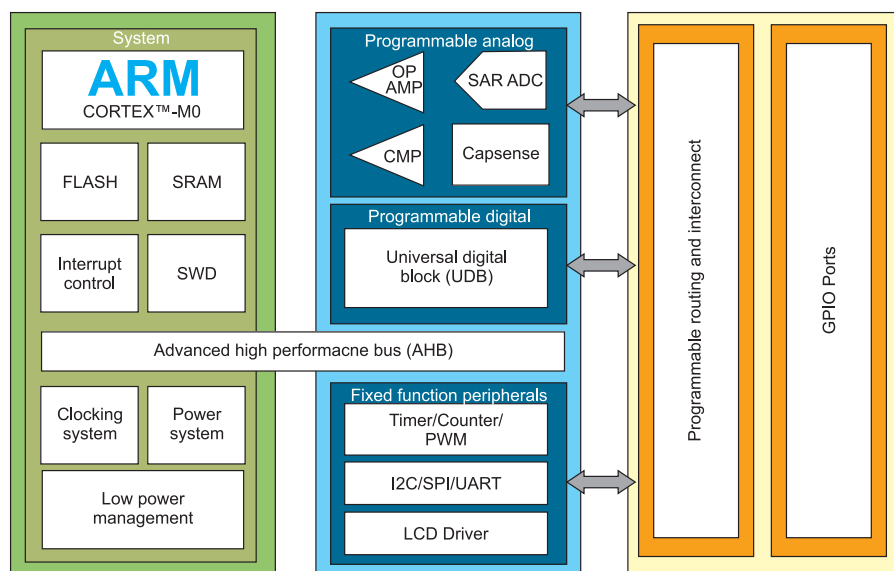


Psoc4 – PSoC z ARM Cortex M0 za dolara

Nieco ponad rok temu Cypress wprowadził do oferty zestaw PSoC 4 Pioneer Kit (CY8CKIT-042) zawierający pierwszy PSoC z rdzeniem ARM Cortex-M0. Sercem zestawu był mikrokontroler typu CY8C4245AXI z rodziny PSoC4. Zestaw miał wyprowadzenia zgodne z Arduino i wbudowany programator/debugger. Jako środowisko programistyczne użyto – podobnie jak w pozostałych rodzinach – programu PSoC Creator umożliwiającego konfigurowanie i programowanie układów. Była to pełna, funkcjonalna wersja oprogramowania bez żadnych ograniczeń na wielkość kodu wynikowego, zawartość bibliotek itp. Kolejny krok mający na celu upowszechnienie się tych bardzo interesujących układów ma stanowić zestaw CY8CKIT-049 4xxx.



Rysunek 1. Schemat blokowy mikrokontrolera z rodziny PSoC4 (za nota producenta)

Aktualnie w ofercie producenta są zestawy zawierające mikrokontrolery CY8C4245AXI i CY8C4125AXI. Dla przypomnienia, budowę wewnętrzną układów rodziny PSoC4 pokazano na **rysunku 1**, a porównanie podstawowych parametrów tych mikrokontrolerów zamieszczono w **tabeli 1**.

W dalszym ciągu zestawy można programować za pomocą darmowego środowiska PSoC Creator. Najlepiej, aby było ono w wersji 3.0 lub nowszej, ponieważ do tej wersji udo-

stępiono możliwość wykorzystania wszystkich możliwości zestawu.

Firma Cypress ma ambicję wyróżniania się na rynku procesorów z rdzeniem Cortex-M0. Poprzednio był to pierwszy PSoC z rdzeniem M0, dodatkowo o najniższym poborze mocy w porównaniu z wyrobami konkurencyjnymi. Teraz zestawami „049” Cypress ustanowił rekord cenowy dla zestawu uruchomieniowego Cortex-M0 oferując go w cenie 4 dolarów, niezależnie od zastosowanego w nim mikrocontro-

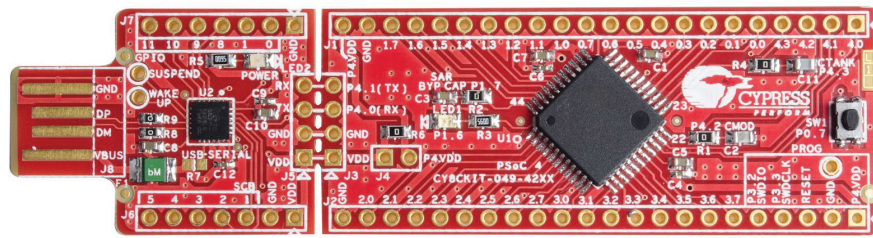
lera. Patrząc po sposobie oznaczenia zestawu można spodziewać się kolejnych z PSoC4, wyposażonych w dostępne już na rynku procesory z rodziny 4000. Takie podejście producenta, zapewne zaskarbi przychylność nowych użytkowników. Wygląd zestawu pokazano na **fotografii 2**.

Niestety, w rzeczywistości niczego nie ma za darmo, więc pewnie gdzieś czai się jakiś kruczek umożliwiający uzyskanie tak niskiej ceny. Pierwszą cechą rzucającą się w oczy jest „niskobudżetowe” opakowanie, będące jednocześnie skróconą instrukcją obsługi. Płytką – w przeciwieństwie do wcześniej opisywanych zestawów – jest umieszczona w kartonowej kopercie i nie są do niej dołączone żadne akcesoria typu: kabel USB, zworki, złącza. Sama konstrukcja mechaniczna zestawu opiera się na dzielonej płytce drukowanej. Część z programatorem ma wtyk USB wykonany w formie złącza krawędziowego. W niektórych komputerach PC może więc być konieczne zaopatrzenie się w kabel przedłużający USB, gdyż dosyć szeroka płytka może zablokować dostęp do pozostałych gniazd USB lub jej dołączenie może być wręcz niemożliwe. Wszystkie wyprowadzenia GPIO mikrokontrolera PSoC4 są dostępne w dwóch rzędach o rozstawie zgodnym z uniwersalną płytką drukowaną lub stykową, co ułatwia samodzielną egzystencję drugiej części zestawu w docelowym układzie prototypowym.

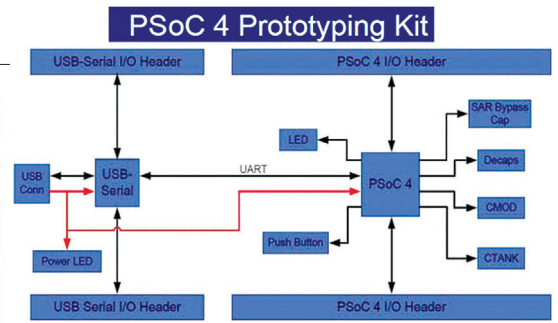
Bardzo istotną zmianą wprowadzoną w „049” jest sposób zapisu programu do pamięci Flash wbudowanego mikrokontrolera. W zestawie Pioneer Kit wlotowany w płytkę mikrokontroler PSoC5 pełnił funkcję programatora-debuggera zastępując funkcjonalnie programator MiniProg3. W płytkach „049”, podobnie jak w Arduino, wbudowano tylko most komunikacyjny dla bootloadera mikrokontrolera testowego. Most jest oparty o procesor CY7C65211 będący uniwersalnym układem interfejsowym USB/Serial, domyślnie pracującym w trybie USB/UART. Zestaw można wykorzystać do eksperymentowania z CY7. Niewykorzystane wyprowadzenia wyprowadzone są na złącza GPIO w części USB. Taki sposób

Tabela 1. PSoC4 „049” porównanie wyposażenia mikrokontrolerów z zestawów „049”

Typ	VCC [V]	Flash [KB]	SRAM [kB]	Fclk [MHz]	GPIO	PWM Timer CLK	USART	UDB	LCD	CAP SENSE	A/C	OPA
CY8C4125AXI-483 CY8CKIT-049-41xx	1.71-5.5V	32	4	24	36	4	2	0	Y	Y	12-bit 806 ksp/s	2
CY8C4245AXI-483 CY8CKIT-049-42xx	1.71-5.5V	32	4	48	36	4	2	4	Y	Y	12-bit 1 msp/s	2



Fotografia 2. Wygląd płytki ewaluacyjnej typu „049”



programowania umożliwia obniżenie kosztów zestawu, ale nie daje możliwości debugowania programu. Do zapisu programu oprócz bootloadera można użyć także programatora MiniProg3. Płytkę ma możliwość podłączenia programatora zewnętrznego do złącza J2 w części oznaczonej PROG.

Wyposażenie budżetowego zestawu musi być skromne. Do dyspozycji użytkownika oddano: diodę LED, mikroprzełącznik, dwa złącza GPIO, wspomniany wcześniej most USB/UART. Jest to standardem wśród tanich zestawów. Wsparcie techniczne i niezbędna dokumentacja, w tym schemat zestawu, zamieszczono na stronie producenta

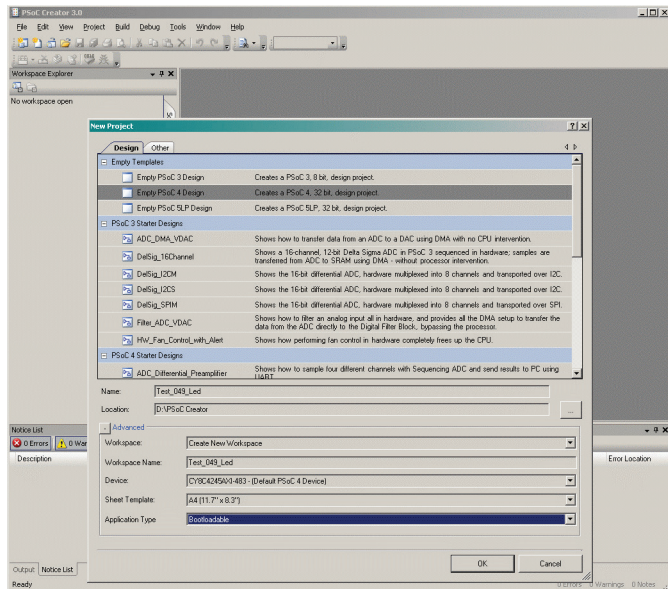
Praca z zestawem

Aby rozpocząć pracę z zestawem, należy ze strony producenta pobrać, zainstalować i zarejestrować oprogramowanie PSoC Creator.

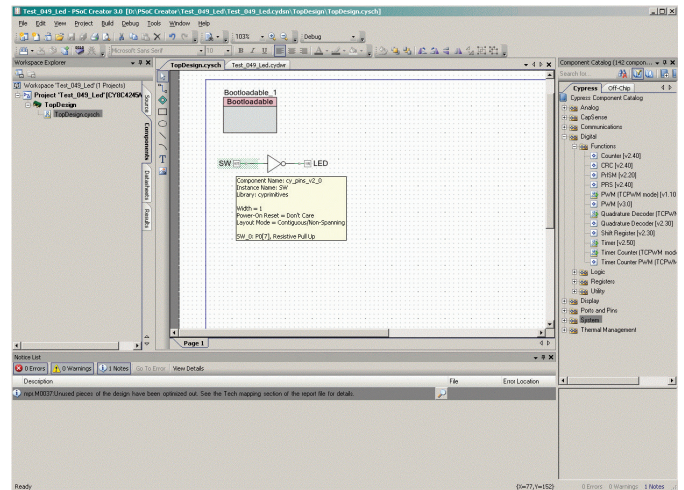
Po dołączeniu płytki i zainstalowaniu driverów USB jest możliwe sprawdzenie zestawu za pomocą aplikacji testowych. W pierwszej kolejności należy zwrócić uwagę na konieczność przygotowania aplikacji uwzględniającej obsługę programowania układu przez Bootloader. W tym celu otwieramy nowy projekt, uwzględniający typ procesora oraz rodzaj aplikacji – ważne, aby zmienić ją z *Normal* na *Bootloadable* zgodnie z **rysunkiem 3**.

Kolejną przechodzimy do konfiguracji tworzenia najprostszej aplikacji testowej, pokazanej na **rysunku 4**, wykorzystującej zestaw „049”. W tym celu, na Pulpit przeciągamy elementy bloków bootloadera – *Bootloadable* i aplikacji, czyli wejście i wyjście cyfrowe oraz bramkę *Not*. Prosta aplikacja sygnalizuje przyciśnięcie SW1 poprzez świecenie diody LED. Konfigurację GPIO pokazano na **rysunku 5**. Wejście SW1 należy ustawić w trybie *PullUp*.

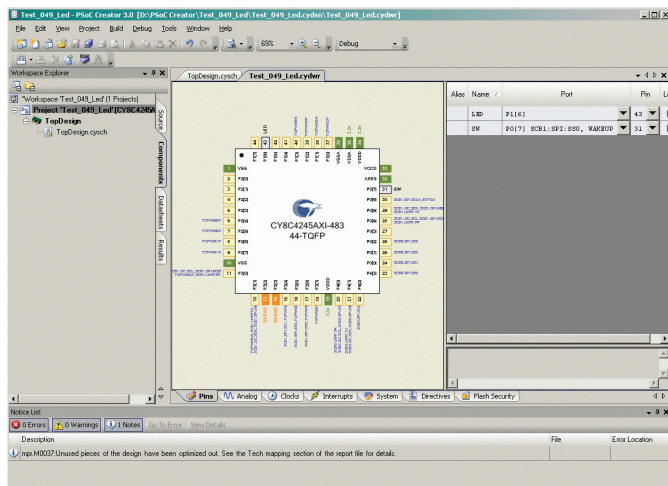
Następnie do projektu należy dołączyć aplikację bootloadera. Kody źródłowe i pliki *hex* oraz *elf* znajdują się w przykładowych projektach do pobrania na stronie zestawu znajdującej się pod adresem <http://goo.gl/3r6Ozv>. W celu przygoto-



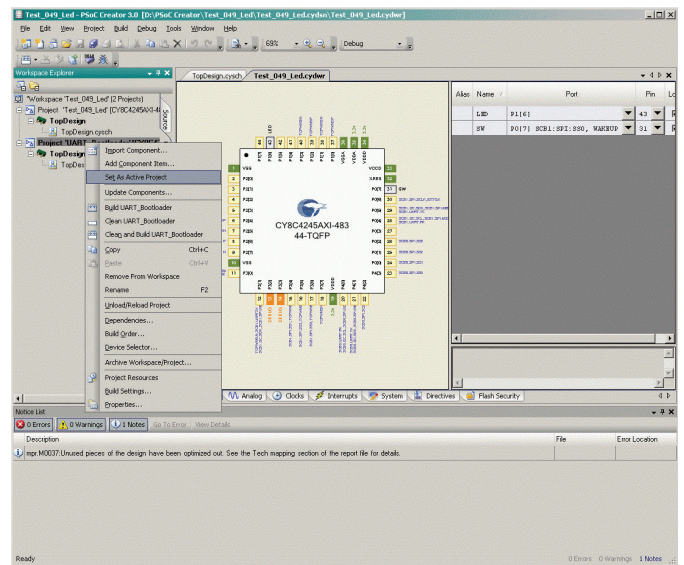
Rysunek 3. Tworzenie nowej aplikacji „Bootloadable”



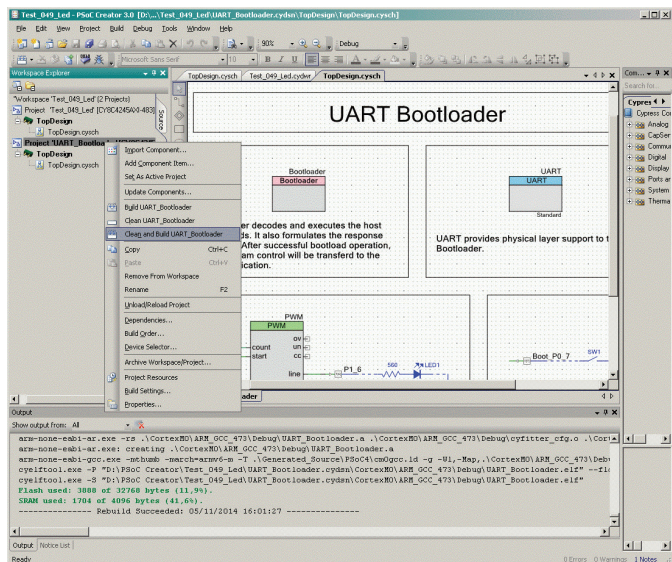
Rysunek 4. Wygląd aplikacji testowej



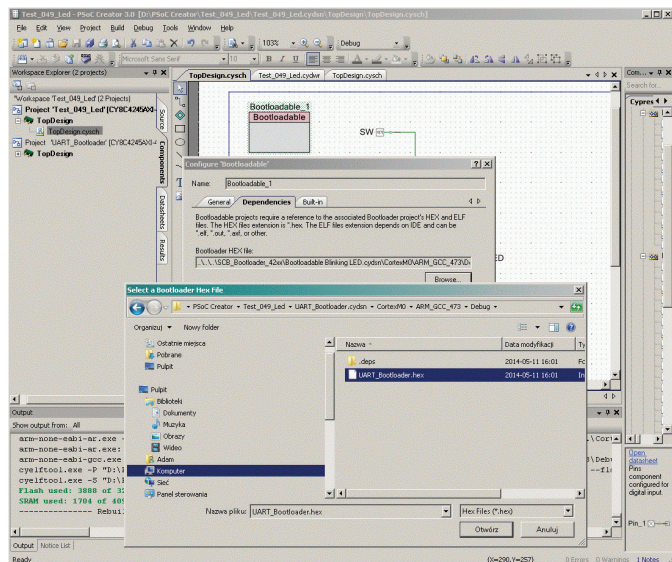
Rysunek 5. Konfigurowanie GPIO



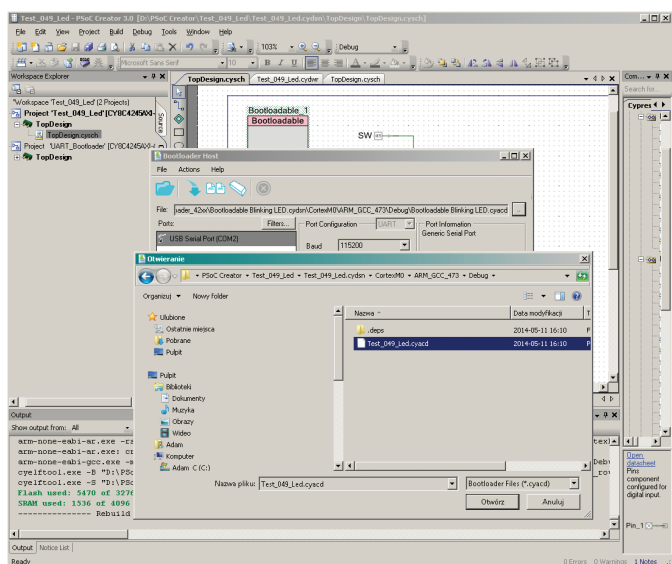
Rysunek 6. Ustawienie aktywnego projektu bootloadera



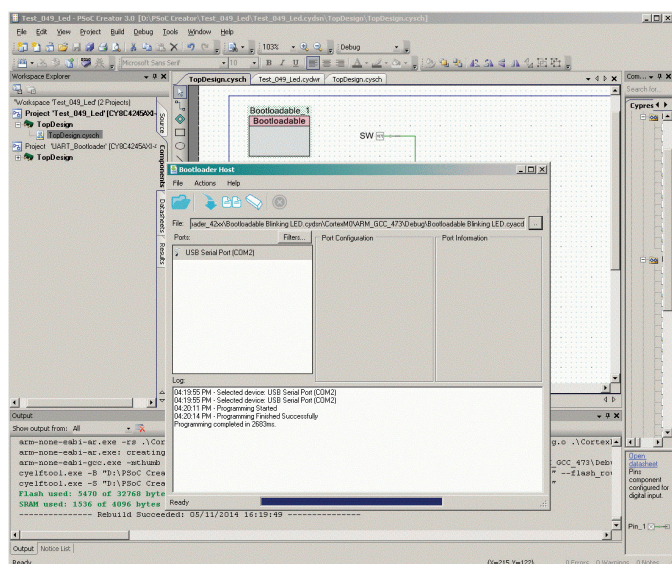
Rysunek 7. Kompilowanie bootloadera



Rysunek 8. Konfigurowanie bootloadera



Rysunek 9. Aplikacja obsługująca ładowanie projektu



Rysunek 10. Programowanie mikrokontrolera na płytce testowej

wania bootloadera UART (*UART_Bootloader.cydsn*) jest konieczne ustawienie jego projektu jako domyślnego, jak na **rysunku 6** oraz skompilowanie źródła (**rysunek 7**). Aby łatwiej poruszać się w gąszczu katalogów projektu, najwygodniej przekopiować projekt bootloadera do katalogu własnego projektu. Jeżeli wszystko przebiegło bez problemów, aplikacja jest gotowa do użycia bootloadera UART.

Następnie wracamy do projektu testowego, ustawiamy go jako domyślny i informujemy kompilator, gdzie znajdują się pliki bootloadera wybierając na schemacie blok *Bootloadable* i zakładkę *Configure/Dependencies*. Ustawiamy ścieżki do plików bootloadera (**rysunek 8**) i kompilujemy aplikację testową. Ostatnim krokiem jest załadowanie aplikacji do pamięci mikrokontrolera. Po wybraniu z menu *Tools/Bootloader/host* aplikacji ładującej, informujemy ją o położeniu projektu testowego przygotowanego do załadowania, zgodnie z **rysunkiem 9**.

Ze względu na ograniczoną liczbę obsługiwanych portów COM, warto wcześniej w *Me-*

nedżerze Urządzeń systemu Windows ustawić jak najniższy numer portu – musi on być mniejszy lub równy 10 (COM1...COM10). Następnie usuwamy płytkę z portu USB, przytrzymujemy przycisk SW1 i ponownie dołączamy do portu USB. Włączenie zasilania przy przytrzymanym SW1 aktywuje bootloader, sygnalizuje to szybkie miganie niebieskiej diody LED. Rozwiązanie nie jest zbyt eleganckie, można by zastosować dodatkowy przycisk, ale pewnie podniosłoby to koszt zestawu.

Aby zaprogramować procesor zestawu uruchomieniowego w aplikacji hosta wybieramy z menu *Actions/Program* i po chwili, jeżeli wszystko przebiegło pomyślnie, otrzymujemy komunikat o zaprogramowaniu mikrokontrolera (**rysunek 10**).

Podsumowanie

Pomimo odbiegającego od standardowego sposobu programowania, łatwość użycia i więcej niż przystępna cena są dużym plusem dla zestawu CY8CKIT-049. Co prawda nie jest on

pełnoprawną platformą startową ze względu na ograniczenia debugera, ale dzięki niskiej cenie w połączeniu z funkcjonalnym, darmowym środowiskiem PSoC Creator, jest wyjątkowo atrakcyjną propozycją dla użytkowników rdzeni Cortex-M0 ułatwiającą podjęcie decyzji czy kontrolery PSoC staną się naszymi ulubionymi i czy warto w przyszłości zainwestować w droższe narzędzia tym bardziej, że firma Cypress już przygotowuje serię mikrokontrolerów 4000 słabiej wyposażonych w peryferia analogowe, ale za to dostępnych w obudowach o zmniejszonej liczbie wyprowadzeń typu SO8, SO16, QFN16 i w wyjątkowo niskich cenach (np. CY8C4013SXI za 0,29 USD/szt. przy zamówieniu 100 szt.), idealnych dla zastosowań mobilnych, mogących nawet wyprzeć z wielu zastosowań mikrokontrolery firmy Atmel ATtiny45/85 czy Microchip PIC12xx.

Opis zestawu KIT040 oraz rodziny 4000 wkrótce opublikujemy na łamach EP.

Adam Tatuś, EP
adam.tatus@ep.com.pl