Zestaw narzędzi dla użytkowników mikrokontrolerów XE8000

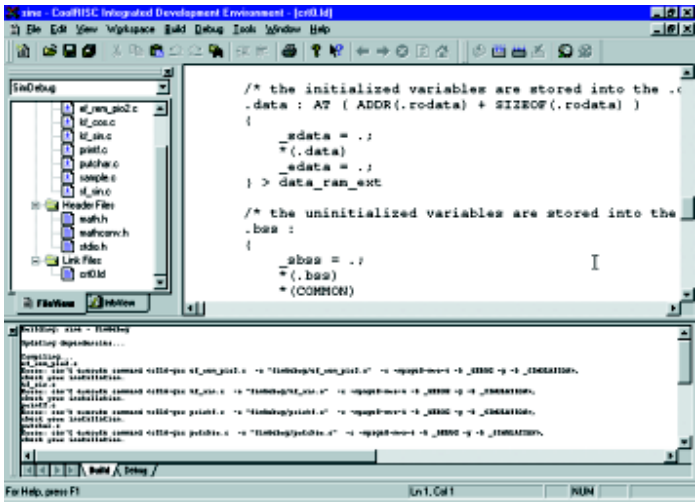
W artykule na str. 51 przedstawiliśmy rodzinę nowych mikrokontrolerów produkowanych przez szwajcarską firmę Xemics. Ich parametry eksploatacyjne - przede wszystkim bardzo mały pobór prądu i możliwość zasilania z pojedynczej baterii litowej - powinny zainteresować wielu konstruktorów. Opisy mikrokontrolerów jakie można znaleźć w pedeefowych dokumentach nie zawsze w pełni je charakteryzują. Lepiej jest je po prostu „dotknąć".

XE8000 ProStart

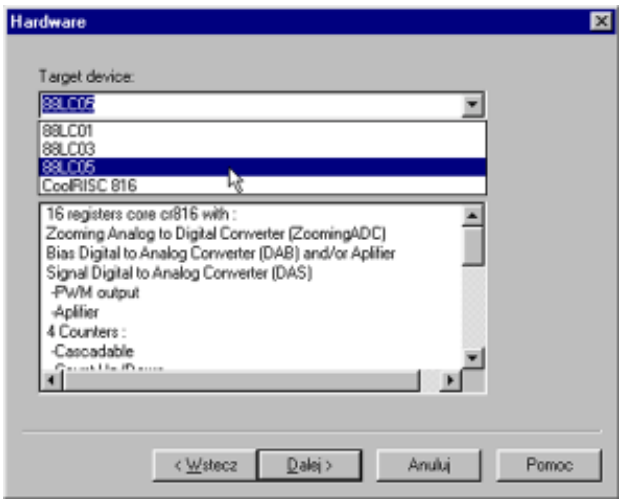
Prezentujemy starter kit, jaki firma Xemics opracowała w celu promocji swoich nowych produktów. Nazywa się on *XE8000 ProsStart* i służy do uruchamiania prostych aplikacji dla mikrokontrolerów XE88LC01M, XE88LC03M i XE88LC05M. Jak pamiętamy, w skład rodziny XE8000 wchodzą ponadto układy XE88LC02, XE88LC04, XE88LC06 i XE88LC08. ProsStart zawiera płytkę próbną z podstawką dla mikrokontrolera, płytkę programatora umożliwiającego programowanie wewnętrznej pamięci Flash mikrokontrolera poprzez interfejs RS232 oraz dwa CD-ROM-y z dokumentacją techniczną mikrokontrolerów oraz z oprogramowaniem *CoolRIDE*.

Płytka ewaluacyjna

Na płytce ewaluacyjnej nie widzimy zbyt wielu elementów. Wszystkie mikrokontrolery Xemicsa są wykonywane wyłącznie w wersjach do montażu powierzchniowego. Głównym elementem jest więc, niezbyt często spotykana, specjalna podstawka pod układy wykonane w obudowie LQFP64. Podstawka to nie jest właściwe określenie. W plastikowym korpusie zamontowano precyzyjnie ułożone, pozłacane, lekko sprężynujące styki, na których umieszczany jest układ scalony. O dokładności wykonania świadczy przyjęty dla wyprowadzeń raster 0,5 mm. Układ jest dociskany do styków przykrywką z zatrzaskami gwarantującymi pewność kontaktu. Niestety, mimo powszechnie stosowanego ścięcia krawędzi obudów takich jak TQFP czy LQFP, nie przewidziano żadnej blokady mechanicznej zabezpieczającej przed nieprawidłowym zorientowaniem układu w podstawce. Jedynie w miejscu, w którym powinien znaleźć się ścięty róg obudowy układu, umieszczono czerwony punkt. Wkładanie mikrokontrolera do podstawki musi więc być wykonywane bardzo uważnie. Jego nieprawidłowe zorientowanie może zakończyć się zniszczeniem układu. Jak już wiemy, mikrokontrolery XE8000 nie są zbyt wymagające pod względem napięcia zasilającego. Cecha ta może być w łatwy sposób zweryfikowana za pomocą płytki ewaluacyjnej. Znajduje się na niej gniazdo do umieszczenia baterii litowej w obudowie pastylkowej o średnicy do 15 mm. Bateria może być odłączona za pomocą specjalnego przełącznika. Wówczas zasilanie jest pobierane z płytki programatora. Jeśli jednak tego nie zrobimy, to w przypadku występowania napięcia na łączówce płytki próbnej z płytką programatora, bateria zostanie odłączona automatycznie. Mikrokontroler może być zerowany ręcznie przyciskiem, chociaż odpowiednie elementy zapewniają poprawny start procesora po włączeniu zasilania. Do komunikacji z operatorem służą diody świecące sterowane z por-



Rys. 1



Rys. 2

tu PB oraz mikroprzyciski dołączone do portu PA. Jeśli nie przewidujemy ich wykorzystania, mogą być odłączone poprzez wyjęcie jumperów. Wszystkie wyprowadzenia mikrokontrolera są dołaczone do złącza szpilkowego umieszczonego na krawędzi płytki. Tuż przy podstawce znajduje się rezonator kwarcowy 32768 Hz, który może być odłączany za pomocą jumpera. Mikrokontroler „przechodzi" wówczas na pracę z wewnętrznym oscylatorem RC. W skład zestawu uruchomieniowego wchodzą 3 mikrokontrolery XE88LC05M. Wszystkie mają zapisany w pamięci program zapalający kolejno diody świecące umieszczone na płytce próbnej. Wykorzystanie samej płytki do innych celów niż demonstracja wspomnianych efektów jest dość uciążliwe, gdyż nie przewidziano na niej nawet centymetra kwadratowego uniwersalnego pola montażowego. Układy własne, których działanie chcielibyśmy sprawdzić, muszą być więc zmontowane na zewnątrz. Połączenia z mikrokontrolerem na płytce starter kitu trzeba wykonać dodatkowym kablem, którego nie ma w zestawie. Można też umieścić w swoim układzie taką podstawkę, jaka jest na płytce próbnej zestawu uruchomieniowego. Tylko skąd ją wziąć? Cały zestaw ewaluacyjny służyłby w takim przypadku jedynie jako programator, a mikrokontroler byłby przenoszony z płytki na płytkę - to dość męczące. Inne rozwiązanie to umieszczenie na własnej płytce specjalnej łączówki, która służyłaby do łączenia jej z płytką programatora ze starter kitu.

Płytka programatora

Typowa konfiguracja zestawu uruchomieniowego, przewidziana przez producenta, to płytka próbna połączona z płytką programatora (**fot. 1**). Przewidziano do tego celu gniazda DSUB25. Na płytce programatora umieszczono zasilacz zapewniający napięcia wymagane podczas programowania mikrokontrolera, a także podczas pracy w trybie z zasilaniem zewnętrznym (nie z baterii). Aby mógł on prawidłowo działać, niezbędne jest doprowadzenie do specjalnego gniazda napięcia 13...15 V (może być niestabilizowane). Programowanie układu odbywa się z komputera PC poprzez interfejs RS232. Na płytce programatora znajdują się dwa gniazda umożliwiające dołączenie komputera. Jedno z nich jest wykorzystywane wyłącznie przez programator, drugie zaś służy do transmisji między mikrokontrolerem a komputerem z uruchamianym programem. W instrukcji zwraca się uwagę na to, że kabel połączeniowy nie może być typu *null modem*.

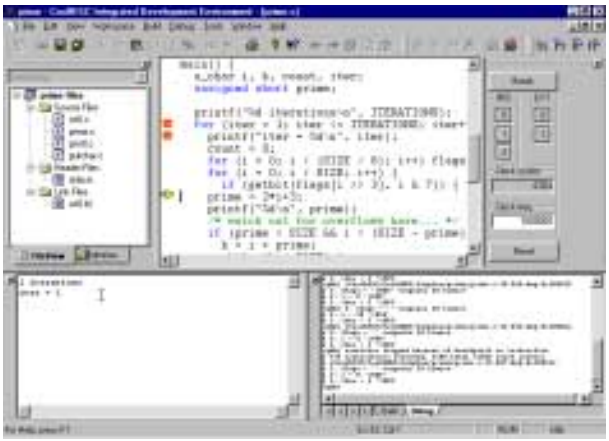
Wróćmy do możliwych konfiguracji uruchamianego układu. Wspominałem o możliwości łączenia własnej płytki z płytką programatora w celu zaprogramowania mikrokontrolera we własnym układzie. Do tego celu nie jest stosowany typowy interfejs ISP. Układy rodziny XE8000 wymagają na wybranych wyprowadzeniach odpowiednich sygnałów i napięć, a także dołączenia rezystorów podciągających i kondensatorów blokujących. To wszystko jest dość znacznym utrudnieniem dla operatora, o tyle niezrozumiałym, że przecież znane są rozwiązania, w których wystarczy np. chwilowe zajęcie portu szeregowego podczas programowania.

CoolRIDE

CoolRIDE to pakiet narzędzi programowych zawierający: profiler i menedżer projektu, asembler, kompilator języka C, linker, symulator, pomoc *online* oraz debuger mogący wyświetlać instrukcje języka C. Wszystkie programy są zintegrowane w jednym pakiecie, z typowym dla Windows interfejsem graficznym. Główne okno programu przedstawiono na **rys. 1**.

CoolRIDE umożliwia utworzenie własnego projektu, w skład którego będą wchodzić wszystkie niezbędne pliki przygotowanego oprogramowania. Zastosowane rozwiązania są podobne do stosowanych w wielu innych programach tego rodzaju. Dzięki *CoolRIDE* programista może w sposób uporządkowany tworzyć i archiwizować swoje „dzieło". Pracę trzeba rozpocząć od skonfigurowania środowiska. Najważniejsze to podanie w menu *Tools*, z którym portem szeregowym będzie współpracować zestaw uruchomieniowy i jakie będzie napięcie zasilające mikrokontroler. Tworzenie nowego projektu jest ułatwione dzięki wbudowanemu wizardowi, prowadzącemu użytkownika „za rękę".

Na wstępie należy podać, na jaką wersję mikrokontrolera będzie pisany program. Po podświetleniu odpowiedniej pozycji na liście rozwijanej, w okienku poniżej jest przedstawiana krótka charakterystyka wybranego typu (**rys. 2**). Po przebrnięciu przez wszystkie podpowiedzi kreatora projektu można rozpocząć pisanie programu. Mamy do wyboru tworzenie plików ciała programu w języku C, plików nagłówkowych C (*C header files*), programów źródłowych w asemblerze (ich domyślne rozszerzenie to.*s*) i plików dołączanych (*include files*). Można też dokumentować projekt w plikach tekstowych. Dobrym zwyczajem, pozwalającym na łatwiejsze panowanie nad projektem, jest dzielenie programu na moduły. Moduły mogą być ponadto wykorzystywane wielokrotnie w innych projektach. Zakończenie przygotowania programu polega na skompletowaniu poszczególnych jego składników i po-



Rys. 3

łączeniu w całość. Wszystkie moduły przed linkowaniem muszą być wyczyszczone z błędów. Ewentualne nieprawidłowości są wykrywane na etapie kompilacji. Kompilacja wszystkich modułów, szczególnie w przypadku rozbudowanych projektów, może zajmować sporo czasu. Aby niepotrzebnie go nie wydłużać, usuwanie błędów najlepiej jest przeprowadzać na kolejnych modułach programu, wykorzystując komendę *Build Single File*. Dopiero na końcu wykonujemy polecenie *Incremental Build*, powodujące zlinkowanie wszystkich modułów w jeden pakiet. Kompilator bezlitośnie wskazuje nam miejsca programu, w których występują błędy formalne, a których wcześniej nie zauważyliśmy. Wprawny programista dużą ich część może dostrzec już podczas pisania programu, ale zamienienie np. średnika z dwukropkiem nie zawsze jest łatwe do wychwycenia.

Znacznie gorzej jest z wykryciem błędów logicznych. Tych niestety nie widać, a ujawniają się dopiero podczas działania programu. To powszechnie nazywane „pluskwy" (*bugs*). Program trzeba więc oczyścić z tych nieprzyjemnych, a czasami wręcz wrednych robaczków. Służy do tego, jak sama nazwa wskazuje, debuger. Uruchamiamy go, naciskając na przycisk *Start Debugger* lub klawisz F5. Jednak wcześniej program musi być prawidłowo zlinkowany. Okno debugera jest podzielone na pięć części (**rys. 3**). Widzimy strukturę projektu (właściwie w tym momencie informacja ta jest mało przydatna, można więc okno to zamknąć), okno z fragmentem uruchamianego programu, okno wyników programu i okno komunikatów debugera. Z menu *View* można ponadto dodać szereg innych informacji, przy czym czytelne ich rozmieszczenie na ekranie jest nie lada sztuką. *CoolRIDE* ma pod tym względem pewną niedogodność. Komunikaty w oknie debugera i oknach zasobów procesora są wyświetlane tak małą czcionką, że nawet na monitorze 17-calowym pracującym z rozdzielczością 800x600 były bardzo nieczytelne i nie udało mi się tego zmienić.

Rozmiary artykułu nie pozwalają na omówienie szczegółów programu *CoolRIDE*. Jest to o tyle mało istotne, że działa on jak wiele mu podobnych. Istotne jest to, że można w nim wykonywać pojedyncze instrukcje procesora zarówno na poziomie języka C, jak i rozkazów asemblerowych. Możemy podglądać i zmieniać zawartość rejestrów, pamięci, śledzić wykonywanie programu w trybie ciągłym, ewentualnie z ustawionymi pułapkami. Wszystkie lub wybrane zmienne programu są uwidocznione w oknie *Watches*, w którym można również zmieniać ich wartość.

No cóż, program można debugować w nieskończoność. Jak wiadomo, nie ma programu bez błędów. W którymś momencie trzeba jednak podjąć męską decyzję i uznać, że pozostałe uchybienia są do zaakceptowania. Pracę nad projektem kończymy zapisując ostateczną wersję programu do pamięci mikrokontrolera. Odpowiednie polecenie znajduje się w menu *Tools->Starter Kit*. Trzeba jeszcze określić typ mikrokontrolera, wartość napięcia zasilającego, wskazać plik binarny, który będzie zapisany w pamięci Flash i nacisnąć klawisz *Upload software*. Programowanie układu trwa zaskakująco długo. Warto więc „przyłożyć się" do uruchamiania programu „na sucho", podczas symulacji.

Mikrokontrolery XE8000 to nowość na rynku zdominowanym przez innych „wielkich" producentów. W tej dziedzinie jest niezwykle trudno przejąć klientów. Mają oni swoje narzędzia, przyzwyczajenia, własne biblioteki tworzone często latami.

Zachętą do zainteresowania się świeżymi produktami jest zaproponowanie nowej jakości. Rodzina XE8000 z pewnością może zwrócić na siebie uwagę. Ma szansę wypełnienia pewnych nisz rynku elektronicznego.

Jarosław Doliński, AVT
jaroslaw.dolinski@ep.com.pl

Dodatkowe informacje

Zestaw prezentowany w artykule udostępniła redakcji firma JM elektronik, tel. (32) 339-69-01, www.jm.pl.

Dodatkowe informacje można znaleźć na stronie producenta: http://www.xemics.com.