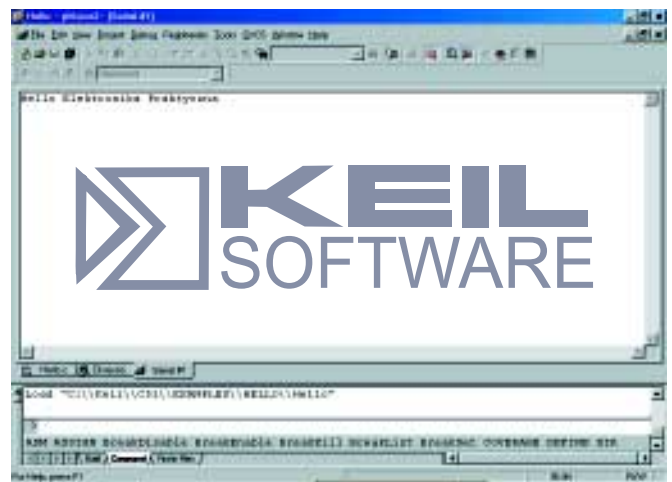


# Keil C51

Język programowania wysokiego poziomu i mikrokontroler. Dla wielu elektroników brzmi to jak herezja - pokutuje pogląd, że tylko assembler pozwala w pełni wykorzystać możliwości procesora. Opisany kompilator języka C dla procesorów rodziny 8051 może przekonać wszystkich, że tak nie jest. Umożliwia on pełną kontrolę nad tworzonym programem i w wielu przypadkach generuje kod lepszy niż można było się spodziewać.

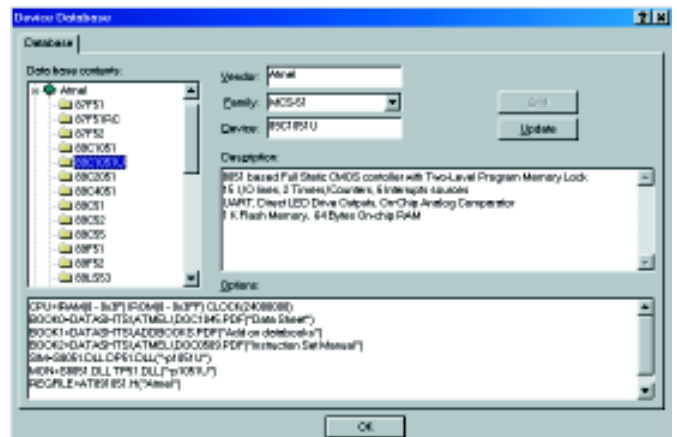
Keil C51 jest kompilatorem języka C dla procesorów rodziny 8051 zgodnym ze specyfikacją ANSI C. Premiera najnowszej wersji, oznaczonej numerem 6, odbyła się w lutym bieżącego roku. W porównaniu do wcześniejszych wersji nastąpiła znaczna poprawa algorytmów optymalizacji kodu wynikowego i to zarówno pod względem czasu wykonywania, jak i wielkości programu wynikowego. Powiększona została również lista obsługiwanych procesorów.

*Firma Keil jest kojarzona przede wszystkim z kompilatorami języka C dla procesorów rodziny 8051. Kariera tych kompilatorów rozpoczęła się w 1988 roku od prostego kompilatora pracującego pod kontrolą MS-DOS. Obecnie trudno mówić o samym kompilatorze, gdyż oferowany system zawiera szereg narzędzi wspomagających zarówno pisanie programu, jak i jego analizowanie. Kompilatory firmy Keil zawsze były uznawane za jedne z najlepszych. Czy tak jest faktycznie? Na to pytanie postaramy się odpowiedzieć w artykule.*

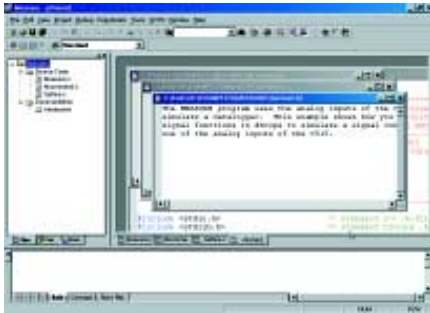


Obecnie za pomocą Keil C51 można pisać programy dla procesorów zgodnych z 8051 produkowanych przez następujące firmy: Analog Devices, Atmel, Dallas, Infineon, Intel, ISSI, OKI, Philips, ST Microelectronics, TDK, Temic, Triscent i Winbond. Wprowadzone algorytmy optymalizacji umożliwia-

ją zmniejszenie kodu wynikowego o 5..15% w porównaniu do poprzedniej wersji kompilatora. Dostępnych jest ponad 100 funkcji języka ANSI C oraz zmienne i dyrektywy charakterystyczne dla wybranego typu procesora. Zmienne mogą być umieszczane w dowolnym miejscu pamięci, również w pamię-



Rys. 1.



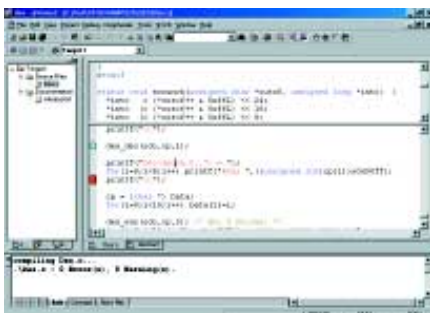
Rys. 2.

ci stałej. Program dla mikrokontrolera zasadniczo nie różni się od standardowego programu w języku C, więc przy podstawowej znajomości języka oraz wiedzy o budowie mikrokontrolera napisanie pierwszego programu nie zajmie dużo czasu.

Prawdziwa siła Keil C51 nie tkwi jednak w jego kompilatorze. Kompilator C51 wchodzi w skład pakietu narzędzi programistycznych, które pracują pod kontrolą zintegrowanego środowiska programisty o nazwie  $\mu$ Vision2. We wszystkich fazach programowania  $\mu$ Vision2 stara się wspierać programistę. Podczas wyboru procesora docelowego (rys. 1) jest wyświetlana jego krótka charakterystyka. Baza danych o procesorach zawiera informacje o pamięci, układach peryferyjnych oraz specyficznych właściwościach układu. Zmiana typu procesora docelowego jest możliwa również po napisaniu programu, np. gdy okazuje się, że tworzony projekt łatwiej zrealizować przy wykorzystaniu układu o większej liczbie liczników. Podczas kompilacji programu jednocześnie może być generowanych wiele plików wynikowych. Pliki te, mimo że generowane dla jednego typu mikrokontrolera, są kompilowane z różnymi ustawieniami. Dzięki temu podczas jednej kompilacji można wygenerować pliki wynikowe dla układów taktowanych sygnałami o różnych częstotliwościach, zawierających zewnętrzną pamięć programu lub pracujących tylko z wbudowaną pamięcią itp.

Na CD-ROM-ie znajdują się pliki w formacie PDF z dokumentacjami wszystkich obsługiwanych przez Keil C51 procesorów. Po wybraniu typu procesora docelowego w oknie zarządcy projektu w zakładce Books stają się dostępne pliki z dokumentacją producenta wybranego procesora. Dzięki temu programista ma stały i szybki dostęp do bogatej dokumentacji.

Nad bardziej rozbudowanymi projektami łatwiej panować dzieląc je na mniejsze części. Zakładka Files w oknie zarządcy projektu umożliwia łatwą nawigację pomiędzy plikami wchodzącymi w skład projektu (rys.



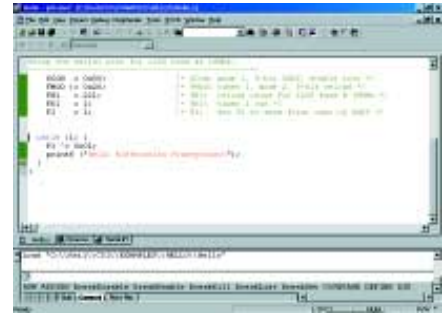
Rys. 3.

2). Pliki można łączyć w grupy, można również utworzyć grupę zawierającą dokumentację tekstową projektu. Narzędzie to nie jest jednak zbyt dopracowane, na przykład nie można zmienić nazwy istniejącej grupy. Po naciśnięciu prawego klawisza myszy wyświetla się menu kontekstowe, w którym można ustawiać specyficzne opcje zarówno dla grupy plików, jak i dla pojedynczych plików. Po wybraniu pliku zostaje wyświetlona jego zawartość w oknie edytora. Wchodzący w skład  $\mu$ Vision2 edytor wykonuje wcięcia tekstu zgodnie ze standardami przyjętymi dla języka C. Dodatkowo kod źródłowy jest wyświetlany w różnych kolorach, podobnie jak ma to miejsce np. w kompilatorach firmy Inprise. Okno edytora można podzielić i jednocześnie edytować dwa fragmenty tego samego pliku (rys. 3).

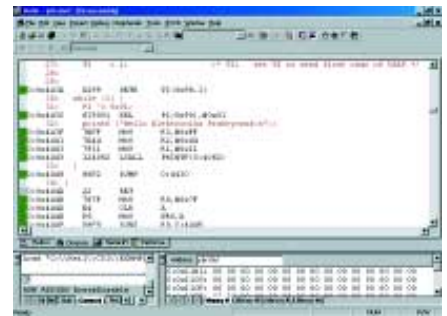
W skład  $\mu$ Vision2 wchodzi zaawansowany debugger. Śledzenie wykonywania programu może odbywać się przy podglądzie kodu w C (rys. 4) lub w assemblerze (rys. 5). Program może być wykonywany linia po linii, do kursora lub do zaznaczonego miejsca. W trakcie śledzenia wykonywania programu istnieje możliwość podglądu stanu wszystkich elementów procesora (rejestrów, pamięci, portów, liczników itp.) oraz pamięci i wartości zmiennych użytych w programie (rys. 6). Możliwe jest utworzenie funkcji symulujących analogowe zmiany napięcia na liniach wejściowych procesora. Funkcje te mogą być aktywowane automatycznie przez debugger w miejscach określonych przez programistę lub mogą być włączane ręcznie.

Jednym z elementów  $\mu$ Vision2 jest system operacyjny czasu rzeczywistego RTX51. Umożliwia on łatwe i szybkie budowanie aplikacji wymagających odporności na załamek czasowe. Dzięki RTX51 można również tworzyć programy wielozadaniowe, co wydawać by się mogło przesadą w przypadku mikrokontrolerów. Wystarczy jednak wyobrazić sobie sterownik wysyłający szeregowo wyniki pomiarów, który jednocześnie je wykonuje, odczytuje klawiaturę i wyświetla wyniki na wyświetlaczu. Zrealizowanie takiego projektu jest naprawdę nieskomplikowane i przede wszystkim łatwiejsze w późniejszej modyfikacji, gdyż kod z poszczególnymi zadaniami może być rozdzielony. Przełączanie między zadaniami może odbywać na dwa sposoby - do każdego zadania można określić czas wykonywania przez procesor lub zadania są wykonywane zgodnie z priorytetami określonymi przez programistę.

Dla kogo jest przeznaczony Keil C51? Ze względu na wysoką cenę (kilka tysięcy marek) krąg odbiorców jest ograniczony do programistów wymagających specyficznych możliwości systemu. Zintegrowane środowisko  $\mu$ Vision2, znacznie ułatwiające pracę programistę, bardzo dobry debugger i szybki symulator, umożliwiające przeprowadzenie dokładnej analizy programu - te rzeczy są spotykane w znacznie tańszym lub wręcz darmowym oprogramowaniu. Do rzadkości natomiast należy system operacyjny czasu rzeczywistego RTX51 z wbudowaną wielozadaniowością. Dla większości elektroników hobbystów istotne znaczenie mają biblioteki funkcji dodatkowych, których producent niestety nie dołącza do C51. Dodatkowe funkcje, obsługujące szynę I<sup>2</sup>C lub wyświetlacz



Rys. 4.



Rys. 5.

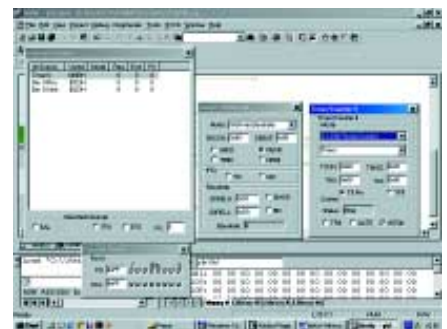
LCD można znaleźć nawet w kompilatorze Basica BASCOM. Umożliwiają one błyskawiczne uzyskanie ciekawych efektów.

Ile czasu zajmie napisanie programu szyfrowatora/deszzyfrowatora realizującego algorytm DES w assemblerze? Ile czasu zajmie napisanie programu zegara z programowanym alarmem (oczywiście również w assemblerze)? A ile czasu zajmie napisanie tych samych programów w języku wysokiego poziomu (sprawą drugorzędna jest wybór tego języka, z równym powodzeniem może to być C, Pascal, Fortran czy Basic)? Odpowiedzi na te pytania jednoznacznie wskazują na wzrost znaczenia programowania w językach wysokiego poziomu, gdyż coraz bardziej istotny staje się czas tworzenia programu i łatwość jego modyfikacji. Oczywiście nie oznacza to, że assembler odchodzi do lamusa. Bez dobrej znajomości assemblera oraz budowy wewnętrznej procesora niemożliwe będzie pełne wykorzystanie możliwości procesora.

**Paweł Zbysiński**

*Keil C51 do testów w redakcji dostarczyła firma WG Electronics s.c. z Warszawy, tel. (0-22) 621-77-04, <http://www.wg.com.pl>.*

*Testowa wersja kompilatora Keil C51 znajduje się na płycie CD-EP06/2000.*



Rys. 6.