

C dla mikrokontrolerów 8051

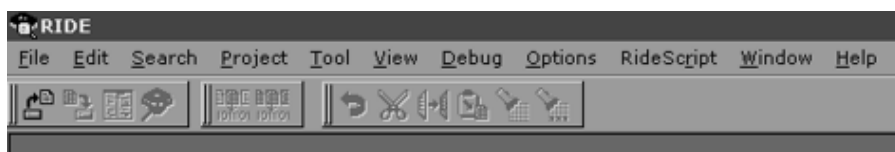
część 7

W związku z pytaniami Czytelników dotyczącymi programów zawartych w pakiecie RIDE, w kolejnym

odcinku kursu zdecydowałem się wrócić do opisu jego użytkowania.

W artykule zawarto informacje na temat edycji programu, tworzenia zbioru projektu, symulacji wykonania programu oraz tworzenia zbiorów wynikowych w formatach BIN i HEX. Trzeba jednak podkreślić, że objętość tego artykułu nie pozwala na wyczerpujące omówienie wszystkich aspektów użytkowania pakietu.

Jak posługiwać się kompilatorem C, część 1



Rys. 1

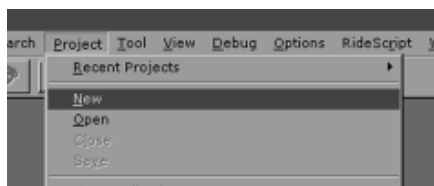
Raisonance RIDE

Pakiet RIDE firmy Raisonance jest zbiorem programów narzędziowych umożliwiających przygotowywanie i uruchamianie programów w języku C i w assemblerze dla mikrokontrolerów ST62 firmy STM, XA firmy Philips oraz innych mikrokontrolerów należących do rodziny 8051. Ze względu na planowane następne odcinki kursu, pominię zagadnienia dotyczące środowiska programowania ST62 i XA - zajmujemy się wyłącznie mikrokontrolerem 8051. Nie oznacza to jednak, że nie można tego opisu wykorzystać także dla innych mikrokontrolerów.

RIDE (*Raisonance Integrated Development Environment*) zawiera następujące programy przeznaczone dla mikrokontrolerów z rodziny 8051:

- kompilator języka assembler **MA-51** (jest to makroassembler),
- kompilator języka C **RC-51**,
- linker **LX-51**,
- symulator współpracujący ze sprzętowym emulatorem,
- edytor przystosowany do składni języków assembler oraz C (wyróżniane są słowa kluczowe obu języków).

Wersja demonstracyjna pakietu jest na płycie CD dołączonej do EP6/2002 oraz w Internecie, na przykład na stronie producenta pod adresem: <http://www.raisonance.com/download.php>.



Rys. 2

Instalacja i uruchomienie

Jeśli masz oryginalną płytę od producenta, pakiet zainstaluje się prawie bez Twojego udziału. Program instalacyjny znajdujący się na płycie CD uruchamia się po umieszczeniu jej w napędzie. Jeśli natomiast pobrałeś wersję instalacyjną z Internetu, poprzez wybranie *Start* i *Uruchom* możesz rozpocząć instalację.

Po zainstalowaniu pakiet gotowy jest do pracy - bez żadnych dodatkowych ustawień czy restartowania komputera. Ustaw rozdzielczość ekranu 800x600 lub większą - łatwiej będzie odczytać wyświetlane informacje. Po pierwszym uruchomieniu zobaczysz tylko menu główne pakietu. Ponieważ wcześniej nie otwierałeś żadnych plików, to na ekranie będzie wyłącznie otwarte okno programu i szare tło.

Menu główne

Omówmy poszczególne, najbardziej dla nas użyteczne pozycje menu (**rys. 1**). Nie jest moim celem pisanie instrukcji obsługi programu - tym zajął się już producent. Chcę natomiast podać tyle informacji, ile jest koniecznych, aby zacząć pracę z programem.

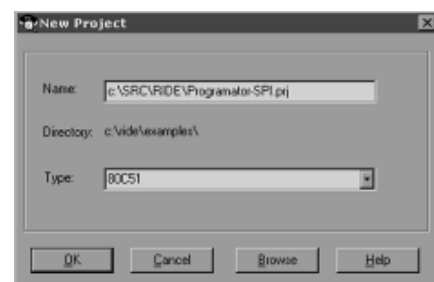
Do rozpoczęcia pisania pierwszego programu w języku C wystarczy wybrać w menu *File>New* (**rys. 2**). Następnie należy wybrać *C Files* i edytor gotowy jest do pracy. Możesz zawsze tak postępować w przypadku krótkich programów, ale nie jest to zbyt dobry sposób.

Dlaczego? Menu zawiera pozycję *Options*. Po jej wyborze można ustawić ważne dla programu źródłowe oraz urządzenia docelowe paramet-

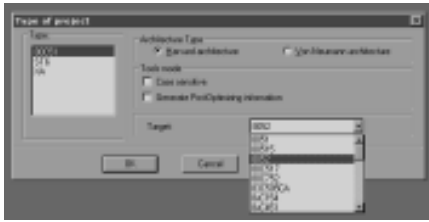
ry. W przypadku, gdy nie jest aktywny żaden z projektów (opiszę to w dalszej części artykułu), ustawione parametry są domyślnymi dla wszystkich tworzonych programów i wszystkich tworzonych urządzeniach, w których mikrokontroler ma współpracować. Jest to na przykład typ mikrokontrolera i opcje jego rozszerzeń. Rzadko zdarza się, że są one identyczne dla różnych urządzeń. Jeśli jednak wykorzystujesz tylko jeden typ mikrokontrolera od określonego producenta oraz twój program nie zawiera modułów w assemblerze, to możesz wybrać właśnie ten prosty sposób. Nie polecam jednak tego. Znacznie bardziej efektywne jest posługiwanie się projektami.

Każdy tworzony projekt jest obiektem, dla którego indywidualnie możesz określić właściwości i składowe. Później otwarcie zapamiętanego projektu powoduje przywrócenie cech środowiska, dla którego napisana została dana aplikacja - parametry domyślne zostaną zastąpione właściwymi dla stosowanego przez Ciebie sprzętu.

Aby utworzyć projekt, wybierz z menu *Project>New*. Pojawi się okienko (**rys. 3**) z pytaniem o folder, w którym na dysku ma być zapamiętany twój projekt i jak ma się nazywać. Folder będzie zawierał również zbiory wynikowe typu HEX, BIN lub w obu tych formatach jednocześnie. Ja przechowuję wszystkie projekty w jednym katalogu. Ułatwia to ich archiwizację, a selekcją zbiorów zaj-



Rys. 3

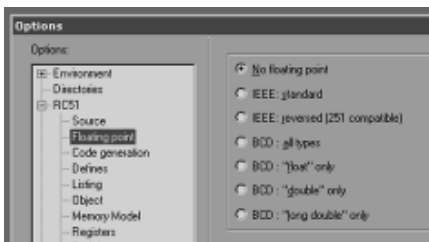


Rys. 4

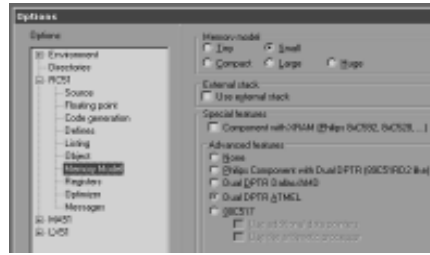
muje się RIDE. System Windows oferuje możliwość tworzenia nazw opisowych, więc należy z nich korzystać. Nazywaj swoje pliki tak, abyś mógł po jakimś czasie je rozpoznać. Tu jedna ważna uwaga - plik projektu nie może mieć takiej samej nazwy, jak któryś z jego składników. To oznacza, że jeśli twój projekt ma nazwę *Kontroler-LCD*, to żaden z modułów nie może mieć tej samej nazwy. Nie powinieneś tworzyć plików *Kontroler-LCD.c* albo *Kontroler-LCD.a51*. Wytłumaczenie jest bardzo proste. Kompilator nie przetwarza całego kodu źródłowego, lecz poszczególne jego moduły jeden po drugim. Na początku na przykład może być kompilowany napisany w języku C, potem w assemblerze. Na podstawie plików - nazwijmy je przejściowych - linker tworzy plik w formacie HEX (lub BIN albo też oba jednocześnie) przeznaczony do zapisania w pamięci mikrokontrolera. I tu docieramy do sedna sprawy. Kompilator generując pliki przejściowe, używa tej samej nazwy, którą ma zbiór źródłowy, zmieniając jedynie rozszerzenia. Czyli plikiem wynikowym kompilacji *Kontroler-LCD.a51* będzie *Kontroler-LCD.obj*. Łatwo jest więc wysnuć wniosek, że nowo tworzone pliki będą zastępowały te starsze. Być może ta cecha kompilatora RC-51 została usunięta w najnowszej wersji pakietu, jednak jest to stosowane również przez innych producentów programów tego rodzaju.

Ustawianie opcji zbioru projektu

Po otwarciu okna nowego projektu wpisz jego nazwę i wybierz model mikrokontrolera. Ten wybór jest wyborem rodzaju kompilatora, a nie nazw rejestrów. Te zdefiniujesz później, dołączając zbiory definicji oraz



Rys. 5



Rys. 6

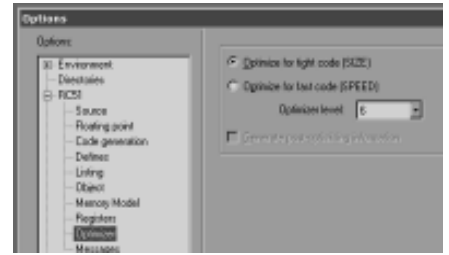
określając właściwości sprzętu. Zakładam, że utworzyłeś już nowy projekt i że dotyczy on mikrokontrolera z rdzeniem podobnym do 80C51. Zatwierdź wybór za pomocą OK. Teraz ustaw opcje środowiska, dla którego pisany jest program. Wybierz *Option>Target* (rys. 4).

Następnie zaznacz *Harvard architecture* i wybierz typ mikrokontrolera. Załóżmy, że do pierwszego projektu będziesz używał AT89S8252 - wybierz więc na liście 8052. Nie jest to dokładnie ten sam mikrokontroler, ale bardzo do niego podobny. Zatwierdź wybór za pomocą OK.

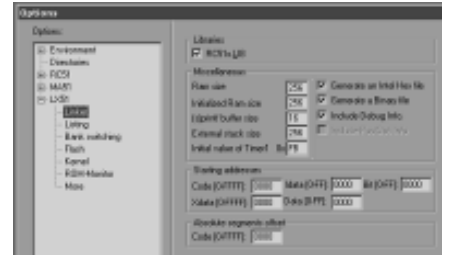
Teraz kolejna grupa ustawień, z których typowo potrzebujesz ustawić tylko kilka. Wybierz *Options>Project*. Raczej nie musisz nic zmieniać w *Environment* i *Directories*. Są tam nastawy dotyczące dołączanych zbiorów bibliotek oraz kolorów, czcionek i sposobów w jaki wyświetla komunikaty kompilator. Najbliższą „gałęzią” drzewa *Options*, którą powinieneś się zainteresować jest *RC51*. Zawiera ona wiele istotnych informacji dotyczących kompilatora:

- jeśli nie używasz liczb zmiennoprzecinkowych, zaznacz *No floating point*, w przeciwnym przypadku proponuję *IEEE standard* (rys. 5),
- wybierz zakładkę *Memory Model* (rys. 6) i zdecyduj, jakiego modelu pamięci będzie używał mikrokontroler (typowo wystarczy *Small*: tego rodzaju nastawy używam dla AT89S8252, gdy planowany rozmiar programu nie wykracza poza 8kB pamięci Flash) - zaznacz opcje specjalne dla AT89S8252 (jest to *Dual DPTR ATMEL*),
- wybierz zakładkę *Optimizer* (rys. 7) i zaznacz, czy kod wynikowy ma być optymalny pod względem rozmiaru (*Size*), czy szybkości wykonywania programu (*Speed* - zazwyczaj ją zaznaczam) - nastawę *Optimizer level* możesz pominąć,
- pozostałe parametry nastaw kompilatora dla RC51 nie są ważne. Przejdź zatem do grupy *LX51* i wybierz *Linker* - jest tu wiele bardzo ważnych nastaw (rys. 8).

Większość programatorów korzysta ze zbiorów wynikowych w formacie Intel HEX, tak więc zostaw za-



Rys. 7



Rys. 8

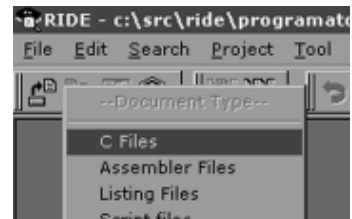
znaczoną opcję *Generate an Intel Hex File*. Jeśli twój programator wymaga pliku w formacie binarnym, zaznacz również *Generate a Binary File (BIN)*. Wpisz rozmiar pamięci RAM size, *Initialized RAM size* - dla AT89S8252 jest to 256 bajtów. W dolnej części okienka masz jeszcze możliwość określenia od jakiego adresu w pamięci Flash czy RAM umieszczony zostanie kod wynikowy i zmienne (zazwyczaj nie potrzebujesz jednak zmieniać tych nastaw).

Na tym możesz zakończyć ustawianie opcji projektu - wybierz OK. Jak zdążyłeś się zapewne zorientować, w poszczególnych zakładkach znajduje się jeszcze wiele innych użytecznych opcji, jednak do napisania pierwszej aplikacji w języku C dla AT89S8252 wystarczy to, co już poznałeś.

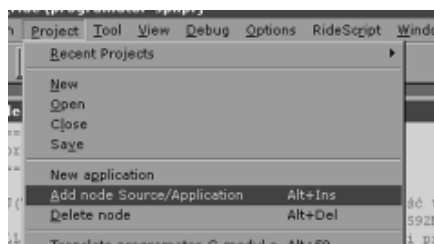
Dołączanie plików źródłowych do projektu

Wcześniej utworzyłeś plik projektu. Jest on jednak pusty. Oprócz właściwości nie zawiera żadnych składowych. Teraz postaram się pokazać, jak do projektu dodać już istniejące oraz nowe pliki źródłowe.

W celu utworzenia nowego pliku należy wybrać *File>New* (rys. 9). W zależności od rodzaju tworzonego pliku wybierz odpowiedni typ. Załóżmy, że tym pierwszym tworzymy

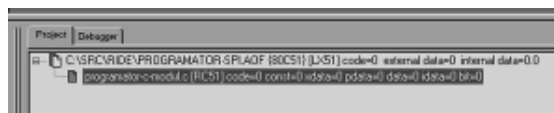


Rys. 9



Rys. 10

przez Ciebie zbiorem będzie plik źródłowy programu w języku C. Wybierz więc *C Files*. Na ekranie pojawi się okienko edytora, w którym będziesz mógł wpisać program. Na wszelki wypadek zapamiętaj utworzony zbiór, wybierając *File>Save as*. Pamiętaj, aby zapisywany na dysku zbiór źródłowy miał inną nazwę niż projekt. Zapisywane pliki typu *C Files* mają domyślnie nadawane rozszerzenie **.c*. Zapamiętując plik na dys-



Rys. 11



Rys. 12

ku, nie musisz podawać rozszerzenia - zostanie one dodane automatycznie. Inaczej jest ze zbiorami nagłówkowymi typu „h”. Po nazwie musisz dopisać rozszerzenie. Ja zapamiętałem swój zbiór projektu pod nazwą *programator-ISP*, a moduł w języku C pod nazwą *programator-C-modul*.

Utworzenie nowego pliku i zapamiętanie na dysku nie powoduje dodania go do projektu! Aby dodać zbiór, musisz posłużyć się poleceniem *Project>Add node Source/Application* (rys. 10 - skrót klawiszowy Alt + Insert).

Nazwa dodanego w ten sposób pliku powinna pojawić się na liście w okienku *Project* znajdującym się zazwyczaj na dole ekranu (rys. 11).

Podobnie postępuj z innymi zbiorami projektu. Ja dołączyłem jeszcze *programator-asm-modul.a51*. Dołączaj tylko pliki źródłowe niewymienione



Rys. 13

po dyrektywie *#include* w tekście programu - te dołączają się automatycznie przy pierwszym użyciu.

Kompilowanie projektu

Aby skompilować tak utworzony projekt, zapamiętaj wszystkie jego pliki, wybierając *File>Save all*. Następnie z menu *Project* wybierz *Make all* (rys. 12) lub posłuż się przyciskiem umieszczonym w menu (rys. 13).

Jeśli napisałeś program poprawnie i kompilacja przebiegła bez błędów, na dysku powstaną pliki wynikowe o nazwie identycznej z nazwą projektu i rozszerzeniem **.HEX* (lub/i **.BIN*). W innych przypadkach na dole ekranu wyświetli się informacja o błędach.

Jacek Bogusz, AVT
jacek.bogusz@ep.com.pl